

Programmatūras integrācijas instrukcija (PR)

Valsts digitālās attīstības aģentūra

DATU IZPLATĪŠANAS UN PĀRVALDĪBAS PLATFORMA (DAGR)

Projekts Nr. VRAA/2023/113

VERSIJA 1.35

IZMAIŅU VĒSTURE

Versija	Datums	Apraksts
1.0	23.08.2022	Sākotnējā 1.laidiena versija
1.1	30.09.2022	Papildināta tabula 1.3 Termiņi un saīsinājumi
1.2	28.10.2022	Papildinājumi un labojumi atbilstoši Pasūtītāja sniegtajiem komentāriem
1.3	08.11.2022	Papildinājumi sadaļā 4.2
1.4	10.01.2023	Aktualizētas Postman piemēru kolekcijas saites
1.5	19.04.2023	Sākotnējā 2.laidiena versija
1.6	12.05.2023	Papildinājumi, labojumi un atbildes uz Pasūtītāja komentāriem atbilstoši 2.laidiena akcepttestēšanas rezultātiem
1.7	30.09.2023	3.laidiena papildinājumi
1.8	04.12.2023	3.laidiena labojumi
1.9	13.03.2024	Pievienota sadaļa "5.5.1 DAGR REST API izmantošana", ekspluatācijas laikā konstatēti precizējumi un papildinājumi
1.10	02.05.2024	Pievienota sadaļa "Datu anonimizācija". Dzēsta sadaļa "Datu nodošanas veida izvēle", kas dublēja sadaļu "Datu nodošanas mehānisma izvēle".
1.11	27.05.2024	Pievienota sadaļa "Pēdējo izmaiņu datums"
1.12	05.07.2024 09.07.2024	Pievienota sadaļa "Datu izmaiņu monitorings" Aktualizēta sadaļa "Audita apskate portālā"
1.13	16.10.2024	Pievienota sadaļa par jaunu Wiki JS "DAGR satura centra izveide, atbilstoši lietotāju vajadzībām"
1.14	29.11.2024 18.12.2024	Pievienota sadaļa ("Datu struktūras pielāgošana filtrēšanai") par VIRSIS filtriem. Ir veikti labojumi atbilstoši saņemtajiem komentāriem
1.15	20.12.2024	Pievienota sadaļa "DAGR audita informācijas paneļa izstrāde"
1.16	20.01.2025	Pievienots apraksts ar procesu aprakstu par DAGR pielāgojumiem portāla publiskās pieejas nodrošināšanai
1.17	28.04.2025	Pievienota sadaļa "Iestāžu lietotāju saraksts"
1.18	14.05.2025	Dokumentā mainīts lomu nosaukums "paraksttiesīgā kontaktpersona" uz "pilnvarotā persona", atbilstoši DU-8 prasībām
1.19	18.06.2025	Ir aktualizēti ekrānšāviņi šādas sadaļās: • Pielāgoti servisi • Pieteikšanās par datu devēju • Izplatāmo datu kopu definēšana • Datu devēja savienojumu reģistrēšana • Datu ņēmēja savienotāja reģistrēšana
1.20	20.06.2025	Pievienota sadaļa "Paziņojumi datu saņēmējiem" DU-9
1.21.	11.07.2025	Pievienota sadaļa"" 6.2 Apkārtraksti"
1.22.	18.07.2025	

1.23.	25.07.2025	Papildinātas sadaļas "5.6.2.2 Datu devēja savienotāja vispārīgā konfigurācija", "5.6.2.3 Datu struktūru kartējums" un "5.6.2.5 Žurnālēšanas uzstādījumi" ar informāciju par konfigurācijas un žurnāla notikumu nodošanu uz DAGR pašapkalpošanās portālu (DU12)>
1.24.	15.08.2025	Papildināta sadaļa "Izplatāmo datu kopu definēšana" ar informāciju par papildus meklēšanas atslēgu pievienošanu, kā arī papildināta sadaļa "Datu objekta izgūšana pēc objekta identifikatora" ar informāciju kā meklēt ierakstus pēc papildus meklēšanas atslēgas.
1.25.	26.09.2025	Sadaļas 5.3 saturs precizēts saskaņā ar DU-10 izmaiņām.
1.26.	16.10.2025	Saskaņā ar DU-11 ir aktualizētas šādas sadaļas: 4.3.3. Datu objekta izgūšana pēc objekta identifikatora 5.3.1. Datu kopas pievienošana priekšlikumu videi (sandbox)
1.27.	30.10.2025	Saskaņā ar DU-11 ir aktualizētas šādas sadaļas: - Meklēšanas atslēgu definēšana aprakstīta sadaļā: 5.3 Izplatāmo datu kopu definēšana → 5.3.1 Datu kopas pievienošana priekšlikumu videi (sandbox) - Praktisks atslēgu pielietojuma piemērs iekļauts pielikumā: 7.1 Datu kopu definīcijas - Par atļauju noskaidrošanu no datu saņēmēja puses papildināta sadaļa: 4.3.2 Atļauju noskaidrošana - Par atslēgu izmantošanu papildināta sadaļa: 4.3.3 Datu objekta izgūšana pēc objekta identifikatora
1.28	12.12.2025	Pievienota sadaļa "4.4 Datu izgūšana ar datu saņēmēja savienotāju"
1.29	30.12.2025	Pievienota sadaļa "Datu kopas reģistrēšana Latvijas atvērto datu portālā"
1.30	05.02.2026	Ir aktualizētas sadaļas: 4.4 Datu izgūšana ar datu saņēmēja savienotāju, apakšsadaļa 4.4.2.1 Vispārīgā konfigurācija (application.conf) 4.4.2.2. Datu kartējuma konfigurācija (connector.yaml)
1.31	27.02.2026	Pievienota sadaļa "Monitorings" (DU-18)
1.32	16.03.2026	Pievienota sadaļa "Auditācijas uzlabojumi" (DU-19)
1.33	27.03.2026	Ir aktualizēta sadaļa "4.4 Datu izgūšana ar datu saņēmēja savienotāju" atbilstoši DAGR2-1096
1.34	10.04.2026	Ir aktualizēta sadaļa "Auditācijas uzlabojumi" (DU-19)

1.35	20.04.2026	DAGR pašdiagnostikas risinājuma (DU-15) izstrādes ietvaros pievienota datu devējiem paredzēta sadaļa "Datu nodošanas procesa atklūdošana".
	28.04.2026	Ir aktualizēta sadaļa "Auditācijas uzlabojumi" (DU-19) atbilstoši komentāriem "DAGR datu devēja savienotāja pielāgojumi datu sinhronizācijai ar Fizisko personu reģistra informācijas sistēmas servisiem – ziņas par personu" (DU-17) ietvaros aktualizētas sadaļas – "Datu objekta izgūšana pēc objekta identifikatora" -> "PMLP datu īpatnības" un "Datu izmaiņu izgūšana un pilnas datu kopas pārlasīšana" -> "PMLP datu īpatnības".

Satura rādītājs

1	Ievads	7
1.1	Dokumenta nolūks	7
1.2	Projekta mērķi.....	7
1.3	Termini un saīsinājumi	7
2	DAGR vispārīgā arhitektūra.....	8
3	Integrācijas soļu plāns.....	9
3.1	Datu ņēmēja integrācijas soli	9
3.2	Datu devēja integrācijas soli	10
4	Datu ņēmēja integrācija	11
4.1	Iepazīšanās ar pieejamajiem datiem un servisiem	11
4.2	Pieteikšanās par datu ņēmēju.....	12
4.2.1	Datu ņēmēja savienojuma reģistrēšana.....	13
4.3	Datu izgūšana izmantojot DAGR REST API	15
4.3.1	Autentifikācija (drošības talona pieprasīšana).....	15
4.3.2	Atļauju noskaidrošana.....	17
4.3.3	Datu objekta izgūšana pēc objekta identifikatora	19
4.3.4	Datu izmaiņu izgūšana un pilnas datu kopas pārlasīšana	23
4.3.5	Monitorings.....	29
4.3.6	Pielāgoti servisi	43
4.3.7	Pēdējo izmaiņu datums.....	45
4.4	Datu izgūšana ar datu saņēmēja savienotāju	46
4.4.1	Savienojumu reģistrēšana.....	46
4.4.2	Savienotāja konfigurācija	46
4.4.3	Datu bāzes sagatavošana	51
4.4.4	Savienotāja palaišana un atklūdošana	53
4.4.5	Savienotāja darbības rezultāti.....	53
5	Datu devēja integrācija	54
5.1	Pieteikšanās par datu devēju	54
5.2	Iestāžu lietotāju saraksts.....	54
5.3	Izplatāmo datu kopu definēšana	57
5.3.1	Datu kopas pievienošana priekšlikumu videi (sandbox)	57
5.3.2	Datu kopas pārcelšana uz pamata sadaļu (dataset)	61
5.3.3	Datu kopas reģistrēšana Latvijas atvērto datu portālā	64

5.3.4	Specializētās datu kopas	66
5.3.5	Datu pievienošana caur failu portālā	68
5.3.6	Datu lejupielāde caur REST API	70
5.4	Datu nodošanas mehānisma izvēle.....	73
5.5	Datu devēja savienojuma reģistrēšana	75
5.6	Datu nodošana uz DAGR	77
5.6.1	DAGR REST API izmantošana.....	77
5.6.2	Datu devēja savienotāja uzstādīšana	83
5.6.3	Debeziū savienotājs	93
5.7	Personas datu izmantošanas žurnālpieraksti.....	94
5.8	Datu anonimizācija.....	94
5.8.1	Datu anonimizācija izmantojot datu devēja savienotāju	94
5.8.2	Datu anonimizācija caur DAGR pašapkalpošanās portālu	97
5.9	Datu pievienošana un lejupielāde portālā	99
5.9.1	Datu lejupielāde	99
5.10	Datu nodošanas procesa atklūdošana	99
6	Auditācijas uzlabojumi (DU-19)	99
6.1	Audita notikumi	99
6.1.1	Audita izgūšana caur REST API	100
6.2	Audita aplūkošana portālā	102
7	DAGR satura centra izveide, atbilstoši lietotāju vajadzībām	104
7.1	Paziņojumi datu saņēmējiem.....	104
7.2	Apkārtraksti.....	104
8	Pielikums – papildus paraugi.....	106
8.1	Datu kopu definīcijas.....	106

1 Ievads

Šis dokuments sagatavots saskaņā ar uzdevumu Nr.2 "Izmaiņu monitoringa un auditācijas pierakstu uzlabojumi" pie vispārīgās vienošanās ID Nr. VRAA EIS 2021/14/AK/CI-118-3-PKP_1 pasūtījumam Nr. VRAA/2023/113.

1.1 Dokumenta nolūks

Šis programmatūras integrācijas instrukcijas nolūks ir sniegt trešo pušu izstrādātājiem visu nepieciešamo informāciju, lai tie varētu veiksmīgi integrēt trešo pušu risinājumus ar DAGR.

1.2 Projekta mērķi

Programmatūras integrācijas instrukcijas sagatavota ņemot vērā projektam uzstādītos mērķus:

- Nodrošināt vienotu un ātrdarbīgu datu izplatīšanas risinājumu;
- Izveidot vienotu datu aprites auditēšanas mehānismu;
- Izstrādāt pašapkalpošanās sistēmu datu atļauju piešķiršanai datu devēja un datu saņēmēja datu aprites pārvaldībai.

1.3 Termini un saīsinājumi

Termins	Skaidrojums
DAGR	Datu izplatīšanas un pārvaldības platforma
DAGR API	Datu izplatīšanas un pārvaldības platformas programmatūras saskarne
Datu devējs	tiesību subjekts (iestāde, juridiskā persona, publiskā persona), kas tiek identificēts ar reģistrācijas numuru un nodod datus DAGR, lai tos izplatītu datuņēmējiem
Datu devēja savienotājs	DAGR reģistrēta tehnoloģiska komponente ar unikālu identifikatoru (<i>client id</i>), kas datu devēja vārdā veic datu ielādi DAGR kodolā
Datuņēmējs	tiesību subjekts, kas tiek identificēts ar reģistrācijas numuru un saņem datus no datu devēja caur DAGR
Datuņēmēja savienotājs	DAGR reģistrēta tehnoloģiska komponente ar savu identifikatoru (<i>client id</i>), kas datuņēmēja vārdā veic datu izguvi no DAGR
Datu kopa	datu devēja izplatāmie dati ar unikālu nosaukumu un noteiktu datu struktūru, kas sastāv no viena vai vairākiem datu objektiem
Datu objekts (datu objekta tips)	datu struktūras loģiska vienība ar noteiktu nosaukumu, kas sastāv no parametriem (t.sk. datu kvalitātes kritērijiem) un citiem datu objektiem
IS	Informācijas sistēma
Lietotājs	fiziska persona, kas sistēmā tiek autentificēta ar VPM un veic darbības datu devēja, datuņēmēja vai DAGR vārdā, atbilstoši lietotājam piešķirtajām sistēmas tiesībām
VIRSIS	Valsts informācijas resursu, sistēmu un sadarbības informācijas sistēma
VPM	Vienotais pieteikšanās modulis

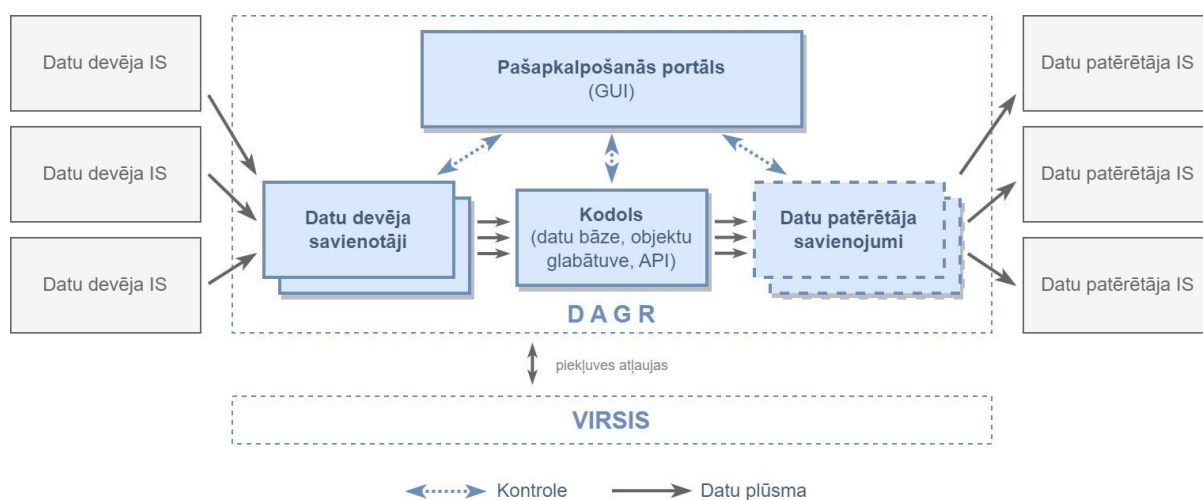
2 DAGR vispārīgā arhitektūra

DAGR nodrošina datu izplatīšanu datu ņēmējiem no datu devējiem. Lai nodrošinātu datu ņēmēju datu pieprasījumu apstrādes ātrdarbību un datu pieejamību arī gadījumos, kad datu devēja informācijas sistēmas nav pieejamas, caur DAGR izplatāmo datu kopija tiek uzglabāta DAGR kodolā.

Gan sākotnējā datu ielāde, gan datu aktualizācija no datu devēju IS notiek izmantojot datu devēja savienotājus. Datu devēja savienotāji ir tehnoloģiska komponente, kas atbalsta dažādus datu izgūšanas variantus no datu devēju IS un nodrošina šo datu sekmīgu nodošanu DAGR kodolam. Lai arī faktiskā datu aktualitāte ir atkarīga no izvēlēta datu izgūšanas variantā un datu devēja IS iespējām, datu devēja savienotāji tiek veidoti tā, lai varētu nodrošināt caur DAGR izplatāmo datu aktualitāti, kas ir mērāma sekundēs.

Datu ņēmējiem dati ir pieejami REST API formā no DAGR kodola. Datu izgūšanai tiek izmantoti DAGR pašapkalpošanās portālā definēti datu ņēmēja savienojumi, kas ir unikāli identificēti un reprezentē konkrētus datu ņēmēja pieslēgumus.

Pašapkalpošanās portāls ir DAGR lietotāja saskarne, kas ļauj lietotājiem pieteikties par datu ņēmējiem vai datu devējiem un pēc atļaujas saņemšanas pārvaldīt savus savienojumus un datu devēju gadījumā arī datu kopas.



Ilustrācija 1 - DAGR vispārīgās arhitektūras shēma

Dati DAGR tiek uzturēti datu kopās, kur katra datu kopa reprezentē noteikta datu devēja biznesa datus ar konkrētu datu struktūru.

DAGR funkcija ir nodrošināt datu apmaiņu, savukārt datu piekļuves tiesības tiek pārvaldītas sistēmā VIRISIS. DAGR ļauj datu ņēmējiem iegūt datus atbilstoši VIRISIS reģistrētajām datu piekļuves atļaujām. Datu piekļuves atļaujas ir ar granularitāti līdz konkrētiem datu laukiem.

Tālāk šajā dokumentā aprakstīts kā pieteikties par datu ņēmēju un iegūt datus caur DAGR vai kā pieteikties par datu devēju un izplatīt datus caur DAGR.

3 Integrācijas soļu plāns

3.1 Datu ņēmēja integrācijas soļi

Lai pieteiktos par datu ņēmēju un sekmīgi integrētu izstrādājamās risinājumus ar DAGR ir veicamas šādas darbības:

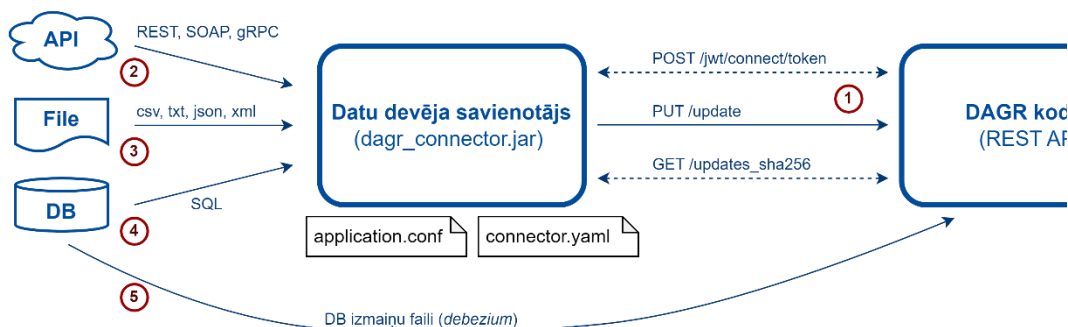
Nr.	Darbība	Apraksts
0.	Iepazīties ar pieejamajām datu kopām	Pirms uzsākt darbus ir vēlams iepazīties ar datu kopām, kuras datu devēji izplata caur DAGR, lai noskaidrotu ka nepieciešamās datu kopas ir pieejamas. Skatīt sadaļu "4.1 Iepazīšanās ar pieejamajiem datiem un servisiem".
1.	Iegūt datu piekļuves atļauju VIRSIS	Lai iegūtu atļauju piekļūt datu kopām iestādei jāreģistrējas VIRSIS, jāreģistrē pieteikums un jāiegūst datu devēja apstiprinājums. Piebilde: Šo soli var veikt paralēli nākošajiem soļiem, līdz 4. soļa uzsākšanai.
2.	Pieteikt datu ņēmēju	Lai uzsāktu darbu ar DAGR, ir jāaizpilda DAGR veidlapa un jāiesniedz DAGR pārzinim, sūtot uz VRAA oficiālo elektronisko adresi caur Latvija.lv portālu vai pa pastu. Vairāk informācijas: https://www.vdaa.gov.lv/lv/datu-izplatisanas-un-parvaldibas-platforma Skatīt sadaļu " Error! Reference source not found. Error! Reference source not found. ".
3.	Reģistrēt datu ņēmēja savienojumu	Pēc pieteikuma apstiprināšanas pieteikumā norādītajai personai ir jāpieslēdzas DAGR pašapkalpošanās portālam un jāreģistrē datu ņēmēja savienojums. Skatīt sadaļu "4.2.1 Datu ņēmēja savienojuma reģistrēšana".
4.	Izgūt datus	Jāizmanto DAGR API datu iegūšanai. Vispirms, norādot reģistrētā datu ņēmēja savienojuma autentifikācijas informāciju, DAGR API jāpieprasa drošības talons un pēc tam ar saņemto drošības talonu var veikt datu pieprasījumus. Skatīt sadaļu "4.3 Datu iegūšana".

3.2 Datu devēja integrācijas soļi

Lai pieteiktos par datu devēju un sekmīgi nodotu datus izplatīšanai uz DAGR ir veicamas šādas darbības:

- | N | Dar | Apraksts |
|---|-----|---|
| 1 | Pie | Lai uzsāktu darbu ar DAGR, ir jāaizpilda DAGR veidlapa un jāiesniedz DAGR pārzinim, sūtot uz VRAA oficiālo elektronisko adresi caur Latvija.lv portālu vai pa pastu. Vairāk informācijas: https://www.vdaa.gov.lv/lv/datu-izplatisanas-un-parvaldibas-platforma |
| 2 | Dat | Datu devējs definē caur DAGR izplatāmās datu kopas. |
| 3 | Dat | Lai uzsāktu datu nodošanu, ir jāreģistrē datu devēja savienojums, ko datu devēja pārstāvis var izdarīt DAGR pašapkalpošanās portāla. Skatīt sadaļu "0 Datu devēja savienojuma reģistrēšana". |
| 4 | Dat | Ja datu devēja savienotājs tiek uzstādīts datu devēja infrastruktūrā, tad šo soli datu devējam jāveic pašam. Ja tas tiek uzstādīts DAGR centrālajā infrastruktūrā, šo soli jālūdz veikt DAGR pārvaldniekam. |
- 3.3 Datu devējs lejupielādē datu devēja savienotāju un tā uzstādīšanas instrukcijas, veic datu devēja savienotāja konfigurēšanu un iedarbina to datu devēja

infrastruktūrā. Skatīt sadaļu "5.4 Datu nodošanas mehānisma izvēle



Ilustrācija 2 - datu nodošana uz DAGR

Pastāv vairāki varianti kā nodot datus no datu avota uz DAGR kodu:

- Izmantojot DAGR REST API.** Datu devējs var organizēt datu nodošanu uz DAGR izmantojot DAGR REST API pieejamos datu nodošanas servissus. Serviss *POST /jwt/connect/token* izmantojams drošības talona pieprasīšanai. Tālāk serviss *PUT /update* izmantojams datu nodošanai. Papildus serviss *GET /updates_sha256*, ļauj izgūt sarakstu ar DAGR kodolā esošo ierakstu identifikatoriem un ierakstu vērtību kontrolsummām, kas var tikt izmantots nododamo izmaiņu atpazīšanai. Šis datu nodošanas variants ir salīdzinoši vienkāršs, bet prasa izmaiņas datu devēja informācijas sistēmā. Detalizētāks datu nodošanas mehānisma apraksts atrodams "5.6.1 DAGR REST API izmantošana".
- Izmantojot datu devēja eksistējošos REST, SOAP vai gRPC datu izplatīšanas servissus.** Ja datu devējam ir eksistējoši datu izplatīšanas servisi, kas nodrošina iespēju atklāt datu kopas ierakstus un to izmaiņas, tad datu nodošanai uz DAGR var izmantot šos servissus. Šajā gadījumā datu nodošana uz DAGR tiks organizēta izmantojot datu devēja savienotāju, kurš nolasīs datus no eksistējošajiem datu devēja servisiem un nodos uz DAGR kodu caur DAGR REST API. Šis variants neprasa izmaiņas datu devēja pusē, bet prasa pielāgotas izstrādes darbus DAGR projekta pusē, lai iestrādātu datu devēja eksistējošo servisu atbalstu datu devēja savienotāja komponentē.
- Nolasot datus no failiem.** Datus uz DAGR kodu var nodot no csv, txt, json vai xml tipa failiem. Šajā gadījumā datu nodošana uz DAGR organizējama izmantojot datu devēja savienotāju, kurš nolasīs datus no failiem un nodos tos uz DAGR kodu caur DAGR REST API. Pieņemot, ka datu avota sistēmai eksistē mehānisms kā eksportēt datus uz failiem, šis datu nodošanas mehānisms neprasa izstrādes pielāgojumus ne datu devēja ne DAGR pusē – viss kas nepieciešams ir uzstādīt un nokonfigurēt datu devēja savienotāju.
- Nolasot datus no DBVS ar SQL.** Datus uz DAGR kodu var nodot izmantojot lasīšanas režīma SQL pieslēgumu avota datu bāzei. Šajā gadījumā datu nodošana uz DAGR organizējama izmantojot datu devēja savienotāju, kurš nolasīs datus no datu bāzes izmantojot JDBC pieslēgumu un nodos tos uz DAGR kodu caur DAGR REST API. Lai nebūtu nepieciešams atvērt piekļuvi avota datu bāzei no ārējā tīkla, šajā gadījumā datu devēja savienotāju rekomendējams uzstādīt datu devēja infrastruktūrā. Šis datu nodošanas mehānisms neprasa izstrādes pielāgojumus ne datu devēja ne DAGR pusē - viss kas nepieciešams ir uzstādīt un nokonfigurēt datu devēja savienotāju.

5. **Nolasot datus no DBVS transakciju datnēm.** Datu nodošanu uz DAGR iespējams organizēt izmantojot *Debezium* (<https://debezium.io/>) tehnoloģiju, kas ļauj datus nolasīt no datu bāzu transakciju failiem. Šajā gadījumā uzstādāms *Debezium* pieslēgums datu avota datu bāzei, kurš datus nodos uz DAGR kodola Kafka rindu. Šis datu nodošanas mehānisms neprasa izstrādes pielāgojumus ne datu devēja ne DAGR pusē, bet tas var prasīt būtiskas izmaiņas avota DBVS konfigurācijā.

Ja datu nodošana tiek organizēta izmantojot datu devēja savienotāju (otrais līdz ceturtais datu nodošanas mehānisms), tad pastāv vēl jautājums kur šo savienotāju izvietot. Pēc noklusējuma datu devēja savienotājs uzstādāms datu devēja infrastruktūrā. Taču vienojoties ar DAGR pārvaldnieku un atverot nepieciešamās piekļuves datu avotam, ir iespējama arī datu devēja savienotāja izvietošana DAGR centrālajā infrastruktūrā.

No esošās pieredzes sevi vislabāk ir pierādījusi datu izgūšana no DBVS ar SQL pieprasījumiem (ceturtais variants) vai failiem (trešais variants) izmantojot datu devēja savienotāju. Salīdzinoši vienkārša ir arī datu tieša ierakstīšana DAGR REST API (pirmais variants).

3.4 Datu devēja savienojuma reģistrēšana

Lai izplatītu datus ir jāreģistrē datu devēja savienojums. Ja tiek izplatīta viena datu kopa no viena avota, tad izmantojams viens savienojums. Bet ja datu devējs izplata vairākas datu kopas no vairākiem avotiem, tad katram datu avotam rekomendējams reģistrēt savu datu devēja savienojumu.

Datu devēja savienojumu iespējams reģistrēt pašapkalpošanās portāla sadaļas “DAGR” apakš sadaļā “Savienotāji”. Šajā sadaļā ir pieejams saraksts ar visiem savienojumiem:

DAGR > Savienotāji > Savienotāju pārskats

DAGR ^

Pieteikumu pārvaldība

Savienotāji

Iestāžu lietotāji

Zināšanu bāze

Monitorings ^

Audita notikumi

Izplatīšana ^

Datu kopu katalogs

Savienotāji

DATU DEVĒJA IZVĒLE

Autotransporta direkcija

Jauns Testa		Neaktīvs
Ptaugusta Sē		Neaktīvs
PT_testATD PtrudensSAV		Neaktīvs
pt_atd_test Sertifikāti: 1. 18.10.2024	5cc9621e964	Aktīvs
VDEĀK test vdeak_test_II Sertifikāti: 1. 2.10.2025.	8a0df3465d	Aktīvs

+ Jauns savienot

Lai reģistrētu jaunu savienojumu jāaizpilda forma:

DAGR

- Pieteikumu pārvaldība
- Savienotāji**
- Zināšanu bāze

Monitorings

- Audita notikumi

Izplatīšana

- Datu kopas
- Anonimizācija

Patērēšana

- VIRSIIS atļaujas
- Izmaiņu monitorings

Datu savienotāja reģistrācija

NOSAUKUMS: Savienojuma nosaukums pilniem vārdiem

IDENTIFIKATORS: VDAA_DEMO_SAV_1

DATU SAVIENOTĀJA TIPS: consumer

☒ **Aktīvs?**

TALONA DZĪVES LAIKS, S: 900

APRAKSTS: Savienojums, kas tiek reģistrēts DAGR zināšanu bāzes ekrānšāviņu v.

✓ Saglabāt Atgriezties pie saraksta

Šajā formā jānorāda portālā savienojuma nosaukums (informatīva nozīme), identifikators (tiks izmantots datu devēja savienotāja konfigurācijā), pazīme vai savienojums aktīvs, drošības talona dzīves cikla ilgums sekundēs un opcionāls apraksts.

Kad savienojums ir saglabāts, tam iespējams pievienot X.509 standarta drošības sertifikātu.

DAGR

- Pieteikumu pārvaldība
- Savienotāji**
- Zināšanu bāze

Monitorings

- Audita notikumi

Datu savienotāja informācija

Vispārīgi **Sertifikāti**

+ Pievienot

1.	SHA256withRSA for CN=DAGR, OU=IT, O=UNISO, L=Rīga, ST=Rīga, C=LV 66d16e0a5c952e443b79b3d9ffa61018e3a9ec30	2.10.2
----	--	--------

Drošības sertifikāta pievienošana datu devēja savienotāja reģistrācijas scenārijā neatšķiras no datu ņēmēja savienotāja reģistrācijas scenārija – detalizētāku aprakstu par sertifikātiem skatīt sadaļā “4.2.1 Datu ņēmēja savienojuma reģistrēšana”.

3.5 Datu nodošana uz DAGR

Datu nodošanu uz DAGR konceptuāli iespējams organizēt vienā no trim variantiem:

- Izmantojot **DAGR REST API**. Šajā gadījumā datu nodošanu uz DAGR kontrolē datu avota sistēma;
- Izmantojot **datu devēja savienotāju**. Šajā gadījumā datus izgūšanu un nodošanu uz DAGR kontrolē datu devēja savienotājs. Datu devēja savienotājs datus var izgūt no SQL tipa datu bāzes, failiem (CSV, TXT, XML, JSON) vai eksistējošajiem datu devēja datu izplatīšanas servisiem;
- Izmantojot **Debezium** savienotāju. *Debezium* savienotājs ļauj izgūt datus no datu bāzu žurnālfailiem.

Tālākajās nodaļās aprakstīts katrs no variantiem.

3.5.1 DAGR REST API izmantošana

Datu nodošanu uz DAGR datu devējs var organizēt pats savā sistēmā izmantojot DAGR REST API.

Vispirms tiek pieprasīts drošības talons un tālāk ar to tiek darbināti šādi servisi:

- **PUT {dagr_adrese}/update/{service_name}**. Šis serviss nosūta strukturētus datus uz DAGR JSON formātā;
- **PUT {dagr_adrese}/file/{namespace}/{path_and_filename}**. Šis serviss uz DAGR nosūta datnes. Datu devējam pieejams arī DELETE serviss datnes dzēšanai un GET serviss datnes lejupielādei;
- **GET {dagr_adrese}/updates_sha256/{service_name}**. Šis serviss ļauj izgūt uz DAGR nosūtīto strukturēto datu ierakstu identifikatorus un kontrolsummas. Serviss izmantojams izmaiņu atpazīšanai kā arī atklādošanas nolūkiem;
- **GET {dagr_adrese}/updates_last_timestamp/{service_name}**. Šis serviss ļauj noteikt pēdējo uz DAGR nosūtīto izmaiņu datumu un laiku. Servisu var izmantot izmaiņu atpazīšanai, ja datu avotā tiek uzturēti izmaiņu veikšanas datumi un laiki.

Minēto servisu apraksti OpenAPI standartā pieejami: <https://dagr.gov.lv/api-docs/update>

3.5.1.1 Drošības talona pieprasīšana

DAGR REST API autentifikācija ir organizēta saskaņā ar *OAuth 2.0 OpenId Connect* standarta *Client Credentials Flow* plūsmu.

Drošības talona pieprasīšanas adrese: <https://dagr.gov.lv/jwt/connect/token>

Drošības talona izsaukuma piemērs:

```
curl --location 'https://dagr.gov.lv/jwt/connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cache-Control: no-cache' \
--data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' \
--data-urlencode
'client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJEQUdSX0FLQ19URVNUXzIiLCJqdGkiOiJKOTQwZTliNC0yYjgwLTQ4NzUtYmEyMS1mYmVmYmVhYzUzZjIiLCJpc3MiOiJEQUdSX0FLQ19URVNUXzIiLCJhdWQiOiJlcm46VEVTVClMTkI6REFHUl5TVFMvY29ubmVjdC90b2t1biIsImV4cCI6IjE3MDEyOTQ5NjQ1LCJmYmYiOiIxNzA5MjZkMzY0In0.Xq3bZNI62hS9OFYx-gNBEbh2SwQt72to_gL3Xwa9yh_uYUqd5__tDJuXkz-F03Jn0CdcRKcdetrKTtSCxIlP3WqQpbotsPHZvuF0ek26koyhtGhK2lw6RNF8mPkIktb4L6F0GuTjwc9ZmQ10lYo86u6_ZZWe2sS8aISJAUBmClW5cIt_qc3xY8no_UUJwDlyRGfxntGw3T_IyvIv0bPOLlOvyfmswpcI7I5S2qR5EIJlKggM4SninbGfBlj_u7HGSziO14kIXmOikdhDviyP0je9kYBcwZgwYeja19dBxo1NkDd6A3BG3jVPlvxJG2QQLHdWNkHbXSonyJajDhGnA' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=DAGR_AKC_TEST_2' \
--data-urlencode 'scope=producer'
```

Šajā izsaukumā kā *client_id* ir jānorāda datu devēja savienojuma identifikators (skatīt sadaļu “0

Datu devēja savienojuma reģistrēšana”) un *client_assertion* ir jāsigatavo atbilstoši RFC 7515 JSON Web Signature (JWS) standartam.

Respektīvi, *client_assertion* sastāv no 'Header' un 'Payload', kas parakstīti ar datu ņēmēja savienotāja privāto atslēgu. 'Header' veidojams kā JSON objekts, kas satur paraksta parametrus, piemēram:

```
{
```

}

1

}

C

}

r

P

8

5

4

L

```
"value": "string"
}
```

Lauku nozīme:

- “key” ir ieraksta primārā atslēga. Ja uz DAGR nododamais ieraksts saturēs jaunu iepriekš uz DAGR nenodotu atslēgas vērtību, tad tas tiks uzskatīts par jaunu ierakstu un pievienots DAGR datu kopai kā jauns ieraksts. Ja uz DAGR nododamais ieraksts saturēs iepriekš uz DAGR jau nodotu atslēgas vērtību, tad tās tiks uzskatītas par ieraksta izmaiņām un jaunā ieraksta vērtība pārrakstīs iepriekšējo vērtību.
- “timestamp” ir neobligāti norādāma vērtība. Pēc noklusējuma par šo vērtību tiks uzstādīts datu nodošanas brīža datums un laiks. Manuāli šo vērtību paredzēts norādīt tikai veicot pirmreizējo datu nodošanu uz DAGR, lai uz DAGR nodotu informāciju par ierakstu pēdējo izmaiņu laiku avota sistēmā. Ja šī vērtība tiek norādīta, tad ieraksti iesūtīšanas brīdī kārtojami pēc šīs vērtības augošā secībā. Konstatējot situāciju, kur ierakstāmā ieraksta “timestamp” mazāks par iepriekšējā ieraksta “timestamp”, DAGR ierakstāmā ieraksta norādīto “timestamp” aizvietos ar ierakstīšanas brīža datumu un laiku.
- “value” ir paša ieraksta biznesa dati. Value ir norādāma kā burtciparu (string) virkne, kas satur eskeiptu JSON datu objektu. Ja uz DAGR tiek nodota tukša ieraksta vērtība, tad tā tiek uzskatīta par dzēšanas operāciju un ieraksts no DAGR tiek dzēsts.

Tātad, izmantojot /update servisu var veikt šādas trīs darbības:

- Pievienot jaunu ierakstu – padodot uz /update servisu ierakstu ar jaunu “key” vērtību;
- Atjaunināt ieraksta vērtību – padodot uz /update servisu ierakstu ar eksistējošu “key” vērtību;
- Dzēst ierakstu – padodot uz /update servisu ierakstu ar eksistējošu “key” vērtību un “value” norādot tukšu (null) vērtību.

Datu nodošanas servisa izsaukuma piemērs (trīs ierakstu pievienošana):

```
curl --location --request PUT
'https://dagr.gov.lv/sandbox/40003562863/update/tst_company_service_report' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlc46VEVTVClMTkI6REFHUi5TVFmILCJzdW
IiOiI0MDAwMzU2Mjg2MyIsImF1ZCI6WyJwcm9kdWNlciIsInVyb2pURVNUUxOQjpeQUdSLlNUUy9yZXNvd
XJjZXMiXSwiZXhwIjoxNzEwNDExNjE2Mjg2MyIsImNsaWVudF9pZCI6ImRlbW9fZGRzX2NvbW51Y3Rvc18xIiwic2N
vcGUlOlsicHJvZHVjZSI6IiXX0.aOahd4FnJXzt5rjBzw_zMtFWL0gBUIT2rwIG7bcdZLkd8NqEsYttq2qQqk
fnncwm-iSn9uYV-lOM6q02NHcXVA68NU56PQkX_DvM-
Vl251GftZFR8z9Tu64JlEaBWI2DR7x6BG7ULxO_brZUaBSn6VrBkGxNe0YoerJ5uMJZex7q461ed5UxCk0H
l8-BE4eWSE-
OJCnaTAHCnLaCEiIQgiwL9jbbq4hmWTe_O_OEO5zwRv0QieLhHQ7whBallI5fX0WSH0cXINT7GRwcqP5Uu
MbpQ_dwnBcEtcS-Iaj3Auz9ycrU3mIkldld4EnVs1OeaLmXakBk8AvR8mRHIq7A' \
--data '{
  {
    "key": "40003562863/2023/5634634",
    "value": "{\n  \"Id\": 5634634,\n  \"CompanyName\": \"UNISO SIA\", \n
    \"RegistrationNumber\": \"40003562863\", \n  \"IsLicensed\": true, \n  \"Year\":
    2023, \n  \"Status\": \"apstiprināts\", \n  \"Submitted\": \"2024-03-
    06T09:22:11.593Z\", \n  \"Clients\": [\n    {\n      \"CompanyName\": \"VARAM\", \n
    \"RegistrationNumber\": \"90000028508\", \n      \"ClientSince\": \"2022-01-12\", \n
    \"Services\": [\n      {\n        \"Name\": \"Programmatūras izstrāde\", \n
    \"DateFrom\": \"2023-01-01\", \n        \"DateTo\": \"2023-12-31\", \n      }
    ] \n    }, \n    {\n      \"VerifiedBy\": {\n        \"Name\": \"Jānis\", \n        \"Surname\":
    \"Sprukts\", \n        \"Position\": \"Sistēmu arhitekts\" \n      } \n    }
  ],
  {
    "key": "40003562863/2022/5634234",
    "value": "{\n  \"Id\": 5634234,\n  \"CompanyName\": \"UNISO SIA\", \n
    \"RegistrationNumber\": \"40003562863\", \n  \"IsLicensed\": true, \n  \"Year\":
    2022, \n  \"Status\": \"apstiprināts\", \n  \"Submitted\": \"2023-03-
    06T09:22:11.593Z\", \n  \"Clients\": [\n    {\n      \"CompanyName\": \"VARAM\", \n
```



```

\ "RegistrationNumber\": \"90000028508\", \n      \ "ClientSince\": \"2022-01-12\", \n
\ "Services\": [ \n          { \n              \ "Name\": \"Programmatūras izstrāde\", \n
\ "DateFrom\": \"2022-01-12\", \n              \ "DateTo\": \"2022-12-31\" \n
}, \n \n \n \n \n          \ "Name\": \"Konsultācijas\", \n              \ "DateFrom\": \"2022-
02-12\", \n              \ "DateTo\": \"2022-03-12\" \n          } \n      ] \n }, \n \n \n \n \n
\ "CompanyName\": \"VRAA\", \n      \ "RegistrationNumber\": \"90001733697\", \n
\ "ClientSince\": \"2022-01-01\", \n          \ "Services\": [ \n              { \n
\ "Name\": \"Programmatūras izstrāde\", \n              \ "DateFrom\": \"2022-01-01\", \n
\ "DateTo\": \"2022-12-31\" \n          } \n      ] \n }, \n \n \n \n \n \n \n \n \n \n \n \n \n
\ "Name\": \"Jānis\", \n      \ "Surname\": \"Sprukts\", \n      \ "Position\": \"Sistēmu
arhitekts\" \n      } \n } \n
},
{
    "key": "40003562863/2022/56332",
    "value": "{ \"Id\": 56332, \"RegistrationNumber\": \"40003562863\",
\ "Year\": 2022 }"
}
] '

```

Piemērā redzams, ka galvenē (*header*) tiek padots drošības talons un tālāk saturā (*data*) tiek padotas divas ierakstu vērtības. Drošības talona pieprasīšana aprakstīta sadaļā “5.6.1.1 Drošības talona pieprasīšana”, savukārt viens no piemērā nododamajiem ierakstiem neeskeipotā formātā izskatās šādi:

```

{
  "Id": 5634634,
  "CompanyName": "UNISO SIA",
  "RegistrationNumber": "40003562863",
  "IsLicensed": true,
  "Year": 2023,
  "Status": "apstiprināts",
  "Submitted": "2024-03-06T09:22:11.593Z",
  "Clients": [
    {
      "CompanyName": "VARAM",
      "RegistrationNumber": "90000028508",
      "ClientSince": "2022-01-12",
      "Services": [
        {
          "Name": "Programmatūras izstrāde",
          "DateFrom": "2023-01-01",
          "DateTo": "2023-12-31"
        }
      ]
    }
  ],
  "VerifiedBy": {
    "Name": "Jānis",
    "Surname": "Sprukts",
    "Position": "Sistēmu arhitekts"
  }
}

```

Atbilstošā datu kopas definīcija (skatīt sadaļu “5.3 Izplatāmo datu kopu definēšana”), izskatās šādi:

```

name:      tst_company_service_report
key:       RegistrationNumber, Year(, Id)
data_description: Gada atskaite par uzņēmuma sniegtajiem pakalpojumiem
fields:
- Id                long                : Atskaites ID
- CompanyName       : Uzņēmuma nosaukums
- RegistrationNumber : Uzņēmuma reģistrācijas numurs
- IsLicensed        boolean            : Ir licenzēts pakalpojumu sniedzējs
- Year              int                 : Gads
- Status            : Atskaites statuss
- Submitted         timestamp          : Atskaites iesniegšanas datums un
laiks
- Clients * :
  comments          : Uzņēmuma klienti
  fields:
  - CompanyName     : Uzņēmuma nosaukums
  - RegistrationNumber : Uzņēmuma reģistrācijas numurs
  - ClientSince      date              : Datums kopš kura ir klients
  - Services * :
    comments        : Sniegtie pakalpojumi

```

```

fields:
- Name : Pakalpojuma nosaukums
- DateFrom date : pakalpojuma sniegšanas sākuma
datums
- DateTo date : Pakalpojuma sniegšanas beigu datums
- VerifiedBy :
  comments : Pārskata pārbaudītājs
  fields:
  - Name : Vārds
  - Surname : Uzvārds
  - Position : Amats

```

Datu nodošanas servisa izsaukuma piemērs (ieraksta dzēšana):

```
curl --location --request PUT \
'https://dagr.gov.lv/sandbox/40003562863/update/tst_company_service_report' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcM46VEVTVClMTkI6REFHUi5TVFMIcCJzdWlTioiIOMDAwMzU2Mjg2MyIsImFlZCI6WyJwcmlmNmNlciIsInVyb3JURVNUUxOQjEpeQUdSLlNUUyYyZXNvdXJjZXMiXSwiZXBhwIjoNazA5NzIxNjIyLGlCYmYiOiB3MDk3MjA3MjIsImRhdiC6MTcwOTcyMDCyMiwiGVnYWxlbnRpdHkiOiIOMDAwMzU2Mjg2MyIsImNsaWVudF9pZCI6ImRlbW9fZGRzX2NvbW5lY3Rvc18xIiwicC2NvcGU0lsicHJVZHVhZDIiXX0.Qm0lUSjb1BRoX9Xkt8HPeERjHf9Ul_e6CYsZLSzYj7d5OHHLSoB8EH-ZS8apIGbk8tQKbdBtBKHKbgVMPIosqndVIguKNTq4drZn_IogkhvucHIO4yHtYSne240HlvIctfLzi2_7vcYb8r816XeS_Y3PdS7CwQWLH5I_w407Uzp71FGS8hNuh8BBE7RcScL4TVZjMeuxVh67BeMNHsrNREumeuPoPHEGABrDTK2YEn_4biphbGW5iYY5a4PnoKtoAN_pChs0Ug0Ojm18eAK8fahDF7c4nNLmemNzyidOmwhOSCClwjW_N82N1qX7OPrFluszc7AYmORLGk5SGMD2Q' \
--data '{
{
    "key": "40003562863/2023/5634634",
    "value": null
}
}'
```

3.5.1.3 Izmaiņu atpazīšana

Ja nepieciešams konstatēt kādi ieraksti ir nodoti uz DAGR, var izmantot servisu **GET {dagr_adrese}/updates_sha256/{service_name}**. Šis serviss atgriež sarakstu ar ierakstu atslēgām un SHA256 ierakstu vērtību kontrolsummām uz DAGR nodotajiem datiem. Ilglaiīgi katram unikālajam ierakstam jeb ieraksta atslēgai DAGR uzglabā pēdējo nodoto vērtību un kontrolsummu, bet īslaicīgi un izmēģinājuma vidē (*sandbox*) tiek uzglabātas visas nodotās ierakstu vērtības un kontrolsummas.

updates_sha256 servisa atbildes piemērs:

```
[
  {
    "key": "40003562863/2023/5634634",
    "offset": 0,
    "sha256":
"b4b5f8353f1d6bc7ab1db2637afa28cb1ce33fb67e4e0cef7f0d785884c8eac5"
  },
  {
    "key": "40003562863/2022/5634234",
    "offset": 1,
    "sha256":
"a6d2b783425ca15c86fbb88c3a05fe643c2e78e7bb57bd581e8fedffc0a4bdcc"
  },
  {
    "key": "40003562863/2023/5634634",
    "offset": 2,
    "sha256": null
  },
  {
    "key": "40003562863/2022/56332",
    "offset": 3,
    "sha256":
"42a70d82a114352e4af97610d8b4c756ff6586cc7557c5ac12af583dd18d5346"
  }
]
```

Piemēra redzam, ka ierakstam ar atslēgu 40003562863/2023/5634634 uz DAGR vispirms nodota viena vērtība, tad cita vērtība un visbeidzot ieraksts dzēsts. Pēc tam uz DAGR nodots ieraksts ar atslēgu 40003562863/2022/56332.

SHA256 kontrolsummu rēķināšanai DAGR izmanto OpenSSL bibliotēku. Online rīks, kas pārbaudīts, ka sakrīt ar DAGR aprēķinātajām SHA256 vērtībām:
<https://tools.keycdn.com/sha256-online-generator>

Lai nodrošinātu, ka uz DAGR tiek nodotas tikai izmaiņas, izmantojot *updates_sha256* servisu, realizējams šāds mehānisms:

- Jāsagatavo saraksts ar datu avotā esošo ierakstu atslēgām un aprēķinātajām ierakstu vērtību sha256 kontrolsummām;
- Jāsaņem saraksts ar DAGR ierakstīto ierakstu atslēgām un sha256 kontrolsummām no DAGR izmantojot *updates_sha256* servisu;
- Jāsalīdzina saraksti un jāveic šādas darbības:
 - Visi ieraksti ar atslēgām, kuras ir avota sistēmā, bet nav DAGR ir jauni ieraksti. Tie ir nododami uz DAGR;
 - Visi ieraksti ar sakrītošām atslēgām un sakrītošām kontrolsummām ir ieraksti kuru vērtības nav mainījušās. Šos ierakstus uz DAGR nodot nav nepieciešams;
 - Visi ieraksti ar sakrītošām atslēgām un dažādām kontrolsummām ir modificēti ieraksti. Šo ierakstu jaunās vērtības ir nododamas uz DAGR;
 - Visi ieraksti ar atslēgām, kuras ir DAGR, bet nav avota sistēmā ir dzēsti ieraksti. Šo ierakstu dzēšanas fakts ir nododams uz DAGR (līdzīgi kā tiek nodots jauns vai modificēts ieraksts, tikai ieraksta vērtības laukā norādot *null*).

Piebilde: Ja DAGR atgriež vairākus ierakstus ar vienādām atslēgām, tad faktiskā pēdējā ierakstītā vērtība būs ar lielāko *offset* vērtību.

Augstāk minētais mehānisms ir tāds pats kā tiek izmantots datu devēja savienotājā.

3.5.1.4 Failu pievienošana

Failu nodošanai uz DAGR izmantojams serviss **PUT**

/file/{namespace}/{path_and_filename}, kur {namespace} ir datu devēja datu kopām izmantotā vārdtelpa.

Datnes pievienošanas piemērs:

```
curl -X 'PUT' \
  'https://dagr.gov.lv/file/vsaa/test_file_name.txt' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJ1cm46VEVTVClMTkI6REFHUl5TVFMiLCJzdW
IiOiI5MDAwMTY2OTQ5NiIsImF1ZCI6WyJwcm9kdWNlciIsInVyb2pURVNUUxOQjpbEQUdSLlNUUy9yZXNvd
XJjZXMiXSswiZlZlIjoxNzEwMzI4MDI4LCJ1YmYiOiJlZ3MTAzMjc3MjgsIm1hdCI6MTcxMDMyNzEyOCwibGVn
YWxlb3RpdHkiOiI5MDAwMTY2OTQ5NiIsImNsaWVudF9pZCI6IjZTUUFfVEVTVF9ERFNFMSIsInNjb3B1Ijpb
bInByb2R1Y2VyIl19.gZgNI6451nrLAMYp4UHRWLBtGivPjIBxPpP1j2xiU5RyUf21rad3CWdmyFyBIrlhz
SjmQ5ITvrtJrvlMm0ID4j3BPoDiUsrOtr-6OFEbR3rwYLruQyu_CiHDBLbyfyWr8KKSbyov834-
37xlwrZjablC-t_tncs7Uyd9-pvQp_RjkVmEYgwfPLiq-
edOyczQQ4WwcYwVoYKgH05yA5TX3LQ7n4oOGvJvMa9KoZenU2a49y4bLAXwQGKtxV_kox75ABMXroaGT51l
1FpTRz1AnzCBh6n4D3s3D0novL2IpgYgBxA5jhybkwJpd1ynD_n4SVQDJN1Lp_UiRYq-gnbcGw' \
  -H 'Content-Type: application/octet-stream' \
  --data-binary '@test_file.txt'
```

Papildus PUT servisam, datu devējam pieejams arī **DELETE**

/file/{namespace}/{path_and_filename} serviss, kas ļauj datni dzēst un **GET**

/file/{namespace}/{path_and_filename}, kas ļauj datni lejupielādēt.

Datu devēja savienotāja uzstādīšana”.

- 4 Dat Alternatīvi datu devēja savienotāja uzstādīšanai datu devēja infrastruktūrā datu devējs
b u DAGR pārvaldniekam var lūgt datu devēja savienotāju uzstādīt DAGR centrālajā
de infrastruktūrā.
vēj
a

sav
ien
otā
ja
uzs
tād
īša
na
un
ko
nfi
gur
ēša
na
DA
GR
infr
ast
ruk
tūr
ā

4 Datu ņēmēja integrācija

4.1 Iepazīšanās ar pieejamajiem datiem un servisiem

DAGR API ir aprakstīts OpenAPI (<https://www.openapis.org/>) standartam atbilstošā formātā un pieejams adresē: <https://dagr.gov.lv/api>

Saraksts ar datu kopām un servisiem caur kuriem tiek izplatīti datu devēju dati pieejams servisā: https://dagr.gov.lv/service_list, kur katrai datu kopai ir norādīts:

- Datu kopas tehniskais un pilnais nosaukums;
- Datu devējs (īpašnieka) reģistrācijas numurs un nosaukums.

Katrai no datu kopām ir pieejams serviss, kas ļauj izgūt datu kopas ierakstus pēc objekta identifikatora un serviss, kas ļauj izgūt sarakstu ar izmainītajiem ierakstiem. Izmainīto ierakstu servisu var izmantot gan pirmreizējai visu datu kopas ierakstu atklāšanai, gan jaunāko izmaiņu monitoringam. Atsevišķām datu kopām ir pieejami arī papildus izmaiņu saraksti, kas satur tikai noteiktus biznesa notikumus.

Atsevišķo servisu apraksti pieejami <https://dagr.gov.lv/api> sadaļā "Datu servisi".

Piebilde: Lai izsauktu datu izgūšanas servisu pieprasījumā ir jānorāda VIRSIS atļaujas numurs. VIRSIS atļaujas numuri tiek piešķirti servisu līmenī. Respektīvi, datu devēji caur VIRSIS var izvēlēties vai piešķirt datu ņēmējiem tikai atļauju izgūt ierakstus pēc jau zināmiem ierakstu identifikatoriem vai piešķirt arī atļauju sekot izmaiņām un izgūt informāciju par visiem ierakstiem.

4.2 Pieteikšanās par datu ņēmēju

Lai uzsāktu darbu ar DAGR, ir jāaizpilda DAGR veidlapa un jāiesniedz DAGR pārzinim, sūtot uz VRAA oficiālo elektronisko adresi caur Latvija.lv portālu vai pa pastu. Vairāk informācijas:

<https://www.vdaa.gov.lv/lv/datu-izplatisanas-un-parvaldibas-platforma>

Veidlapas apstrādes ietvaros DAGR pārzinis reģistrēs datu ņēmēja pieteikumu DAGR sistēmā un veidlapā norādītajai personai piešķirs “pilnvarotā kontaktpersona” lomu. Tālāk šī persona varēs autentificēties DAGR pašapkalpošanās portālā un darboties datu ņēmēja iestādes vārdā. DAGR pašapkalpošanās adreses:

- Testa videi: <https://test.dagr.gov.lv/>
- Produkcijas videi: <https://dagr.gov.lv/>

Loma “pilnvarotā kontaktpersona” ļaus lietotājam iestādes pieteikumam pievienot un noņemt citus lietotājus tādējādi pārvaldot iestādes lietotāju sarakstu:

Valsts digitālās attīstības aģentūra

A A A

Jānis Sprukts
Privātpersona

DAGR > Pieteikumu pārvaldība > Pieteikums

Datu ņēmējs

REĢISTRĀCIJAS NUMURS 40003562863

NOSAUKUMS Sabiedrība ar ierobežotu atbildību "UNISO"

STATUSS ✓ Apstiprināts

VĀRDS Jānis

SURNAME Testiņš

PERSONAS KODS 324324-12344

TĀLRUNIS +37121234235

E-PASTA ADRESE test@test.lv

☒ paraksttiesīgā kontaktpersona

☐ tehniskā kontaktpersona

Pārbaudīt un saglabāt

Dzēst

Pievienot jaunu kontaktpersonu

✓ Saglabāt pieteikuma izmaiņas

Atgriezties pie saraksta

Bez lomas “pilnvarotā kontaktpersona” pastāv vēl arī loma “tehniskā kontaktpersona”, kas ļauj veikt visas pārējās darbības, bet neļauj pārvaldīt iestādes lietotājus.

4.2.1 Datu ņēmēja savienojuma reģistrēšana

Pēc tam, kad ir apstiprināts datu ņēmēja pieteikums, ir jāreģistrē datu ņēmēja savienojums. Ja datu ņēmējam ir vairākas informācijas sistēmas, kas izgūst datus no DAGR, katrai rekomendējams reģistrēt savu savienojumu.

Datu ņēmēja savienojumus iespējams reģistrēt pašapkalpošanās portāla sadaļas “Piekluve DAGR” apakš sadaļā “Savienotāji”. Šajā sadaļā ir pieejams saraksts ar visiem savienotājiem / savienojumiem:

Savienotāji

DATU DEVĒJA IZVĒLE

Valsts digitālās attīstības aģentūra

	peterisTestJunijs Peteristest	✓ Aktīvs	Dzēst
	PtaugustaSav PtaugustaSav	✓ Aktīvs	Dzēst
	VRAA datu patērētāja savienotājs testēšanas vajadzībām VRAA_DEV_DPS_1	✓ Aktīvs	Dzēst
	Alisei Testet alisepanTest	✓ Aktīvs	Dzēst
	Savienotājs portāla anonimizācijas uzdevumu izpildei - PATĒRĒTĀJS dagris_anonymizer	✓ Aktīvs	Dzēst
	Datu devēja savienojums testēšanas vajadzībām VRAA_DEV_DDS_TEST_1	✓ Aktīvs	Dzēst
	Alisei Testet alisepanTest	✓ Aktīvs	Dzēst
	Savienotājs portāla anonimizācijas uzdevumu izpildei - DATU AVOTS dagris_anonymizer	✓ Aktīvs	Dzēst

Lai reģistrētu jaunu savienotāju jāaizpilda forma:

Valsts digitālās attīstības aģentūra

AAA

Jānis Sprukts
Privātpersona

DAGR > Savienotāji > Jauns savienotājs

DAGR

Pieteikumu pārvaldība

Savienotāji

Zināšanu bāze

Monitorings

Audīta notikumi

Izplatīšana

Datu kopas

Anonimizācija

Patērēšana

VIRIS atļaujas

Izmaiņu monitorings

Datu savienotāja registrācija

NOSAUKUMS

Savienojuma nosaukums pilniem vārdiem

IDENTIFIKATORS

VDAA_DEMO_SAV_1

DATU SAVIENOTĀJA TIPS

consumer

☒ Aktīvs?

TALONA DZĪVES LAIKS, S

900

APRAKSTS

Savienojums, kas tiek reģistrēts DAGR zināšanu bāzes ekrānšāviņu vajadzībām

Saglabāt

Atgriezties pie saraksta

Šajā formā jānorāda portālā savienotāja nosaukums (informatīva nozīme), identifikators (tiek izmantots REST API), pazīme vai savienojums aktīvs, drošības talona dzīves cikla ilgums sekundēs un opcionāls apraksts.

Kad savienojums ir saglabāts, tam iespējams pievienot X.509 standarta drošības sertifikātu:

Valsts digitālās attīstības aģentūra

AAA

Jānis Sprukts
Privātpersona

DAGR > Savienotāji > Valsts digitālās attīstības aģentūra

DAGR

Pieteikumu pārvaldība

Savienotāji

Zināšanu bāze

Monitorings

Audīta notikumi

Datu savienotāja informācija

Vispārīgi

Sertifikāti

Pievienot jaunu sertifikātu

1.

SHA256withRSA for CN=DAGR, OU=IT, O=UNISO, L=Rīga, ST=Rīga, C=LV
66d16e0a5c952e443b79b3d9ffa61018e3a9ec30

2.10.2024.

Dzēst

DAGR > Savienotāji > Savienotāju pārskats > Sertifikāts

DAGR

Pieteikumu pārvaldība

Savienotāji

Lietotāju pārvaldība

Portāla teksti

Sertifikāta pievienošana

PUBLISKĀ ATSLĒGA (*.CER)

Choose File

No file chosen

Saglabāt

Atgriezties pie saraksta

24

Šajā izsaukumā *client_id* ir jānorāda datu ņēmēja savienotāja identifikators un *client_assertion* ir jāsigatavo atbilstoši [RFC 7515 JSON Web Signature\(JWS\)](#) standartam. Respektīvi, *client_assertion* sastāv no 'Header' un 'Payload', kas parakstīti ar datu ņēmēja savienotāja privāto atslēgu. 'Header' veidojams kā JSON objekts, kas satur paraksta parametrus, piemēram:

```
{
  "typ" : "JWT",
  "alg" : "RS256"
}
```

'Payload' ir iekļaujama šāda informācija:

```
{
  "sub": {datu ņēmēja savienotāja id},
  "jti": {uuid},
  "iss": {datu ņēmēja savienotāja id},
  "aud": urn:LNB:DAGR.STS/resources,
  "exp": {derīguma termiņš, max nbfd + 1 stunda},
  "nbfd": {parakstīšanas datums un laiks}
}
```

Drošības talona izsaukuma atbildes piemērs:

```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjIxMEZCQkFCNzE3NzFCRTcyOTIyNTk0NTBBMjhDMjcYnZc5NTZFmkVSUzI1NiIsInR5cCI6ImF0K2p3dCIsIngldCI6IklRLTdxM0YzRy1jcElzbEZDaWpDY25lVmJtNCJ9.eyJuYmYiOjE2Nzk1MDYwODYSImV4cCI6MTY3OTUwOTY4NiwiXNzIjoidXJuOlRFU1Q7tTE5COkrBR1IuU1RTIiwiaXVkIjoideXJuOlRFU1QtTE5COMRhZ3IuY29yZS5hcGkiLCJjbGlbnRfaWQiOiJVTKlTT190ZXNOX2Nvbmc5Y3Rvc18xIiwibGVhbnRpdHkiOiIOMDAwMzU2Mjg2MyIsImp0aSI6IjA5OERGMYTYqOY2OEY0Mjk4OTJENTk3Qjk3RjlCNTBBIiwiaWF0IjoxNjc5NTA2MDg2LCJzY29wZSI6WyJkYWdyX2NvcmVfYXBpIl19.Cm0kgNBkfNR0Mo63ZLkCYMZPZ9v8XbfHWE1Lgvz907k1Ama2SqIWE68oMcBlAXJEUYj8ipQCY7aCmjVEN3AZ6qv1lcRpTURQ53EUKXS5B8S3aQfxOu2ne3rnDKhaZXi0rHRMTAUctJWmeXWFA5m-FcvFrhzgTYuM2zq-L5VY-mgJ1VOzOXj9_5eUALBLHLBJF91LPZQckao4VAN-Zwvly_snlioJFK6409AJVzc5pvdAL4AqaMrQ6QtrrxIkOEa8c6qxOMKGIXTMncc_uwjteES3HhVsrJuZkQdDW0zCMKIokXj8UGkd6PskJdP3b28aHAlogGcoZkL5m80sXvfHQ",
    "expires_in": 3600,
    "token_type": "Bearer",
    "scope": "consumer"
}
```

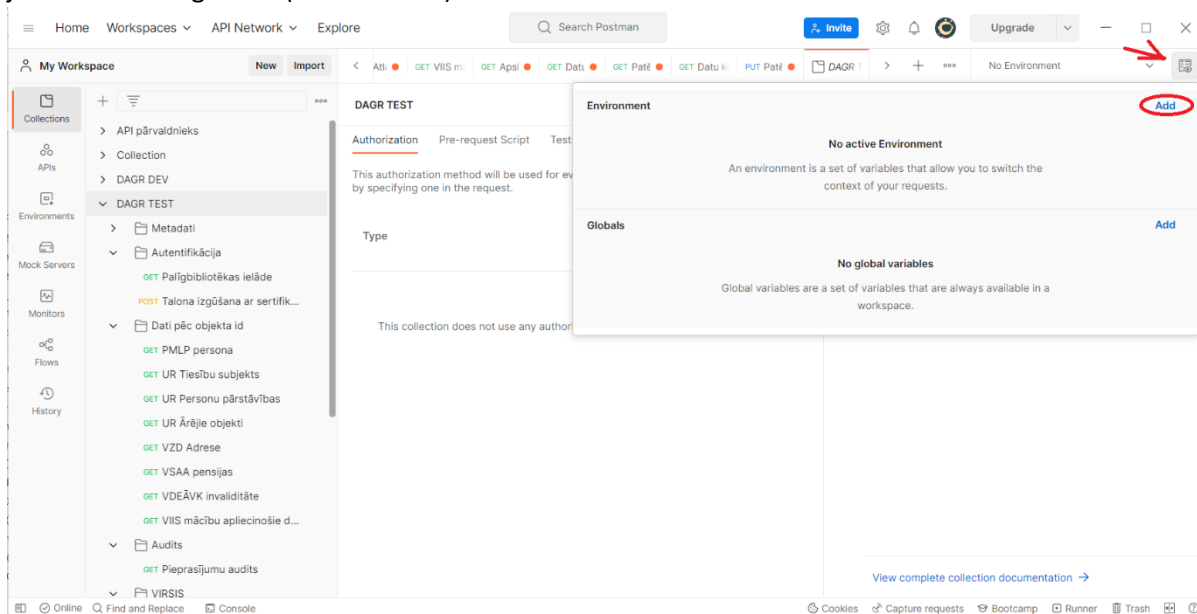
Izsniegtais drošības talons atbilst [RFC 7519](#) standartam un drošības talona satura pārlūkošanai ērti izmantot: <https://jwt.io/>

Postman piemērs drošības talona izsaukumam un datu izguves izsaukumiem:

https://api.postman.com/collections/9631194-e7289644-5aa8-4639-8d4f-80c8591f2a49?access_key=PMAT-01GPE6A92NR3T5PNH5MTB9KMWD

Lai Postman programmā ielādētu kolekciju jāizvēlas darbība “File” -> “Import” un jāiekopē augstākminētajā adresē pieejamais teksts.

Lai varētu izpildīt drošības talona pieprasīšanas piemēru, Postman programmā ir jāpievieno un jāaktivizē mainīgo vide (*environment*).



4.3.2 Atļauju noskaidrošana

Lai izgūtu datus pieprasījumā ir jānorāda VIRSIS izsniegtās atļaujas numurs. Datu ņēmējam pieejamās atļaujas var izgūt arī no DAGR. Ar katru datu kopu var būt saistīta viena vai vairākas aktīvas datu piekļuves atļaujas. DAGR atļauju sarakstu regulāri aktualizē no VIRSIS.

Atļauju sarakstu var aplūkot DAGR portāla sadaļā “VIRSIS atļaujas” vai izgūt ar metodi:

https://dagr.gov.lv/current_user_permissions

Metodes izsaukuma piemērs:

```
curl --location --request GET 'https://dagr.gov.lv/current_user_permissions' \
--header 'Authorization: Bearer {autentifikācijas ietvaros saņemtais drošības talons}'
```

Atbildes piemērs (servisa nosaukums un atļaujas numurs iekrāsoti spilgti sarkanā):

```
[
  {
    "service_name": "vzd_address",
    "permission_number": "1321",
    "purpose_of_use": "Vienošanās pieteikuma datu izmantošanas mērķis (TEST JH)",
    "use_count": 1,
    "use_count_quota": 1321,
    "valid_from": null,
    "valid_to": null,
    "access_control_tree": {
      "name": "vzd_address",
      "read_access": "full",
      "properties": null
    }
  },
  {
    "service_name": "lb_pens_manag_list",
    "permission_number": "29579",
    "purpose_of_use": "TESTS",
  }
]
```

```

    "use_count": 0,
    "use_count_quota": 0,
    "valid_from": "2025-10-15T21:00:00Z",
    "valid_to": "2098-12-31T22:00:00Z",
    "access_control_tree": {
      "name": "lb_pens_manag_list",
      "read_access": "partial",
      "properties": [
        {
          "name": "ListName",
          "read_access": "partial",
          "properties": null
        },
        {
          "name": "List",
          "read_access": "partial",
          "properties": [
            {
              "name": "RegNum",
              "read_access": "partial",
              "properties": null
            },
            {
              "name": "Name",
              "read_access": "partial",
              "properties": null
            },
            {
              "name": "ValidFrom",
              "read_access": "partial",
              "properties": null
            }
          ]
        }
      ]
    },
    "filters": [],
    "search": [
      {
        "name": "pk"
      }
    ]
  }
}
]

```

Atļaujas datu struktūras skaidrojums:

- **service_name** - nosaka DAGR datu izsniegšanas servisu, uz kuru atļauja attiecināta. Servisu saraksts aplūkojams <https://dagr.gov.lv/api> sadaļā "Datu servisi";
- **permission_number** - atļaujas unikālais numurs, kas jānorāda pieprasījuma galvenē "X-DAGR-Permission", izsaucot datu servisu;
- **purpose_of_use** - atļaujas izmantošanas mērķis, kas tiek saglabāts datu pieprasījumu auditā;
- **use_count** - datu servisa izmantoto izsaukuma reižu skaits, kas tiek uzturēts tikai gadījumā, ja atļaujā ir noteikta datu izsaukumu reižu skaita kvota;
- **use_count_quota** - datu izsaukumu reižu skaitu kvota, kas nosaka, cik datu pieprasījumus ar šo atļauju kopumā var veikt;
- **valid_from** - atļaujas sākuma datums. Atļauja nav aktīva pirms šī datuma. Ja vērtība ir null, tad atļaujai nav sākuma datuma ierobežojuma;
- **valid_to** - atļaujas beigu datums. Atļauja nav aktīva pēc šī datuma. Ja vērtība ir null, tad atļaujai nav beigu datuma ierobežojuma;

- **access_control_tree** - koka veida struktūra, kas nosaka datu servisa atbildes datu struktūras datu laukus / hierarhijas elementus, kuriem datu ņēmējam ir piekļuve:
 - **name** - datu lauka vai hierarhijas elementa nosaukums. Pirmajā hierarhijas līmenī sakrīt ar datu kopas nosaukumu;
 - **read_access** - piekļuves līmenis. Iespējamās vērtības - **full**, **partial** un **none**. "**full**" nozīmē, ka datu ņēmējam ir piekļuve datu laukam vai hierarhijas elementam ar visiem apakšelementiem. "**partial**" nozīmē, ka datu ņēmējam ir piekļuve datu laukam vai elementam, bet apakšelementiem katram ir noteikts atsevišķs piekļuves līmenis. "**none**" nozīmē, ka datu ņēmējam nav piekļuves datu laukam vai hierarhijas elementam un tā apakšelementiem. Ja apakšelementa piekļuves līmenis ir pretrunā ar virselementa piekļuves līmeni, tad par noteicošo tiek uzskatīts virselementa piekļuves līmenis.
 - **properties** - saraksts ar nākamā līmeņa ierakstiem, kas atkal var saturēt **name**, **read_access** un **properties** elementus.
 - **filters** - saraksts ar VIRSIS sistēmā reģistrētajiem filtriem;
 - **search** - saraksts ar atļauto meklēšanas atslēgu nosaukumiem. Atslēga ar nosaukumu "pk" nozīmē, ka drīkst meklēt pēc primārās atslēgas. Ja šis saraksts ir tukšs, tad DAGR pieņem, ka meklēt drīkst tikai pēc primārās atslēgas.

4.3.3 Datu objekta izgūšana pēc objekta identifikatora

Visām DAGR izplatāmajām datu kopām ir nodrošināta datu izgūšana pēc objekta identifikatora.

Katrai datu kopai ir savs datu izgūšanas serviss. Katram datu izgūšanas servisam ir savs noteikts objekta identifikators pēc kā ieraksts tiek meklēts. Šis identifikators ir norādīts servisa aprakstā. Par objektu identifikatoriem tiek izmantotas faktiskas biznesa vērtības, piemēram – personas kods, reģistrācijas numurs, dokumenta numurs, adreses kods. Dažos gadījumos tiek izmantoti salikti kodi, kas sastāv no vairākām vērtībām, piemēram, uzņēmumu gada pārskatiem tiek izmantots gads un uzņēmuma reģistrācijas numurs.

Datu izgūšanas pēc objekta identifikatora servisi veidoti kā GET tipa pakalpes ar adresi vienā no divām formām:

- Meklēšanai pēc primārās atslēgas:
https://{dagr_adrese}/{datu_kopa}/{objekta_identifikators}
- Meklēšanai pēc papildus atslēgas:
https://{dagr_adrese}/{datu_kopa}:{atslēga}/{atslēgas_vērtība}

Servisa galvenē "**X-DAGR-Permission**" ir jānorāda VIRSIS atļaujas numurs.

Servisa izsaukuma piemērs – Valsts adrešu reģistra adrese pēc adreses koda:

```
curl --location --request GET 'https://dagr.gov.lv/vzd_address/112022722' \
--header 'Authorization: Bearer {autentifikācijas_ietvaros_saņemtais_drošības_talons}' \
--header 'X-DAGR-Permission: {VIRSIS_atļaujas_numurs}'
```

Atbilde būs pilni objekta dati JSON formātā.

Atbildes piemērs – adreses dati JSON formātā:

```
{
  "ielCode": 100300132,
  "novName": null,
  "zipCode": "LV-1057",
  "dzvName": "18",
  "nltCode": 101848550,
  "code": 112022722,
  "cieName": null,
  "pagAtvk": null,
  "typ": 109,
  "address": "Aglonas iela 31 - 18\nRīga",
  "lksCoordLat": 307682.939,
  "pilAtvk": "0001000",
  "pagName": null,
  "pagCode": null,
  "adr_code": "112022722",
  "dzvCode": 112022722,
  "novAtvk": null,
  "ielName": "Aglonas iela",
  "novCode": null,
  "nltName": "31",
  "irAdrese": true,
  "history": [
    {
      "adrese": "Aglonas iela 31, dz. 18, Rīga"
    }
  ],
  "lksCoordLong": 511254.388,
  "pilCode": 100003003,
  "cieCode": null,
  "pilName": "Rīga"
}
```

Papildus servisa izsaukuma piemērs – UR juridiskā persona pēc reģistrācijas numura:

```
curl --location --request GET 'https://dagr.gov.lv/ur_legal_entity/40001016757' \
--header 'Authorization: Bearer {autentifikācijas ietvaros saņemtais drošības talons}' \
--header 'X-DAGR-Permission: {VIRSIS atļaujas numurs}'
```

Atbilde būs pilni objekta dati JSON formātā.

Papildus: Dažām datu kopām ir iespēja meklēt datus ne tikai pēc datu kopas primārās atslēgas, bet arī citām papildus meklēšanas atslēgām. Lai meklētu pēc papildus meklēšanas atslēgas servisā aiz datu kopas nosaukuma jāliek kols un jānorāda meklēšanas atslēgas nosaukums. Respektīvi, https://dagr.gov.lv/{datu_kopa}:{meklēšanas_atslēga}/{meklējamā_vērtība}

Iespēja meklēt pēc papildus meklēšanas atslēgām iespējama tikai tad, ja datu devējs datu kopai šādas papildus meklēšanas atslēgas ir definējis un tās ir atzīmētas kā pieejamas izmantotajā VIRSIS atļaujā.

4.3.3.1 PMLP datu īpatnības

PMLP fizisko personu datu kopai *fpr_person* ir īpašs PMLP noteikts veids, kā tiek izplatīti dati par personām, kurām ir mainījies personas kods. Ja atlasa personas datus pēc vecā personas koda, tad ierakstā visi lauki izņemot "PersonCode" un "ActualPersonCode" saturēs tukšas (*null*) vērtības. Lai iegūtu personas datus šajos gadījumos, tie ir jāpieprasa par jaunu, izmantojot jauno personas kodu. Vienkāršs veids, kā noteikt šos gadījumus, ir salīdzināt "PersonCode" un "ActualPersonCode" - ja tie atšķiras, tad jāpieprasa dati ar jauno personas kodu.

Izsaukuma piemērs - personas dati pēc vecā personas koda:

```
curl --location 'https://dagr.gov.lv/fpr_person/03109919998' \
--header 'X-DAGR-Permission: 1339' \
--header 'Authorization: .....
```

Atbildes piemērs - personas dati pēc vecā personas koda:

```
{
  "PersonCode": "03109919998",
  "ActualPersonCode": "32777181979",
  "PersonsCode": null,
  "PersonsName": null,
  "PersonsBirth": null,
  "PersonsCitizenship": null,
  "PersonsDocument": null,
  "PersonsPoliticallyRepressedStatus": null,
  "PersonsResidence": null,
  "PersonsResistanceMemberStatus": null,
  "PersonsStatusFPR": null,
  "PersonsDeath": null
}
```

Izsaukuma piemērs - personas dati pēc jaunā personas koda:

```
curl --location 'https://dagr.gov.lv/fpr_person/32777181979' \
--header 'X-DAGR-Permission: 1339' \
--header 'Authorization: .....
```

Atbildes piemērs - personas dati pēc jaunā personas koda:

```
{
  "PersonCode": "32777181979",
  "ActualPersonCode": "32777181979",
  "PersonsCode": [
    {
      "PersonCode": "32777181979",
      "ValidFrom": "2022-08-12T00:00:00+03:00",
      "ValidUntil": null
    },
    {
      "PersonCode": "03109919998",
      "ValidFrom": "1999-10-05T00:00:00+03:00",
      "ValidUntil": "2022-08-12T00:00:00+03:00"
    }
  ],
  "PersonsName": {
```

```

      "LastName": "DAGRE",
      "FirstName": "MAIGA",
      "ReasonName": "LATVIJAS VALSTSPIEDERĪGĀ PIRMUZSKAITE",
      "ReasonCode": "PUZL",
      "ValidFrom": "1999-10-05T00:00:00+03:00",
      "OtherNames": null
    },
    "PersonsBirth": {
      "BirthPlaceCountryCode": "LVA",
      "BirthPlaceFormatted": "LIEPĀJA",
      "BirthDate": "1999-10-03T00:00:00+03:00",
      "IsBornDead": false,
      "ReasonName": "LATVIJAS VALSTSPIEDERĪGĀ PIRMUZSKAITE",
      "BirthPlaceCountryName": "LATVIJA",
      "ValidFrom": "1999-10-05T00:00:00+03:00",
      "ReasonCode": "PUZL",
      "BirthTime": "2024-01-04T15:25:00+02:00"
    },
    "PersonsCitizenship": [
      {
        "CountryCode": "LVA",
        "CountryName": "LATVIJAS",
        "ValidFrom": "1999-10-05T00:00:00+03:00",
        "CitizenshipTypeCode": "P",
        "CitizenshipTypeName": "PILSONIS"
      }
    ],
    "PersonsDocument": [
      {
        "DocumentIssuingCountryCode": "LVA",
        "DocumentKindCode": "PRAP",
        "ReasonName": "OTRA VEIDA DOKUMENTA SAŅEMŠANA",
        "DocumentNumber": "PA4091765",
        "ValidFrom": "2022-08-12T00:00:00+03:00",
        "DocumentIssuingDate": "2022-08-12T00:00:00+03:00",
        "DocumentKindName": "PERSONAS APLIECĪBA",
        "ReasonCode": "OTRADOKS",
        "DocumentIssuingPlace": null,
        "DocumentIssuingCountryName": "LATVIJA",
        "DocumentValidityDate": "2032-08-11T00:00:00+03:00",
        "DocumentIssuingOrganizationName": "PMLP RĪGAS 4.NODAĻA"
      }
    ],
    "PersonsPoliticallyRepressedStatus": null,
    "PersonsResidence": {
      "CountryCode": "LVA",
      "LocalGovernmentCode": "0044000",
      "CountryName": "LATVIJA",
      "ArisCode": "118747094",
      "AddressFormatted": "\"ACONES STACIJA\" - 1, STOPIŅU PAGASTS, ROPAŽU
NOVADS, LV-2130",

```



```

        "ReasonName": "DEKLARĒTĀS DZĪVESVIETAS ZIŅAS",
        "RegionCode": null,
        "ValidFrom": "2003-06-16T00:00:00+03:00",
        "ReasonCode": "DKDZ",
        "TerritorialUnitCode": "0044420"
    },
    "PersonsResistanceMemberStatus": null,
    "PersonsStatusFPR": {
        "StatusValidFrom": "1999-10-05T00:00:00+03:00",
        "StatusValueCode": "1",
        "StatusValueTitle": "ir statuss AKTĪVS"
    },
    "PersonsDeath": null
}

```

4.3.4 Datu izmaiņu izgūšana un pilnas datu kopas pārslaišana

Visām DAGR izplatāmajām datu kopām ir nodrošināta iespēja izgūt datu izmaiņas.

Izmaiņu izgūšanas servisi veidoti kā GET RESTful tipa pakalpes ar adresi šādā formā:

https://dagr.gov.lv/{datu_kopa}_changes

Servisa galvenē “X-DAGR-Permission” ir jānorāda VIRSIS atļaujas numurs.

Šiem servisiem ir trīs opcionāli parametri:

- **offset** – izmaiņu ieraksta kārtas numurs no kura sākt izmaiņu atlasī. Var izmantot arī negatīvus skaitļus, tādā gadījumā serviss atgriezīs norādītā skaita pēdējos ierakstus. Piemēram, norādot -2, tiks atgriezti divi pēdējie ieraksti;
- **timestamp** – datums un laiks no kura sākt izmaiņu atlasī;
- **limit** – atbildē atgriežamo ierakstu skaits. Noklusētā vērtība ir 100.

Ja netiek norādīts ne *offset*, ne *timestamp* parametrs, tad serviss atgriezīs vienu pēdējo izmaiņu ierakstu.

Izmaiņu servisa atbilde satur izmaiņu ierakstus JSON formātā, kur katram ierakstam ir šāda struktūra:

```

{
  "offset": izmaiņu_kārtas_numurs,
  "value": {
    ieraksta_dati_kā_JSON_objekts
  },
  "key": "ieraksta_identifikators",
  "timestamp": "izmaiņu_pievienošanas_datums_un_laiks"
}

```

BŪTISKI: Datu kopas izmaiņu rinda satur tikai informāciju par pēdējo reizi kad konkrētais ieraksts ir mainīts. Līdz ar to izmaiņu servisu var izmantot visu datu kopas **ierakstu atklāšanai** un **aktuālo izmaiņu monitoringam**, bet to nevar izmantot **vēstures datu atlasei**. Gadījumos, kur datu kopa satur vēsturisku izmaiņu datus, tās ir iekļautas ierakstu datu struktūrā. Kā blakus efekts, izmaiņu kārtas numuri var izkrist, bet izgūstot izmaiņu datus tie vienmēr tiks atgriezti augošā secībā.

Datu izmaiņu izgūšanas pielietojumi:

- **Ierakstu atklāšana.** Lai atklātu visus datu kopas datu ierakstus ir vispirms jāizsauc izmaiņu serviss bez *offset* vai *timestamp* parametriem. Tad jānolasa atbilde un jāpiefiksē *offset* vērtība pēdējam saņemtajam izmaiņu ierakstam. Tālāk jāveido nākamais pieprasījums *offset* parametrā norādot piefiksēto pēdējo nolasīto izmaiņu kārtas numuru. Šādā veidā iterējot cauri visiem izmaiņu ierakstiem, līdz tiek saņemta tukša atbilde, var atklāt visus datu kopas ierakstus.
- **Izmaiņu monitorings.** Lai sekotu aktīvajām izmaiņām, var izmantot izmaiņu *offset* vai *timestamp* vērtības. Sistēmā, kas monitorē datu kopu, ir jāizveido process, kas ar noteiktu regularitāti pieprasa datus atbilstošajam DAGR izmaiņu servisam. Pēc datu nolasīšanas sistēmā, kas monitorē datu kopu, ir jāpiefiksē pēdējo nolasīto izmaiņu *offset* vai *timestamp*, lai pie nākamās datu nolasīšanas reizes tiktu saņemtas tikai jaunākās izmaiņas.

Izmaiņu servisa izsaukuma piemērs:

```
curl --location 'https://dagr.gov.lv/ur_legal_entity_changes?offset=0&limit=100' \
--header 'Authorization: Bearer {autentifikācijas ietvaros saņemtais drošības talons}' \
--header 'X-DAGR-Permission: {VIRSIS atļaujas numurs}'
```

Izmaiņu servisa atbildes piemērs:

```
[{
  "key": "40900000031",
  "offset": 0,
  "timestamp": "2023-03-20T09:53:03.031",
  "value": {
    "address": {
      "addressComplete": "\"Striķu skola\", Zvārdes pag., Saldus nov., LV-3883",
      "addressRegisterCode": 106245206,
      "county": "Saldus nov.",
      "houseName": "Striķu skola",
      "parish": "Zvārdes pag.",
      "postalCode": "LV-3883"
    },
    "cleanedShortName": "STRIKU SAKUMSKOLA",
    "isAnnulled": false,
    "lastModifiedAt": "2022-03-24T09:14:53.615Z",
    "links": {
      "self": "/legal-entity/40900000031"
    },
    "liquidations": [],
    "name": "Striķu sākumskola",
  }
}]
```

```

"officers": [
  {
    "appointedOn": "2001-07-05",
    "id": 100010293,
    "isAnnulled": false,
    "lastModifiedAt": "2019-10-31T14:12:00.114Z",
    "naturalPerson": {
      "forename": "Daina",
      "latvianIdentityNumber": "071063-12857",
      "surname": "Zandberga"
    },
    "position": "GENERAL_MANAGER",
    "positionDetails": "Testādes direktore",
    "registeredOn": "2019-10-31"
  },
],
"publicAuthorityDetails": {
  "contactDetails": {
    "email": "striku.sakumskola@saldus.lv",
    "phone": "+37163846132",
    "website": "http://strikuskola.saldus.lv/"
  },
  "delegatedEntities": [],
  "establishingAct": {
    "date": "2016-01-28",
    "legislatorName": "Saldus novada dome",
    "legislatorNumber": "90009114646",
    "number": "1.7.",
    "title": "Saldus novada domes lēmums",
    "type": "DECISION"
  },
  "registrationDetails": {
    "authorityType": "INSTITUTION_OF_INDIRECT_ADMINISTRATION",
    "establishedOn": "2016-01-28",
    "higherAuthorityCurrentName": "Saldus novada pašvaldība",
    "higherAuthorityNumber": "90009114646",
    "isIndependentTaxpayer": false,
    "isInstitutionOfHigherEducation": false,
    "subordinationType": "CONTROL",
    "taxpayerUnitNumber": "90009189123"
  },
  "schoolBranches": []
},
"register": "PUBLIC_PERSONS_INSTITUTIONS",
"registeredOn": "2018-03-08",
"registrationNumber": "40900000031",
"registrationNumberAssignedOn": "2018-03-08",
"reorganizations": [],
"status": "REGISTERED",
"type": "PUBLIC_BODY"
}

```

```
}  
]
```

PMLP datu īpatnības

PMLP datu kopai *fpr_person* izmaiņas uz DAGR tiek nodotas nevis uzreiz visam ierakstam, bet pa daļām. Līdz ar to izmaiņu datiem ir pielāgots serviss */fpr_person_all_changes*, kur izmaiņu dati ir sadalīti pa biznesa notikumiem jeb datu blokiem. Katram notikumam ir sava atslēga formā "{personas_kods}/{notikuma_veids}".

Izsaukuma piemērs:

```
curl --location 'https://dagr.gov.lv/fpr_person_all_changes?limit=1000&timestamp=2024-10-07T08%3A17%3A33.985Z' \  
--header 'X-DAGR-Permission: 23091' \  
--header 'Authorization: .....'
```

Atbildes piemērs:

```
[  
  {  
    "key": "14097314007/ActualPersonCode",  
    "offset": 47008638,  
    "timestamp": "2024-10-07T10:17:31.631Z",  
    "value": {  
      "PersonCode": "14097314007",  
      "ActualPersonCode": "32190664656"  
    }  
  },  
  {  
    "key": "14097314007/PersonsCode",  
    "offset": 47008639,  
    "timestamp": "2024-10-07T10:17:31.655Z",  
    "value": {  
      "PersonCode": "14097314007",  
      "PersonsCode": null  
    }  
  },  
  {  
    "key": "14097314007/PersonsName",  
    "offset": 47008640,  
    "timestamp": "2024-10-07T10:17:31.664Z",  
    "value": {  
      "PersonCode": "14097314007",  
      "PersonsName": null  
    }  
  },  
  {  
    "key": "14097314007/PersonsBirth",  
    "offset": 47008641,  
    "timestamp": "2024-10-07T10:17:31.670Z",  
    "value": {  
      "PersonCode": "14097314007",  
      "PersonsBirth": null  
    }  
  },  
  {  
    "key": "14097314007/PersonsCitizenship",  
    "offset": 47008642,  
    "timestamp": "2024-10-07T10:17:31.677Z",  
    "value": {  
      "PersonCode": "14097314007",  
      "PersonsCitizenship": null  
    }  
  }  
]
```

```

    },
    {
      "key": "13087215007/PersonsDocument",
      "offset": 47008643,
      "timestamp": "2024-10-07T10:17:31.689Z",
      "value": {
        "PersonCode": "13087215007",
        "PersonsDocument": null
      }
    },
    {
      "key": "14097314007/PersonsPoliticallyRepressedStatus",
      "offset": 47008644,
      "timestamp": "2024-10-07T10:17:31.690Z",
      "value": {
        "PersonCode": "14097314007",
        "PersonsPoliticallyRepressedStatus": null
      }
    },
    {
      "key": "140973140077/PersonsResidence",
      "offset": 47008645,
      "timestamp": "2024-10-07T10:17:31.694Z",
      "value": {
        "PersonCode": "14097314007",
        "PersonsResidence": null
      }
    },
    {
      "key": "14097314007/PersonsResistanceMemberStatus",
      "offset": 47008646,
      "timestamp": "2024-10-07T10:17:31.696Z",
      "value": {
        "PersonCode": "14097314007",
        "PersonsResistanceMemberStatus": null
      }
    },
    {
      "key": "14097314007/PersonsStatusFPR",
      "offset": 47008647,
      "timestamp": "2024-10-07T10:17:31.702Z",
      "value": {
        "PersonCode": "14097314007",
        "PersonsStatusFPR": null
      }
    },
    {
      "key": "14097314007/PersonsDeath",
      "offset": 47008648,
      "timestamp": "2024-10-07T10:17:31.703Z",
      "value": {
        "PersonCode": "14097314007",
        "PersonsDeath": null
      }
    },
    {
      "key": "32180763556/PersonsCode",
      "offset": 47008649,
      "timestamp": "2024-10-07T10:17:31.705Z",
      "value": {
        "PersonCode": "32180763556",
        "PersonsCode": [
          {
            "ValidFrom": "2024-10-07T00:00:00+03:00",
            "ValidUntil": null,
            "PersonCode": "32180763556"
          },
          {
            "ValidFrom": "1989-10-01T00:00:00+03:00",
            "ValidUntil": "2024-10-07T00:00:00+03:00",

```

```

        "PersonCode": "14097314007"
    }
}
],
{
    "key": "32180763556/ActualPersonCode",
    "offset": 47008650,
    "timestamp": "2024-10-07T10:17:31.705Z",
    "value": {
        "PersonCode": "32180763556",
        "ActualPersonCode": "32180763556"
    }
},
{
    "key": "32180763556/PersonsName",
    "offset": 47008651,
    "timestamp": "2024-10-07T10:17:33.195Z",
    "value": {
        "PersonCode": "32180763556",
        "PersonsName": {
            "ReasonCode": "PUZL",
            "ReasonName": "LATVIJAS VALSTSPIEDERĪGĀ PIRMUZSKAITE",
            "ValidFrom": "1988-10-01T00:00:00+03:00",
            "FirstName": "JURIS",
            "LastName": "KAULIŅŠ",
            "OtherNames": null
        }
    }
},
{
    "key": "32180763556/PersonsBirth",
    "offset": 47008652,
    "timestamp": "2024-10-07T10:17:34.894Z",
    "value": {
        "PersonCode": "32180763556",
        "PersonsBirth": {
            "ReasonCode": "PUZL",
            "ReasonName": "LATVIJAS VALSTSPIEDERĪGĀ PIRMUZSKAITE",
            "ValidFrom": "1989-10-01T00:00:00+03:00",
            "BirthDate": "1973-08-13T00:00:00+03:00",
            "BirthTime": null,
            "IsBornDead": false,
            "BirthPlaceCountryCode": "LVA",
            "BirthPlaceCountryName": "LATVIJA",
            "BirthPlaceFormatted": "RĪGA"
        }
    }
},
{
    "key": "32180763556/PersonsDocument",
    "offset": 47008653,
    "timestamp": "2024-10-07T10:17:41.019Z",
    "value": {
        "PersonCode": "32180763556",
        "PersonsDocument": [
            {
                "ReasonCode": "PUZL",
                "ReasonName": "LATVIJAS VALSTSPIEDERĪGĀ PIRMUZSKAITE",
                "ValidFrom": "1989-10-01T00:00:00+03:00",
                "DocumentKindCode": "LNPS",
                "DocumentKindName": "NEPILSONA PASE",
                "DocumentIssuingDate": "1989-08-14T00:00:00+03:00",
                "DocumentNumber": "LN000007",
                "DocumentIssuingCountryCode": "XXX",
                "DocumentIssuingCountryName": "PSRS",
                "DocumentIssuingOrganizationName": null,
                "DocumentIssuingPlace": "2. PASU DAĻA",
                "DocumentValidityDate": null
            }
        ]
    }
}
]

```

```

    },
    {
      "key": "32180763556/PersonsCitizenship",
      "offset": 47008654,
      "timestamp": "2024-10-07T10:17:42.713Z",
      "value": {
        "PersonCode": "32180763556",
        "PersonsCitizenship": [
          {
            "ValidFrom": "1989-10-01T00:00:00+03:00",
            "CountryCode": "LVA",
            "CountryName": "LATVIJAS",
            "CitizenshipTypeCode": "N",
            "CitizenshipTypeName": "NEPILSONIS"
          }
        ]
      }
    },
    {
      "key": "32180763556/PersonsStatusFPR",
      "offset": 47008655,
      "timestamp": "2024-10-07T10:17:44.508Z",
      "value": {
        "PersonCode": "32180763556",
        "PersonsStatusFPR": {
          "PersonCode": "32180763556",
          "StatusValueCode": "2",
          "StatusValueTitle": "ir statuss PASĪVS",
          "StatusValidFrom": "2024-10-05T00:00:00+03:00"
        }
      }
    }
  ]
}

```

Šajā piemērā redzamas izmaiņas, kur personai tiek mainīts personas kods. Piemērs balstīts uz faktiskiem datiem, personu identificējošās daļas aizstājot ar iedomātām vērtībām.

4.3.5 Monitorings

Atšķirībā no izmaiņu servisiem, kas ļauj sekot visu ierakstu izmaiņām, monitorings ļauj sekot specifisku datu saņēmēja noteiktu ierakstu izmaiņām datu kopā. Monitoringam ir nepieciešama VIRSIS monitoringa atļauja.

Lai uzsāktu monitoringu, ir jāizveido **monitoringa kopa** jeb jānosaka monitoringa kopas konfigurācija un jāpievieno monitoringa kopai saraksts ar identifikatoriem jeb ierakstiem, kuri tiks monitorēti. Identifikatoru saraksta izmēram nav ierobežojumu.

DAGR par monitoringa kopai pievienotajiem identifikatoriem uzkrāj izmaiņu pazīmes un izmaiņu datus monitoringa kopas konfigurācijā noteiktajā apjomā. Izmaiņu pazīmes un izmaiņu dati tiek uzkrāti no brīža, kad ieraksta identifikators tika pievienots monitoringa kopai, līdz brīdim kad tas izņemts no monitoringa kopas. Datu saņēmējiem ir iespēja nolasīt uzkrātās izmaiņu pazīmes, izmaiņu datus vai aktuālos datus par izmainītajiem ierakstiem atkarībā no VIRSIS monitoringa atļaujā noteiktā.

4.3.5.1 Kopu konfigurācija

Monitoringa kopas var izveidot DAGR pašapkalpošanās portālā vai izmantojot DAGR REST API. Monitoringa kopas konfigurācija satur šādu informāciju:

- **Datu kopa (obligāts).** Nosaukums DAGR datu kopai, kuras datu izmaiņas tiks monitorētas;

- **Datu saņēmējs (obligāts).** Datu saņēmēja, kas definējis monitoringa kopu, reģistrācijas numurs;
- **Atļauja (obligāts).** VIRSIS monitoringa atļaujas numurs;
- **Monitoringa kopas identifikators (obligāts).** Nemainīgs monitoringa kopas identifikators, kas tiek automātiski ģenerēts no monitoringa kopas nosaukuma un datu saņēmēja reģistrācijas numura;
- **Nosaukums (obligāts).** Datu saņēmēja brīvi izvēlēts monitoringa kopas nosaukums. Nosaukumā pieļaujami latīņu burti, cipari un zemsvītra. Nosaukumam jābūvē ar burtu un tas var būt maksimums 20 zīmes garš. Ja nav norādīts tiek ģenerēts automātiski no datu kopas nosaukuma un datu saņēmēja reģistrācijas numura;
- **Apraksts (neobligāts).** Monitoringa kopas apraksts, kas tiek izmantots DAGR portāla ekrānformās;
- **Monitorējamie datu lauki (obligāts).** Hierarhisks saraksts ar datu kopas datu laukiem, izmaiņas kuros ierosina monitoringa notikumu. Katrs lauks rakstāms jaunā rindā, apakšlīmeņiem atkāpe. Var norādīt tikai VIRSIS monitoringa atļaujā iekļautos monitorējamos datu laukus. Ja sarakstu atstāj tukšu, tad DAGR monitoringa kopas monitorējamos datu laukus aizpilda no VIRSIS atļaujā norādītajiem monitorējamiem datu laukiem.
- **Izgūstamie datu lauki (neobligāts).** Hierarhisks saraksts ar datu kopas datu laukiem, kurus datu saņēmējs par katru monitoringa notikumu var izgūt. Katrs lauks rakstāms jaunā rindā, apakšlīmeņiem atkāpe. Var norādīt tikai VIRSIS atļaujā iekļautos izgūstamos datu laukus. Ja atstāj tukšu, tad izgūstamie datu lauki netiks saglabāti un ar servisiem datu saņēmējs varēs nolasīt tikai izmaiņu pazīmes, pat ja VIRSIS atļauja saturēs izgūstamos datu laukus.

Pēc monitoringa kopas izveides tās konfigurācijai var mainīt tikai apraksta lauku.

Monitoringa kopa ir aktīva VIRSIS atļaujas derīguma termiņa laikā. Termiņa beigu pienākšanas brīdī monitoringa kopa tiek automātiski arhivēta. Trīs mēnešus pēc arhivēšanas brīža monitoringa kopa tiek automātiski dzēsta. Datu saņēmējam ir iespēja arhivēt un dzēst datu kopas arī manuāli pirms termiņa.

Arhivētās monitoringa kopas ir pieejamas monitoringa kopu sarakstā, tām var izgūt datu saņēmēja pievienoto identifikatoru sarakstu, bet nevar vairs izgūt izmaiņu pazīmes un izmaiņu datus.

Dzēstas monitoringa kopas monitoringa kopu sarakstā vairs nav pieejamas.

4.3.5.1.1 Konfigurācija portālā

Monitoringa kopu konfigurācija DAGR portālā notiek sadaļā "Patērēšana" -> "Monitoringa kopas".

Šajā sadaļā ir pieejams saraksts ar to datu saņēmēju monitoringa kopām, kuriem lietotājs ir reģistrēts

kā kontaktpersona:

DAGR > Monitoringa kopas

DAGR

Pieteikumu pārvaldība

Savienojumi

Iestāžu lietotāji

Zināšanu bāze

Monitorings

Audita notikumi

Paziņojumi lietotājiem

Izplatīšana

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Pamata vide (dataset)

Anonimizācijas uzdevumi

Patērēšana

VIRSIS atļaujas

Monitoringa kopas

Pielāgotie servisi

Monitoringa kopas

Meklējamais teksts

Statistika

Izveidot jaunu kopu

Valsts digitālās attīstības aģentūra

90001733697

PTTEST_demo

Atļauja: 30425

Identifikators: 90001733697_PTTEST_demo

VIRSIS tiesību demonstrācija

PTTEST_csv_failsV_piektaDU11

PTTEST_demo_v2

Atļauja: 30425

Identifikators: 90001733697_PTTEST_demo_v2

VIRSIS tiesību demonstrācijas MK

PTTEST_csv_failsV_piektaDU11

UNISO

40003562863

Demo_kopa_1

Atļauja: 40003562863/vbtai

Identifikators: 40003562863_vbtai_orphan_certificate_Demo_kopa_1

vbtai_orphan_certificate

Demo_kopa_2

Atļauja: 40003562863/vbtai

Identifikators: 40003562863_Demo_kopa_2

Vai varu rediģēt

vbtai_orphan_certificate

Demo_kopa_3

Atļauja: 40003562863/vbtai

Identifikators: 40003562863_vbtai_orphan_certificate_Demo_kopa_3

vbtai_orphan_certificate

krex_pex_slashes

Atļauja: atļauja 40003562863/sopa_needy/0024000

Identifikators: 40003562863_sopa_needy/0024000_krex_pex_slashes

sopa_needy/0024000

porquito_mas

Atļauja: multi_permission_test_01

Identifikators: 40003562863_porquito_mas

fpr_person

Postman_pmip_test

Atļauja: multi_permission_test_01

fpr_person

Jaunas monitoringa kopas izveide no monitoringa kopu saraksta uzsākama nospiežot pogu "Izveidot jaunu kopu", kas atver monitoringa kopas izveides ekrānformu:

DAGR > Monitoringa kopas > Jauna kopa

DAGR

Pieteikumu pārvaldība

Savienojumi

Iestāžu lietotāji

Zināšanu bāze

Monitorings

Audita notikumi

Paziņojumi lietotājiem

Izplatīšana

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Pamata vide (dataset)

Anonimizācijas uzdevumi

Patērēšana

VIRSIS atļaujas

Monitoringa kopas

Pielāgotie servisi

Jauna monitoringa kopa

Monitoringa kopas kartiņa, detalizētāku aprakstu skatiet mūsu [wiki](#).

NOSAUKUMS *

Jauna_kopa

Tehniskais kopas nosaukums — pieļaujami latīņu burti, cipari un zemsvītra. Jāsākas ar burtu, maksimums 20 zīmes garš. Atstājot tukšu, tiks uzģenerēts.

APRAKSTS

Jauna monitoringa kopa

Kopas paplašinātais nosaukums

DATU SAŅĒMĒJS

Valsts digitālās attīstības aģentūra

DATU KOPA

PTTEST_csv_failsV_piektaDU11

VIRSIS ATĻAUJAS NUMURS

30425

MONITORĒJAMIE DATU LAUKI

```
model
construction_year
serial_number
registered_on
aircraft_model_category
```

IZCŪSTAMIE DATU LAUKI

```
model
construction_year
serial_number
```

Ielādēt lauku sarakstu

Hierarhisks lauku saraksts, kuri ierosina monitoringa notikumu. Katrs jaunā rindā, apakšlīmenim atkāpe.

Hierarhisks lauku saraksts, kurus saglabāt par notikumu. Katrs jaunā rindā, apakšlīmenim atkāpe.

Izveidot monitoringa kopu

[Atgriezties pie saraksta](#)

Monitoringa kopas apskate savukārt iespējama uzspiežot uz kādas no monitoringa kopām monitoringa kopu sarakstā, kas atver monitoringa kopas apskates ekrānformu:

DAGR > Monitoringa kopas > UNISO > 40003562863_Postman_pmlp_test

Monitoringa kopa Postman_pmlp_test Monitorējamie identifikatori Rediģēt aprakstu

UNISO 40003562863

Monitoringa kopas kartiņa, detalizētāku aprakstu skatiet mūsu [wiki](#).

IDENTIFIKATORS	40003562863_Postman_pmlp_test
NOSAUKUMS	Postman_pmlp_test
APRAKSTS	PMLP kopas apraksts
DATU KOPA	fpr_person
VIRSIS ATĻAUJAS NUMURS	multi_permission_test_01
MONITORĒJAMIE DATU LAUKI	PersonCode PersonsName FirstName LastName
IZGŪSTAMIE DATU LAUKI	PersonCode PersonsName FirstName LastName PersonsBirth BirthDate

[Atgriezties pie saraksta](#) Arhivēt kopu Dzēst kopu

Monitoringa kopas apskates ekrānforma satur iespēju arhivēt monitoringa kopu, dzēst monitoringa kopu, kā arī pārvaldīt monitoringa kopas identifikatoru sarakstu.

4.3.5.1.2 Konfigurācija ar REST API

Monitoringa servisu apraksti OpenAPI standartā pieejami adresē: `{dagr_adrese}/api-docs/monitoring`. Servisi izsaukami ar datu saņēmēja drošības talonu. Attiecībā uz monitoringa kopu konfigurāciju pieejami šādi servisi:

- **POST /monitoring_set Monitoringa kopas konfigurācijas izveide.** Serviss atbildē atgriež monitoringa kopas identifikatoru, bet saturā (*body*) tam jāiesūta JSON ar šādu struktūru:

```
{
  "registration_number": "string",
  "dataset": "string",
  "name": "string",
  "description": "string",
  "permission_number": "string",
  "trigger_fields": "string",
  "stored_fields": "string"
}
```

- **registration_number:** Datu saņēmēja reģistrācijas numurs, kuram monitoringa kopa pieder
- **dataset:** Nosaukums datu kopai, kuras ieraksti tiek monitorēti
- **name:** Monitoringa kopas īsais nosaukums (kods), unikāls datu saņēmēja monitoringa kopu ietvaros. Ja nav norādīts, tiks ģenerēts.
- **description:** Monitoringa kopas pilns nosaukums, kas tiek izmantots DAGR portāla ekrānformās.
- **permission_number:** VIRSIS atļauja izgūt datus pēc identifikatora no norādītās datu kopas

- **trigger_fields:** Monitorējamie datu lauki. Hierarhisks lauku saraksts, kuri ierosina monitoringa notikumu. Katrs jaunā rindā, apakšlīmenim atkāpe. Ja iesūta *null*, tad tiks paņemts no VIRSIS atļaujā norādītajiem monitorējamiem datu laukiem.
- **stored_fields:** Izgūstamie datu lauki. Hierarhisks lauku saraksts, kurus saglabāt par notikumu. Katrs jaunā rindā, apakšlīmenim atkāpe. Ja iesūta *null*, tad monitoringa kopas ietavaros varēs izgūt tikai izmaiņu pazīmes, bet ne datus.
- GET /monitoring_set/{monitoring_set_id} **Monitoringa kopas konfigurācijas apskate** monitoringa kopai ar norādīto identifikatoru. Šis serviss ir izsaukams arī ar datu devēja drošības talonu tām monitoringa kopām, kuras monitorē konkrētā datu devēja datu kopas.
- PUT /monitoring_set/{monitoring_set_id} **Monitoringa kopas konfigurācijas rediģēšana** monitoringa kopai ar norādīto identifikatoru. Rediģējams ir tikai monitoringa kopas nosaukums. Serviss izsaukams saturā (*body*) iesūta JSON ar šādu struktūru:


```
{
  "description": "string"
}
```

 - **description:** Monitoringa kopas pilns nosaukums, kas tiek izmantots DAGR portāla ekrānformās.
- DELETE /monitoring_set/{monitoring_set_id} **Monitoringa kopas konfigurācijas dzēšana** monitoringa kopai ar norādīto identifikatoru.
- POST /terminate_monitoring_set/{monitoring_set_id} **Monitoringa kopas arhivēšana** monitoringa kopai ar norādīto identifikatoru.
- GET /monitoring_set **Datu saņēmēja definēto monitoringa kopu saraksts.** Šis serviss ir izsaukams arī ar datu devēja drošības talonu - šādā gadījumā tiks attēlotas visas monitoringa kopas, kas attiecināmas uz datu devēja datu kopām.

4.3.5.2 Identifikatoru saraksts

Katrai monitoringa kopai tiek uzturēts aktīvo identifikatoru saraksts. Šim sarakstam var pievienot (iekļaut), pārbaudīt, dzēst (izklaut) monitorējamo ierakstu identifikatorus. Identifikatoriem ir jābūt tieši tādiem, kādi tie tiek izmantoti monitorējamā datu kopā.

Papildus aktīvo identifikatoru sarakstam, DAGR uztur arī identifikatoru pievienošanas un izklaušanas vēsturi, kas var tikt izmantota lai noteiktu kuros laika posmos kuri identifikatori ir tikuši monitorēti.

Pēc noklusējuma monitorēt var tikai tādus ierakstus, kas ir pamata datu kopā. Piemēram, fizisko personu reģistram varētu monitorēt tikai tādas personas, kas faktiski ir šajā reģistrā. Taču datu kopām konfigurācijā var norādīt, ka monitorējami ir arī ieraksti, kas vēl nav pamata datu kopā.

Darbības ar monitorējamo ierakstu identifikatoriem var izdarīt gan DAGR pašapkalpošanās portālā, gan izmantojot DAGR REST API.

4.3.5.2.1 Identifikatoru pārvaldība portālā

DAGR pašapkalpošanās portālā identifikatoru saraksts pieejams monitoringa kopas kartīnā nospiežot pogu "Monitorējamie ieraksti", kas atver monitorējamo ierakstu sarakstu:

DAGR > Monitoringa kopas > UNISO > 40003562863_Postman_pmlp_test > Identifikatori

DAGR

Pieteikumu pārvaldība

Savienojumi

Iestāžu lietotāji

Zināšanu bāze

Monitoringa kopas

Audita notikumi

Paziņojumi lietotājiem

Izplatīšana

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Pamata vide (dataset)

Anonimizācijas uzdevumi

Patērēšana

VIRSIS atļaujas

Monitorējamie ieraksti

Pielāgotie servisi

Monitoringa kopa Postman_pmlp_test

UNISO 40003562863

Monitorējamā atslēga

Uzsākt Pārbaudīt Pārtraukt

Pēdējās izmaiņas

Pārtraukts	19018519995	24.02.2026. 14:02:28
Uzsākts	32158761835	24.02.2026. 14:02:14
Uzsākts	19044919992	24.02.2026. 14:02:03
Uzsākts	03109919998	24.02.2026. 14:01:37
Uzsākts	19018519995	24.02.2026. 14:01:26
Uzsākts	30018410604	24.02.2026. 14:00:59
Pārtraukts	30018410604	24.02.2026. 14:00:56
Uzsākts	30018410604	24.02.2026. 14:00:52
Pārtraukts	30018410604	24.02.2026. 14:00:49
Uzsākts	30018410604	3.12.2025. 14:56:38

Lejupielāde

JSON CSV Aktuālo identifikatoru saraksts

JSON CSV Izmaiņu vēsture

[Atgriezties pie saraksta](#)

Nemot vērā, ka monitorējamo identifikatoru var būt ļoti daudz, kartīnā tiek attēloti tikai pēdējie 20. Kartīnā var ievadīt monitorējamā ieraksta identifikatoru un izmantojot pogas vai nu uzsākt tā monitorēšanu, pārtraukt monitorēšanu vai pārbaudīt identifikatora esamību aktīvo identifikatoru sarakstā.

Ja nepieciešams apstrādāt visu identifikatoru sarakstu, tad gan aktīvos identifikatorus, gan identifikatoru izmaiņu vēsturi iespējams lejupielādēt JSON un CSV formātos.

4.3.5.2.2 Identifikatoru pārvaldība ar REST API

Monitoringa servisu apraksti OpenAPI standartā pieejami adresē: {dagr_adrese}/api-docs/monitoring. Servisi izsaucami ar datu saņēmēja drošības talonu. Attiecībā uz identifikatoru pārvaldību pieejami šādi servisi:

- **PUT /monitor/{monitoring_set_id}/{key}** **Pievienot (iekļaut) vienu identifikatoru.** Pievienojamais identifikators norādāms servisa adresē. Serviss atbild ar standarta HTTP statusiem, kļūdas gadījumā norādot problēmas cēloni.
- **POST /monitor/{monitoring_set_id}** **Pievienot (iekļaut) vairākus identifikatorus.** Serviss ļauj pievienot līdz 1000 identifikatoriem ar vienu pieprasījumu. Pievienojamo identifikatoru saraksts padodams pieprasījuma saturā (*body*) JSON formātā. Piemēram šādi:

```
[
  "100344302",
  "546455632",
  "119414704",
  "103301112",
  "119383626",
  "104967291"
]
```

Servisa atbildē tiks atgriezts saraksts ar identifikatoriem, kurus neizdevās pievienot. Piemēram, šādi:

```
[
  {
    "key": "546455632",
    "status": 400,
    "details": "Ierakstu nevar pievienot, jo tas nav atrasts datubāzē. Šai
    monitoringa kopai ir atļauts pievienot tikai eksistējošus ierakstus"
  }
]
```

- **DELETE/monitor/{monitoring_set_id}/{key}** **Dzēst (izklāut) identifikatoru.** Izklājamais identifikators norādāms servisa adresē. Serviss atbild ar standarta HTTP statusiem, kļūdas gadījumā norādot problēmas cēloni.
- **POST /unmonitor/{monitoring_set_id}** **Dzēst (izklāut) vairākus identifikatorus.** Serviss ļauj izklāut līdz 1000 identifikatoriem ar vienu pieprasījumu. Izklājamu identifikatoru saraksts padodams pieprasījuma saturā (*body*) JSON formātā. Piemēram šādi:

```
[
  "100344302",
  "546455632",
  "321312332"
]
```

Servisa atbildē tiks atgriezts saraksts ar identifikatoriem, kurus neizdevās izklāut. Piemēram, šādi:

```
[
  {
    "key": "546455632",
    "status": 400,
    "details": "Ieraksts šobrīd netiek monitorēts, tāpēc to nav iespējams dzēst"
  },
  {
    "key": "321312332",
    "status": 400,
    "details": "Ieraksts šobrīd netiek monitorēts, tāpēc to nav iespējams dzēst"
  }
]
```

- **GET /monitored_keys/{monitoring_set_id}** **Izgūt aktīvo identifikatoru sarakstu.** Serviss atgriež norādītās monitoringa kopas aktīvo identifikatoru sarakstu, kur katram identifikatoram redzams arī pievienošanas datums un laiks. Atbildes piemērs:

```
[
  {
    "key": "119414704",
    "since": "2026-02-24T12:46:37.993Z"
  },
  {
    "key": "119609516",
    "since": "2025-12-04T12:45:21.054Z"
  }
]
```

- **GET /monitored_keys_changes/{monitoring_set_id}** **Izgūt identifikatoru saraksta izmaiņas.** Serviss ļauj izgūt pilnu vai daļēju identifikatoru izmaiņu vēsturi un tam ir šādi parametri:
 - **"offset"** kārtas numurs izmaiņai no kuras sākt atlasīt. Ja nav norādīts ne offset, ne timestamp, tad atgriezīs pēdējo ierakstu. Ja offset norādīts kā negatīvs skaitlis, tad aprēķinās no beigām. Piemēram, offset = -20 atgriezīs pēdējos 20 ierakstus.
 - **"timestamp"** datumlaiks izmaiņai no kuras sākt atlasīt. Ja norādīts kopā ar offset, tad tiks izmantots offset.
 - **"timestamp_to"** datumlaiks izmaiņai līdz kurai atlasīt.
 - **"limit"** ierakstu skaits ko iekļaut atbildē. Pēc noklusējuma 100.

Servisa atbildē tiek atgriezti pievienošanas un izklaušanas notikumi, kur par katru notikumu redzams "key" jeb identifikators, "offset" jeb izmaiņas kārtas numurs, "timestamp" jeb izmaiņas pievienošanas datums laiks un "value", kas satur vērtības - "on" pievienošanai un "off" izklaušanai.

```
[
  {
    "key": "119609516",
    "offset": 15,
    "timestamp": "2025-12-04T12:45:21.034Z",
    "value": "on"
  },
  {
    "key": "104141723",
    "offset": 16,
    "timestamp": "2025-12-08T14:44:48.168Z",
    "value": "off"
  }
]
```

- **GET /monitoring_is_on/{monitoring_set_id}/{key}** **Pārbaudīt identifikatoru.** Pārbaudāmais identifikators norādāms servisa adresē. Ja identifikators nebūs atrasts aktīvo identifikatoru sarakstā, tad atbildē būs HTTP statusa kods 404. Savukārt, ja identifikators būs atrasts, tad tiks atgriezts monitoringa kopas nosaukums, saņēmēja reģistrācijas numurs, datu kopas nosaukums, identifikators un tā iekļaušanas datums. Atbildes piemērs:

```
{
  "monitoring_set_name": "Testa_kopa",
  "registration_number": "90001733697",
  "dataset": "vzd_address",
  "key": "104967291",
  "since": "2026-02-24T12:46:38.214Z"
}
```

4.3.5.3 Izmaiņu izgūšana

DAGR katrai monitoringa kopai reģistrē izmaiņu notikumus pie monitorējamās datu kopas ierakstu izmaiņām, kurām izpildās šādi nosacījumi:

- Ieraksta identifikators ir iekļauts monitoringa kopas aktīvo identifikatoru sarakstā;
- Ierakstam ir mainījusies vērtība kaut vienā datu laukā no datu laukiem, kas norādīti kā monitorējamie datu lauki monitoringa kopas konfigurācijā. Izmaiņas iekļauj ne tikai vērtību maiņu, bet arī ierakstu izveidi un dzēšanu.

Katrā izmaiņu notikumā tiek fiksēta šāda informācija:

- **Ieraksta identifikators** (key) ierakstam kuram mainījušies dati;
- **Izmaiņu notikuma datums un laiks** (timestamp);
- **Izmaiņu notikuma kārtas numurs** (offset);
- **Izmaiņu pazīmes** (changes_overview) jeb informācija par to kuros datu laukos no monitoringa kopas konfigurācijā norādītajiem monitorējamiem datu laukiem ir mainījušās vērtības;
- **Jaunās vērtības** (value) jeb vērtības uz kurām mainījušies dati monitoringa kopas konfigurācijā norādītajos izgūstamajos datu laukos;
- **Iepriekšējās vērtības** (previous) jeb vērtības no kurām mainījušies dati monitoringa kopas konfigurācijā norādītajos izgūstamajos datu laukos.

Datu saņēmējiem ir iespēja nolasīt monitoringa notikumu izmaiņu pazīmes un atkarībā no VIRSIS atļaujas arī:

- monitoringa notikumu jaunās vērtības;

- monitoringa notikumu iepriekšējās vērtības;
- ar monitoringa notikumu saistītā ieraksta aktuālās vērtības.

4.3.5.3.1 Izmaiņu nolasīšana ar REST API

Monitoringa notikumu nolasīšana

Monitoringa notikumu izmaiņu pazīmju, jauno vērtību un iepriekšējo vērtību nolasīšanai izmantojami datu kopu izmaiņu servisi, kas veidoti kā GET tipa pakalpes ar adresi formā: `https://{dagr_vides_adrese}/{datu_kopa}_changes`

Servisa galvenē "X-DAGR-Permission" ir jānorāda VIRSIS monitoringa atļaujas numurs, servisa galvenē "Authorization" ir jānorāda datu saņēmēja drošības talons un ir jānorāda šādi parametri:

- **"monitoring_set"** monitoringa kopas nosaukums. Obligāti norādāms, lai izgūtu monitoringa notikumus. Ja netiek norādīts, tad serviss darbosies visu notikumu atrādīšanas režīmā;
- **"offset"** jeb monitoringa notikuma kārtas numurs. Norāda sākot ar kuru kārtas numuru fiksētie monitoringa notikumi atrādāmi. Ja norādīts gan parametrs "offset", gan parametrs "timestamp", tad tiek izmantots "offset". Ja nav norādīts ne "offset", ne "timestamp", tad tiek atgriezta pēdējā izmaiņa. Var norādīt negatīvas vērtības, lai skaitītu no beigu gala. Piemēram "offset = -2", nozīmē, ka atbildē atgriežami monitoring notikumi sākot no priekšpēdējā;
- **"timestamp"** jeb monitoringa notikuma datums. Norāda kopš kura datuma un laika fiksētie monitoringa notikumi atlasāmi. Ja sākuma datums un laiks nav norādīts, tad atlasa tikai informāciju par pēdējo monitoringa notikumu;
- **"timestamp_to"** jeb monitoringa notikumu beigu datums. Norāda līdz kuram datumam un laikam fiksētie monitoringa notikumi atlasāmi;
- **"limit"** jeb atbildē iekļaujamo ierakstu skaits. Neobligāts parametrs, kura noklusētā vērtība ir 100. Ja tiek norādīts gan "limit", gan "timestamp_to", tad tiek izmantots "timestamp_to"
- **"data_fields"** jeb atribūtu koka filtrs. Neobligāts parametrs. Ļauj norādīt kuri datu lauki un kuros hierarhijas līmeņos no monitoringa kopas izgūstamajiem datu laukiem ir iekļaujami servisa atbildes jauno vērtību un veco vērtību sadaļās. Ja parametrs nav norādīts, tad atbildē tiek iekļauti visi izgūstamie datu lauki;
- **"changes_fields"** jeb izmaiņu pazīmju filtrs. Neobligāts parametrs. Ļauj norādīt kuriem datu laukiem un kuros hierarhijas līmeņos jāparādās izmaiņu pazīmēs, lai monitoringa notikums tiktu iekļauts servisa atbildē. Respektīvi, filtrs ļauj no visiem atlasīt tikai interesējošos izmaiņu notikumus.
- **"with_changes_overview"** neobligāta pazīme (jā/nē), kas nosaka vai servisa atbildē atgriezt izmaiņu pazīmes - pēc noklusējuma "jā";
- **"with_value"** neobligāta pazīme (jā/nē), kas nosaka vai serviss atbildē atgriež jaunās datu lauku vērtības. Pēc noklusējuma "nē". Iespēja pieejama tikai tad, ja VIRSIS atļauja to paredz;
- **"with_previous"** neobligāta pazīme (jā/nē), kas nosaka vai serviss atbildē atgriež iepriekšējās datu lauku vērtības. Pēc noklusējuma "nē". Iespēja pieejama tikai tad, ja VIRSIS atļauja to paredz.

Aktuālo datu nolasīšana

Ja VIRSIS monitoringa atļauja paredz mainīto ierakstu aktuālo vērtību nolasīšanu, tad tas darāms izmantojot ierakstu nolasīšanas pēc identifikatora servisu, kas veidoti kā GET tipa pakalpes ar adresi formā: `https://{dagr_vides_adrese}/{datu_kopa}/{ieraksta_identifikators}`

Servisa galvenē "X-DAGR-Permission" ir jānorāda VIRSIS monitoringa atļaujas numurs, servisa galvenē "Authorization" ir jānorāda datu saņēmēja drošības talons un ir jānorāda šādi parametri:

- **"monitoring_set"** monitoringa kopas nosaukums. Obligāti norādāms, lai izgūtu aktuālos datus monitoringa atļaujas ietvaros. Ja netiek norādīts, tad serviss darbosies pamata atļaujas ietvaros;
- **"data_fields"** jeb atribūtu koka filtrs. Neobligāts parametrs. Ļauj norādīt kuri datu lauki un kuros hierarhijas līmeņos no monitoringa kopas izgūstamajiem datu laukiem ir iekļaujami servisa atbildē. Ja parametrs nav norādīts, tad atbildē tiek iekļauti visi izgūstamie datu lauki.

Būtiski: Aktuālo datu nolasīšanas serviss, kas tiek darbināts ar monitoringu saistīto izmaiņu nolasīšanas režīmā, izsniedz datus tikai par ierakstiem ar identifikatoriem, kas iekļauti norādītās monitoringa kopas aktīvo identifikatoru sarakstā.

4.3.5.3.2 Izmaiņu izgūšanas scenāriji ar piemēriem

Personas datu monitorēšana ar monitoringa notikumu izgūšanu

Scenārijs piemērojams gadījumos kur tiek monitorēti ieraksti datu kopai, kas satur personas datus, un VIRSIS atļauja paredz izmainīto vērtību izgūšanu no monitoringa notikumiem. Apstrādājot personas datus ir būtiski:

- pārliecināties, ka pastāv pamats datus apstrādāt. Tamdēļ process sadalīts divās daļās - vispirms tiek izgūtas izmaiņu pazīmes, lai atsijātu tikai nepieciešamās izmaiņas. Pēc tam tiek atlasīti izmaiņu dati tikai par atsijātajām izmaiņām;
- izgūt datus minimālajā nepieciešamajā apjomā. Tamdēļ procesā lietojami filtri, kas atgriež tikai interesējošos monitoringa notikumus un tikai apstrādei nepieciešamos izmaiņu datus, ja VIRSIS atļauja un monitoringa kopas konfigurācija pieļauj plašākas iespējas.

Solis 1 - izmaiņu pazīmju izgūšana. Izgūstam izmaiņu pazīmes

izmantojot `https://{dagr_vides_adrese}/{datu_kopa}_changes` servisu ar šādiem parametriem:

- **"monitoring_set"** - monitoringa kopas nosaukums;
- **"offset"** - kārtas numurs monitoringa notikumam sākot ar kuru nolasīt izmaiņu pazīmes. Pirmajā atlasīšanas reizē var norādīt 0. Nākošajās aktualizācijas reizēs izmantojams offset, kas ir par vienu lielāks nekā iepriekšējā aktualizācijas reizē pēdējā nolasītā monitoringa notikuma kārtas numurs;
- **"changes_fields"** - norādāmi tie datu lauki ar hierarhijas līmeņiem, kuru izmaiņas apstrādājamas konkrētā procesa ietvaros. Var nenorādīt, ja monitoringa kopas konfigurācijā starp monitorējamiem laukiem nav tādu, kuru izmaiņas konkrētajam procesam nav jāapstrādā.

Izsaukuma piemērs:

```
curl --location
'https://dagr.gov.lv/vzd_address_changes?monitoring_set=Postman_testa_kopa&offset=13&changes_fields=address%2C%20code%2C%20metadata' \
--header 'X-DAGR-Permission: 90001733697/vzd_address' \
--header 'Authorization: .....'
```

Atbildes piemērs:

```
[
  {
    "key": "119383626",
    "offset": 13,
    "timestamp": "2025-12-05T14:59:54.381Z",
    "changes_overview": "address\ncode"
```

```

    },
    {
      "key": "119609516",
      "offset": 15,
      "timestamp": "2025-12-09T16:23:45.479Z",
      "changes_overview": "address"
    },
    {
      "key": "119414704",
      "offset": 16,
      "timestamp": "2026-01-15T15:59:43.173Z",
      "changes_overview": "address"
    }
  ]

```

Solis 2 - izmaiņu datu izgūšana. Izgūstam izmaiņu datus katram iepriekšējā solī atlasītajam ierakstam izsaucot https://{dagr_vides_adrese}/{datu_kopa}_changes servisu ar šādiem parametriem:

- "monitoring_set" - monitoringa kopas nosaukums;
- "offset" - iepriekšējā solī atlasītā ieraksta kārtas numurs;
- "limit = 1" - norādam 1, lai atlasītu tikai interesējošo ierakstu;
- "data_fields" - norādam interesējošos datu laukus ar to hierarhijas līmeņiem. Var nenorādīt, ja monitoringa kopas konfigurācijā starp izgūstamajiem datu laukiem nav tādu, kas konkrētajam procesam nav nepieciešami.
- "with_value" - norādam "true";
- "with_changes_overview" - norādam "false".

Izsaukuma piemērs:

```

curl --location
'https://dagr.gov.lv/vzd_address_changes?monitoring_set=Postman_testa_kopa&offset=13&with_
changes_overview=false&with_value=true&limit=1&data_fields=adr_code%2CpilName' \
--header 'X-DAGR-Permission: 90001733697/vzd_address' \
--header 'Authorization: ..... '

```

Atbildes piemērs:

```

[
  {
    "key": "119383626",
    "offset": 13,
    "timestamp": "2025-12-05T14:59:54.381Z",
    "value": {
      "adr_code": "119383626",
      "pilName": "Rīga"
    }
  }
]

```

Personas datu monitorēšana ar aktuālo datu izgūšanu

Scenārijs piemērojams gadījumos kur tiek monitorēti ieraksti datu kopai, kas satur personas datus, un VIRSIS atļauja paredz aktuālo datu izgūšanu. Tāpat kā iepriekšējā scenārijā, arī šeit apstrādājot personas datus ir būtiski:

- pārliecināties, ka pastāv pamats datus apstrādāt. Tamdēļ process sadalīts divās daļās - vispirms tiek izgūtas izmaiņu pazīmes, lai atsijātu tikai nepieciešamās izmaiņas. Pēc tam tiek atlasīti aktuālie dati izmainītajiem ierakstiem;
- izgūt datus minimālajā nepieciešamajā apjomā. Tamdēļ procesā lietojami filtri, kas atgriež tikai interesējošos monitoringa notikumus un tikai apstrādei nepieciešamos izmaiņu datus, ja VIRSIS atļauja un monitoringa kopas konfigurācija pieļauj plašākas iespējas.

Solis 1 - izmaiņu pazīmju izgūšana. Izgūstam izmaiņu pazīmes

izmantojot `https://{dagr_vides_adrese}/{datu_kopa}_changes` servisu ar šādiem parametriem:

- "monitoring_set" - monitoringa kopas nosaukums;
- "offset" - kārtas numurs monitornga notikumam sākot ar kuru nolasīt izmaiņu pazīmes. Pirmajā atlasīšanas reizē var norādīt 0. Nākošajās aktualizācijas reizēs izmantojams offset, kas ir par vienu lielāks nekā iepriekšējā aktualizācijas reizē pēdējā nolasītā monitoringa notikuma kārtas numurs;
- "changes_fields" - norādāmi tie datu lauki ar hierarhijas līmeņiem, kuru izmaiņas apstrādājamas konkrētā procesa ietvaros. Var nenorādīt, ja monitoringa kopas konfigurācijā starp monitorējamiem laukiem nav tādu, kuru izmaiņas konkrētajam procesam nav jāapstrādā.

Skatīt iepriekšējā scanārija piemērus, jo pirmais solis sakrīt.

Solis 2 - aktuālo datu izgūšana. Izgūstam aktuālos datus katram iepriekšējā solī atlasītajam ierakstam izsaucot `https://{dagr_vides_adrese}/{datu_kopa}/{ieraksta_identifikators}` servisu ar šādiem parametriem:

- "monitoring_set" - monitoringa kopas nosaukums;
- "ieraksta_identifikators" - iepriekšējā solī atlasītā ieraksta identifikators;
- "data_fields" - norādam interesējošos datu laukus ar to hierarhijas līmeņiem. Var nenorādīt, ja monitoringa kopas konfigurācijā starp izgūstamajiem datu laukiem nav tādu, kas konkrētajam procesam nav nepieciešami.

Izsaukuma piemērs:

TIKS PIEVIENOTS PĒC IZSTRĀDES

Atbildes piemērs:

TIKS PIEVIENOTS PĒC IZSTRĀDES

Datu monitorēšana vispārīgā gadījumā

Ja monitorējamie dati nesatur personas datus vai arī monitoringa konfigurācija izveidota tā, ka tā ietver tikai konkrētajam procesam nepieciešamos monitorējamus datu laukus un izgūstamos datu laukus, monitoringa datu nolasīšanu var veikt vienā solī

izmantojot `https://{dagr_vides_adrese}/{datu_kopa}_changes` servisu ar šādiem parametriem:

- "monitoring_set" - monitoringa kopas nosaukums;
- "offset" - kārtas numurs monitoringa notikumam sākot ar kuru nolasīt izmaiņu pazīmes. Pirmajā atlasīšanas reizē var norādīt 0. Nākošajās aktualizācijas reizēs izmantojams offset, kas ir par vienu lielāks nekā iepriekšējā aktualizācijas reizē pēdējā nolasītā monitoringa notikuma kārtas numurs;
- "with_value" norādāma vērtība "true".

Izsaukuma piemērs:

```
curl --location  
'https://dev.dagr.gov.lv/vzd_address_changes?monitoring_set=Postman_testa_kopa&with_value=  
true&offset=13' \
```

```
--header 'X-DAGR-Permission: 90001733697/vzd_address' \  
--header 'Authorization: .....'
```

Atbildes piemērs:

```
[  
  {  
    "key": "119383626",  
    "offset": 13,  
    "timestamp": "2025-12-05T14:59:54.381Z",  
    "value": {  
      "adr_code": "119383626",  
      "address": "Strēlnieku iela 4B - 40\nRīga",  
      "code": 119383626,  
      "cieName": null,  
      "pilName": "Rīga"  
    },  
    "changes_overview": "address\ncode"  
  },  
  {  
    "key": "119609516",  
    "offset": 15,  
    "timestamp": "2025-12-09T16:23:45.479Z",  
    "value": {  
      "adr_code": "119609516",  
      "address": "Aleksandra Čaka iela 78B - 6\nRīga",  
      "code": 119609516,  
      "cieName": null,  
      "pilName": "Rīga"  
    },  
    "changes_overview": "address"  
  },  
  {  
    "key": "119414704",  
    "offset": 16,  
    "timestamp": "2026-01-15T15:59:43.173Z",  
    "value": {  
      "adr_code": "119414704",  
      "address": "Silakroga iela 11 - 12\nSilakrogs\nRopažu pag.\nRopažu nov.",  
      "code": 119414704,  
      "cieName": "Silakrogs",  
      "pilName": null  
    },  
    "changes_overview": "address"  
  }  
]
```

4.3.5.4 Monitoringa statistika

Datu devējiem ir pieejama monitoringa statistika par datu saņēmēju veiktajiem monitoringa pieprasījumiem par savām datu kopām. Statistika pieejam DAGR portāla atverot sadaļu "Patērēšana" -> "Monitoringa kopas" un nospiežot pogu "Statistika".

Alternatīvi statistikas datus datu saņēmēji var izgūt izmantojot šādus servissus:

- **GET /dwhm_monitoring_requests** Saraksts: Monitoringa pieprasījumu kopskaits norādītajā periodā. Izsaucot servisu iespējams norādīt šādus parametrus:
 - request_hour_from: Perioda sākums (no). Datuma laika lauks.
 - request_hour_to: Perioda beigas (līdz). Datuma laika lauks.
 - service: Izsauktais monitoringa serviss. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visiem servisiem;
 - dataset_name: Datu kopa. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visām datu devēja datu kopām;
 - permission_number: Atļaujas numurs. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visām atļaujām, kas izniegtas uz datu devēja datu kopām.

- `GET /dwhm_monitoring_requests_by_hour` Saraksts: Monitoringa pieprasījumu skaits pa stundām norādītajā periodā. Izsaucot servisu iespējams norādīt šādus parametrus:
 - `request_hour_from`: Perioda sākums (no). Datuma laika lauks.
 - `request_hour_to`: Perioda beigas (līdz). Datuma laika lauks.
 - `service`: Izsauktais monitoringa serviss. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visiem servisiem;
 - `dataset_name`: Datu kopa. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visām datu devēja datu kopām;
 - `permission_number`: Atļaujas numurs. Neobligāts lauks. Ja nenorāda, tad atgriež informāciju par visām atļaujām, kas izniegtas uz datu devēja datu kopām.

4.3.5.5 Papildus norādes par monitoringu

Monitorējamie un izgūstamie datu lauki. Ņemot vērā, ka monitorējamie un izgūstamie datu lauki tiek konfigurēti atsevišķi, var rasties interesantas situācijas:

- Ja datu lauks ir starp monitorējamajiem datu laukiem, bet nav starp izgūstamajiem datu laukiem, tad monitoringa ļaus uzzināt tikai faktu, ka šajā datu laukā mainījusies vērtība, bet ne pašu vērtību. Ja nav neviens izgūstamais datu lauks, tad monitoringa ietvaros var tikai uzzināt izmaiņu faktu, bet ne izmaiņu datus;
- Ja datu lauks ir starp izgūstamajiem datu laukiem, bet nav starp monitorējamajiem datu laukiem, tad šī lauka vērtību izmaiņu brīdī varēs noskaidrot, bet izmaiņa šī lauka vērtībā pati par sevi monitoringa notikuma fiksēšanu neierosinās.

Par sekošanu līdz monitoringa datiem: DAGR neatzīmē, kuri monitoringa notikumi jau ir nolasīti. Sekošana līdz monitoringa notikumu nolasīšanas stāvoklim ir realizējama datu saņēmēja pusē. Tas ir viegli izdarāms, izmantojot monitoringa notikumu nolasīšanas servisa parametru `offset` vai `timestamp`. Ja datu saņēmējam savā informācijas sistēmā ir process, kas ar noteiktu regularitāti pieprasa monitoringa datus, tad pēc datu nolasīšanas ir jā saglabā pēdējā nolasītā ieraksta `timestamp` vai `offset` vērtība. Šo saglabāto vērtību nākošajā procesa izpildes ciklā jāpado monitoringa notikumu izgūšanas servisam kā ieejas parametru un tas garantēs, ka nolasītas tiek tikai jaunpienākušās izmaiņas.

4.3.6 Pielāgoti servisi

Datu izgūšana no DAGR iespējama arī definējot un izmantojot pielāgotus servisi. Pielāgoti servisi ļauj precizēt izgūstamo datu struktūru, fiksēt izmantojamās atļaujas un atlasīt datus uzreiz no vairākām datu kopām.

Pielāgota servisa definēšana. Pielāgotus servisius datu ņēmējs var apskatīt un definēt pašapkalpošanās portāla sadaļā “Patērēšana” → “Pielāgotie servisi”.

Pielāgotie servisi

 Jauns serviss

Tests Martins

90000074738 Jelgavas valstspilsētas pašvaldības iestāde "Jelgavas izglītības pārvalde"

Tests_Martins_T

90000074738 Jelgavas valstspilsētas pašvaldības iestāde "Jelgavas izglītības pārvalde"

LBIS_TAPIS_TEST_FPR	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
Testa serviss	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
Testa_serviss	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
demo_custom_test	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
demo_test	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
demo_test2	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
PT_TEST_serviss2	90001733697 Valsts digitālās attīstības aģentūra
PT_TEST_serviss3	90001733697 Valsts digitālās attīstības aģentūra
tests_VIIS_person_education	90001840100 Lauksaimniecības datu centrs
pmlp_piel_test	90000054163 Pilsonības un migrācijas lietu pārvalde
DEMO_AKC	40003562863 Sabiedrība ar ierobežotu atbildību "UNISO"
DEMO_TEST	40003562863 Sabiedrība ar ierobežotu atbildību "UNISO"
PT_TEST_serviss	40003562863 Sabiedrība ar ierobežotu atbildību "UNISO"
Testa servissPT2	90000069830 Centrālā statistikas pārvalde

Pielāgota servisa definīcija satur:

- **Servisa nosaukumu.** Ņemot vērā, ka nosaukums tiks izmantots servisa izsaukuma ceļā to vēlams veidot īsu un bez atstarpēm vai citiem speciālajiem simboliem.
- **Reģistrācijas numuru.** Datu ņēmēja reģistrācijas numurs, kuram serviss tiek veidots. Ņēmēja nosaukums pēc reģistrācijas numura tiek pielasīts automātiski.
- **Atlases parametrus.** Saraksts ar servisa atlases parametriem, kas tiks izmantoti servisa izsaukuma ceļā. Šajā sarakstās jānorāda visi parametri, kas tālāk parādās kā atslēgas atlasāmajās datu kopās.
- **Datu kopas.** Pielāgotajam servisam var norādīt vienu vai vairākas pamata datu kopas no kurām datus izgūst. Katrai datu kopai jānorāda:
 - **Pamata servisa ceļš.** Var norādīt jebkuru DAGR API pamata datu servisu, kurš aprakstīts <https://dagr.gov.lv/api>. Ja pamata servisa ceļš satur atlases parametrus, tie jāiekļauj pielāgotā servisa parametru sarakstā.
 - **Atļaujas numurs.** Katram pamata datu servisam jānorāda saistītās VIRSIS atļaujas numurs.
 - **Datu struktūra.** Pēc izvēles datu kopām var norādīt arī atgriežamo datu struktūru. Šo iespēju var izmantot, lai sašaurinātu atgriežamo datu lauku skaitu no tiem datu laukiem, kas datu ņēmējam pieejami izmantotās atļaujas ietvaros.

Pielāgotais serviss Extra_serviss

SERVISA NOSAUKUMS: Extra_serviss

REĢISTRĀCIJAS NUMURS: 90000068977

PATERĒTĀJS: Rīgas Tehniskā universitāte

PARAMETRI

PATH_PARAMETER: person_code

PATH_PARAMETER: registrationNumber

DATU KOPAS

Datu kopa 1

URL_TEMPLATE: /fpr_person/{person_code}

PERMISSION_NUMBER: 1339

Datu kopa 2

URL_TEMPLATE: /ur_legal_entity/{registrationNumber}

PERMISSION_NUMBER: VIRSIS_UR_DEFAULT

Servisa izmantošana. DAGR pašapkalpošanās portālā definētie pielāgotie servisi datu ņēmējiem pieejami DAGR API ar šādu ceļu:

https://dagr.gov.lv/custom/{pielagota_servisa_nosaukums}/{pielagota_servisa_parametri}

4.3.7 Pēdējo izmaiņu datums

Ja datu ņēmējam ir atļauja veikt izmaiņu izgūšanu (skatīt **Error! Reference source not found. Error! Reference source not found.**), tad pēdējo datu aktualizācijas datumu un laiku var noteikt izsaucot izmaiņu servisu bez parametriem – tas atgriezīs pēdējo izmainīto ierakstu, izmaiņu kārtas numuru un izmaiņu datumu un laiku.

Ja datu ņēmējam ir atļauja izgūt datus tikai pēc ieraksta identifikatora, tad pēdējo izmaiņu datumu un laiku var uzzināt izmantojot palīg-servisu:

https://dagr.gov.lv/updates_last_timestamp/{datu_kopas_nosaukums}

Tāpat kā datu izgūšanas servisiem arī šim servisam jāpievieno drošības talons un galvenē (*header*) ar nosaukumu “X-DAGR-Permission” ir jānorāda atbilstošās datu atļaujas numurs.

Servisa izsaukuma piemērs:

```
curl --location
'https://dagr.gov.lv/updates_last_timestamp/fpr_person' \
--header 'X-DAGR-Permission: 1339' \
--header 'Authorization: .....'
```

Atbildes piemērs:

```
[
  {
    "key": "328681-81130",
    "offset": 3551,
    "timestamp": "2024-04-23T07:02:01.093Z"
  }
]
```

4.4 Datu izgūšana ar datu saņēmēja savienotāju

Datu saņēmēja savienotājs ir JAVA aplikācija, kas izgūst datus no DAGR REST API un uztur to aktuālu kopiju datu saņēmēja datu bāzē izmantojot datu saņēmējam piešķirtu datu izmaiņu lasīšanas atļauju. Datu saņēmēja savienotājs ir izvietojams datu saņēmēja infrastruktūrā, lai nebūtu jāatver piekļuve datu bāzei ārpus datu saņēmēja tīkla infrastruktūras un datu saņēmējam būtu pilna kontrole pār datu saņēmēja savienotāju un tā konfigurāciju.

Datu saņēmēja savienotāja sekmīgai darbināšanai ir nepieciešamas šādas datnes:

- **dagr-connector.jar** jeb savienotāja aplikācija. Savienotāja jaunākā versija vienmēr pieejama:
 - Produkcijas vidē (stabilā versija): <https://dagr.gov.lv/download/dagr-connector>
 - Testa vidē (testa versija): <https://test.dagr.gov.lv/download/dagr-connector>
- **application.conf** jeb konfigurācijas fails, kas satur informāciju par pieslēgumu DAGR kodolam (datu saņēmēja savienojuma identifikators, privātā atslēga, DAGR kodola adreses), datu bāzes pieslēguma informācija, ceļu uz datu kartējuma definīciju, savienotāja darbināšanas biežumu u.c. konfigurācijas parametrus.
- **connector.yaml** jeb datu struktūras kartējuma fails, kas nosaka, kā dati no nolasāmās datu kopas datu struktūras transformējami uz datu bāzes tabulām datu bāzei, kur šie dati ierakstāmi.
- **logback.xml** jeb žurnālēšanas uzstādījumu fails, kas definē datu saņēmēja savienotāja žurnāla pierakstu veidošanas apjomu.

4.4.1 Savienojumu reģistrēšana

Lai izmantotu datu saņēmēja savienotāju ir jāreģistrē gan datu saņēmēja savienojums (skatīt sadaļu “4.2.1 Datu saņēmēja savienojuma reģistrēšana”), gan datu devēja savienojums (skatīt sadaļu “5.5 Datu devēja savienojuma reģistrēšana”). Datu saņēmēja savienojums nepieciešams datu izgūšanai, savukārt datu devēja savienojums nepieciešams, lai datu saņēmēja savienotājs uz DAGR varētu nodot informāciju par izmantoto savienotāja konfigurāciju un pēdējiem notikumiem.

4.4.2 Savienotāja konfigurācija

4.4.2.1 Vispārīgā konfigurācija (*application.conf*)

```
#datu kopu (servisu) nosaukumi DAGR sistēmā
dataset_names = vzd_address

#savienotāja tips darbībai datu saņemšanas režīmā
datasource_type = jdbc_export

#DAGR serveris no kura dati izgūstami
```



```

target_server_uri          = "https://dagr.gov.lv"

#DAGR datu devēja pieslēguma ID un atslēga
provider.key               = "???"
provider.client_id         = "UNISO_jdbc_export_1"

#DAGR datu saņēmēja pieslēguma ID un atslēga
consumer.key               = "???"
consumer.client_id         = "UNISO_test_connector_1"

#DAGR REST API serviss datu kopas izmaiņu nolasīšanai
consumer_service_uri       = "
https://dagr.gov.lv/vzd_address_changes"

#Atļaujas numurs datu izmaiņu izgūšanai no datu kopas
consumer_permission_number = "34235325"

#Drošības talona izgūšanas adrese un audience
jwt.service.uri            =
"https://dagr.gov.lv/jwt/connect/token"
jwt.payload-audience      = "urn:LNB:DAGR.STS/resources"

#norāde uz datu struktūras kartējuma konfigurācijas failu
metadata_file              = "./configs/jdbcExport/connector.yaml"

#chron izteiksme, kas nosaka cik bieži veicama datu sinhronizācija
pekko.quartz.schedules.connector.expression = "*/10 * * ? * *"

input-files = "/tmp"

#JDBC pieslēgums datu bāzei kurā dati ierakstāmi
jdbc {
  data {
    driverClassName = "org.postgresql.Driver"
    jdbcUrl         = "jdbc:postgresql://localhost:5555/???"
    username        = "???"
    password        = "???"
    autoCommit      = false
    readOnly        = false
    maximumPoolSize = 1
  }
}

#Datu validācija obligāti izslēdzama datu saņemšanas režīmā
data.validation=off

#Datu ierakstīšanas papildus konfigurācija
export {
  root_elements_batch_size = 100
  root_dagr_key_field = dagr_key    #japaredz extra lauks DAGR
  identifikācijas key glabāšanai (string, not null, indexed)
}

```

```

initial_params {
  offset = 0
  limit = 10000
}
common_id_seq_query = "select nextval('seq');"
#max_id_query_from_root_table = #Opcionāls: pēc noklusējuma
'select coalesce(max(id),0) from <root_table>;'
child_table_id_seq_query {
  address_history = "select nextval('seq');"
  #address_metadata = "select nextval('seq');"
}
}

```

Datu ierakstīšanas datu saņēmēja datu bāzē (eksporta) konfigurācijas zem export{} key skaidrojums:

- **export.root_elements_batch_size** = ik pēc uzrādītā skaita tiek saukts connection.commit (ja kļūda connection.rollback). Tas attiecas uz pamattabulas elementu skaitu. Ja ir daudz bērnu tabulu un mazbērnu tabulu, tad batch jātur salīdzinoši mazs.
- **export.root_dagr_key_field = dagr_key** lauka vārds, kam jābūt uzrādītam exporta struktūrā (connector.yaml) pamata tabulas līmenī. Laukā tiks glabāta DAGR datu kopas ieraksta primārā atslēga, kas ir tiek izmantota lai noteiktu vai ieraksti pievienojami, pārrakstāmi vai dzēšami. Optimālai veikspējai sagaidāms, ka šis DB lauks ir indeksēts.
- **export.initial_params.offset offsets** no kura sāk ielādi. Bet konektors šo neņem vērā, ja DB jau ir ieraksti, tad offset vērtība tiek noteikt pēc max(id) [pamattabulā id laukā tiek glabāts offset]
- **export.initial_params.limit** - ierakstu skaits, ko straumē no DAGR change lista. Ja konektoram uzliek schedule ik pa 10 sekundēm, un limit-u, piem., 10000, tad tā ir batch norādīšana saņemšanai elementiem.
- **export.common_id_seq_query** - query bērnu tabulu id vērtību noteikšanā
- **export.max_id_query_from_root_table** - query pamata tabulas lielākās id vērtības noteikšanai. Neobligāts, pēc noklusējuma 'select coalesce(max(id),0) from <root_table>;'. Savienotājs šo vērtību izmanto, lai noteiktu cik tālu dati ar DAGR sinhronizēti.
- **export.child_table_id_seq_query.<bērnu tabulas vārds>** - individuāls query bērnu tabulas id vērtības noteikšanā. Ja nav uzrādīts, ņem export.common_id_seq_query

4.4.2.2 Datu kartējuma konfigurācija (connector.yaml)

Datu kartējums darbinot savienotāju datu saņemšanas jeb *jdbc_export* režīmā satur konfigurāciju kā DAGR datu kopas dati izkārtojami datu saņēmēja datu bāzes tabulās. Ievērojama līdzīga sintakse kā datu nodošanas no DB režīmam, kas ir aprakstīta [datu izplatīšanas sadalā](#).

Taču ir jāievēro dažas prasības attiecībā uz datu bāzes struktūru:

- datu saņēmējam pamata datu tabulā jāuztur lauks DAGR ierakstu primārajām atslēgām. Šis lauks ir atzīmēts vispārējās konfigurācijas (application.conf) parametrā export.root_dagr_key_field un pēc noklusējuma tam nosaukums ir dagr_key;
- visām datu tabulām jāsaturs primārie identifikatori ar nosaukumu "id" - gan pašā datu bāzē, gan kartējumā (connector.yaml failā).

Papildus, uzmanība pievēršama tam, ka datu lauki ir case sensitive.

Piemēra vērtības:

```
#sinhronizējamās datu kopas struktūra, kurai pievienots dagr_key
datu lauks
name:  vzd_address
table: vzd_address
fields:
  - id
  - adr_code
  - address
  - code
  - cieCode
  - cieName
  - dzvCode
  - dzvName
  - ielCode
  - ielName
  - lksCoordLong
  - lksCoordLat
  - nltCode
  - nltName
  - novCode
  - novName
  - pagCode
  - pagName
  - pilCode
  - pilName
  - typ
  - pilAtvk
  - novAtvk
  - pagAtvk
  - zipCode
  - irAdrese
  - dagr_key
  - history *:
    table: address_history
    fields:
      - id
      - vzd_address.id
      - adrese
  - metadata * :
    table: address_metadata
    fields:
      - id
      - vzd_address.id
      - adreses_veids
      - autors
      - datums
      - veids
      - numurs
```

```

- nosaukums

#-----
#definīcijas tabulām, kurās ierakstāmi dati

table: vzd_address
columns:
  - id
  - adr_code                : Adreses kods (indekss)
  - address                 : Adrese
  - code                    int    : Adreses kods
  - cieCode                 long   : Ciema / mazciema kods
  - cieName                 : Ciema / mazciema nosaukums
  - dzvCode                 long   : Telpu grupas kods
  - dzvName                 : Telpu grupa
  - ielCode                 long   : Ielas kods
  - ielName                 : Ielas nosaukums
  - lksCoordLong            decimal : LKS ģeogrāfiskais garums
  - lksCoordLat             decimal : LKS ģeogrāfiskais platums
  - nltCode                 long   : Ēkas, apbūvei paredzēta zemes
vienības (NLT) kods
  - nltName                 : Ēka, apbūvei paredzēta zemes vienība
(NLT)
  - novCode                 long   : Novada kods
  - novName                 : Novada nosaukums
  - pagCode                 long   : Pagasta kods
  - pagName                 : Pagasta nosaukums
  - pilCode                 long   : Pilsētas kods
  - pilName                 : Pilsētas nosaukums
  - typ                     long   : Adresācijas objekta tips
  - pilAtvk                 : Pilsētas ATVK kods
  - novAtvk                 : Novada ATVK kods
  - pagAtvk                 : Pagasta ATVK kods
  - zipCode                 : Pasta indekss
  - irAdrese                boolean : Ir adrese (citas adreses neatsaucas
uz šo adreses ierakstu)
  - dagr_key                ! : Ieraksta identifikācijas atslēga
DAGR
pk: id
idx:
  - dagr_key

table: address_history
columns:
  - id
  - vzd_address.id
  - adrese                  : Vēsturiskā adrese
pk: id
idx:
  - vzd_address_id
refs:

```

```

- vzd_address_id -> vzd_address(id) on delete cascade

table: address_metadata
columns:
  - id
  - vzd_address_id
  - adreses_veids      : Adreses veids
  - autors              : Autors
  - datums              date : Datums
  - veids               : Veids
  - numurs              : Numurs
  - nosaukums           : Nosaukums
pk: id
idx:
  - vzd_address_id
refs:
  - vzd_address_id -> vzd_address(id) on delete cascade

```

4.4.3 Datu bāzes sagatavošana

Datu bāzē ir jāizveido tabulas, kas atbilst *connector.yaml* konfigurācijā norādītajām tabulām. Ņemot vērā, ka savienotājs datu nodošanai uz datu bāzi izmanto JDBC interfeisu, jānodrošina, ka:

- lietotājs, kas norādīts *application.conf* konfigurācijas `jdbc.data.{}` sadaļā, spēs piekļūt visām *connector.yaml* konfigurācijā norādītajām tabulām (SELECT, INSERT, DELETE);
- lietotājs, kas norādīts *application.conf* konfigurācijas `jdbc.data.{}` sadaļā, spēs izpildīt autonumerācijas sekvenses.

Ja ieslēdz savienotāja žurnālēšanu debug režīmā izmantojot `logback.xml`, tad startējot konektoru tiks izdrukāti sql skripti datu bāzes izveidei PostgreSQL sintaksē;

Ja ieslēdz savienotāja žurnālēšanu debug režīmā izmantojot `logback.xml`, tad startējot konektoru tiks izdrukāta `dataset_name` norādītā sanitizēta DAGR struktūra YAML formātā.

To var darīt manuāli, vai alternatīvi, ja ieslēdz savienotāja žurnālēšanai loggeri `"dagr.jdbc.sql"` debug režīmā izmantojot `logback.xml`, tad startējot konektoru tiks izdrukāti SQL skripti datu bāzes izveidei uz PostgreSQL sintaksē.

Taču ir jāievēro dažas prasības attiecībā uz datu bāzes struktūru:

- datu saņēmējam pamata datu tabulā jāuztur lauks DAGR ierakstu primārajām atslēgām. Šis lauks ir atzīmēts vispārējās konfigurācijas parametrā `export.root_dagr_key_field` un pēc noklusējuma tam nosaukums ir `dagr_key`;
- visām datu tabulām jāsaturs primārie identifikatori ar nosaukumu "id".

Papildus, uzmanība pievēršama tam, ka datu lauki ir reģistra jūtīgi (case sensitive).

Piemēram izmantojot iepriekšējā sadaļā minētajai *connector.yaml* struktūrai tiks izdrukātas šādas SQL komandas:

```

create table vzd_address(
  id bigint,
  adr_code text,
  address text,

```

```

code integer,
cieCode bigint,
cieName text,
dzvCode bigint,
dzvName text,
ielCode bigint,
ielName text,
lksCoordLong numeric,
lksCoordLat numeric,
nltCode bigint,
nltName text,
novCode bigint,
novName text,
pagCode bigint,
pagName text,
pilCode bigint,
pilName text,
typ bigint,
pilAtvk text,
novAtvk text,
pagAtvk text,
zipCode text,
irAdrese bool,
dagr_key text not null
);
comment on column vzd_address.adr_code is 'Adrešes kods (indekss)';
comment on column vzd_address.address is 'Adrese';
comment on column vzd_address.code is 'Adrešes kods';
comment on column vzd_address.cieCode is 'Ciema / mazciema kods';
comment on column vzd_address.cieName is 'Ciema / mazciema nosaukums';
comment on column vzd_address.dzvCode is 'Telpu grupas kods';
comment on column vzd_address.dzvName is 'Telpu grupa';
comment on column vzd_address.ielCode is 'Ielas kods';
comment on column vzd_address.ielName is 'Ielas nosaukums';
comment on column vzd_address.lksCoordLong is 'LKS ģeogrāfiskais garums';
comment on column vzd_address.lksCoordLat is 'LKS ģeogrāfiskais platums';
comment on column vzd_address.nltCode is 'Ēkas, apbūvei paredzēta zemes vienības (NLT) kods';
comment on column vzd_address.nltName is 'Ēka, apbūvei paredzēta zemes vienība (NLT)';
comment on column vzd_address.novCode is 'Novada kods';
comment on column vzd_address.novName is 'Novada nosaukums';
comment on column vzd_address.pagCode is 'Pagasta kods';
comment on column vzd_address.pagName is 'Pagasta nosaukums';
comment on column vzd_address.pilCode is 'Pilsētas kods';
comment on column vzd_address.pilName is 'Pilsētas nosaukums';
comment on column vzd_address.typ is 'Adresācijas objekta tips';
comment on column vzd_address.pilAtvk is 'Pilsētas ATVK kods';
comment on column vzd_address.novAtvk is 'Novada ATVK kods';
comment on column vzd_address.pagAtvk is 'Pagasta ATVK kods';
comment on column vzd_address.zipCode is 'Pasta indekss';
comment on column vzd_address.irAdrese is 'Ir adrese (citas adrese neatsaucas uz šo adrese
ierakstu)';
comment on column vzd_address.dagr_key is 'Ieraksta identifikācijas atslēga DAGR';

create table address_history(
    id bigint,
    vzd_address_id bigint,
    adrese text
);
comment on column address_history.adrese is 'Vēsturiskā adrese';

create table address_metadata(
    id bigint,
    vzd_address_id bigint,
    adrese_veids text,
    autors text,
    datums date,
    veids text,
    numurs text,
    nosaukums text
);
comment on column address_metadata.adrese_veids is 'Adrese veids';

```

```

comment on column address_metadata.actors is 'Autors';
comment on column address_metadata.datums is 'Datums';
comment on column address_metadata.veids is 'Veids';
comment on column address_metadata.numurs is 'Numurs';
comment on column address_metadata.nosaukums is 'Nosaukums';

alter table vzd_address add constraint pk_vzd_address primary key (id);
create index idx_vzd_address_dagr_key on vzd_address(dagr_key);

alter table address_history add constraint pk_address_history primary key (id);
create index idx_address_history_vzd_address_id on address_history(vzd_address_id);

alter table address_metadata add constraint pk_address_metadata primary key (id);
create index idx_address_metadata_vzd_address_id on address_metadata(vzd_address_id);

alter table address_history add constraint fk_address_history_vzd_address_id foreign key
(vzd_address_id) references vzd_address(id) on delete cascade;
alter table address_metadata add constraint fk_address_metadata_vzd_address_id foreign key
(vzd_address_id) references vzd_address(id) on delete cascade;

CREATE SEQUENCE seq;

```

4.4.4 Savienotāja palaišana un atklāšana

Palaižam datu devēja savienotāju, izpildot komandu:

```

java -Dconfig.file=./application.conf -
Dlogback.configurationFile=./logback.xml -cp ./postgresql-
42.7.4.jar;./dagr-connector.jar dagr.Connector

```

Komanda satur šādus argumentus:

- *-Dconfig.file=./application.conf* - norāda, ka aplikācija palaižama ar failā "application.conf" aprakstīto vispārējo konfigurāciju;
- *-Dlogback.configurationFile=./logback.xml* - norāda, ka aplikācijas logback komponente darbināma ar failā "logback.xml" norādīto konfigurāciju;

-cp ./postgresql-42.7.4.jar;./dagr-connector.jar dagr.Connector - norāda, ka javai ir jādarbina datu devēja savienotājs (dagr-connector.jar) kā main klasi izmantojot "dagr.Connector"

4.4.5 Savienotāja darbības rezultāti

Datu saņēmēja savienotāja darbības rezultātā datu saņēmēja datu bāzē konfigurācijā norādītajās tabulās:

- būs ierakstīti visi aktuālie DAGR datu kopas ieraksti:
 - pamata tabulas datu lauks ar nosaukumu dagr_key (vai citu alternatīvu nosaukumu, ja tas norādīts application.conf parametrā export.root_dagr_key_field), saturēs DAGR datu kopas ierakstu unikālās atslēgas;
 - pamata tabulas id datu lauks saturēs izmaiņu kārtas numuru (offset) pēdējai izmaiņai no [izmaiņu servisa](#), kuras ietvaros ieraksts ar noteiktu unikālo atslēgu aktualizēts. Pēc šī datu lauka maksimālās vērtības saņēmēja savienotājs nosaka cik tālu ir veikta datu sinhronizācija starp DAGR un datu saņēmēja datu bāzi.
- nebūs atrodami ieraksti, kas iepriekš bijuši datu kopā, bet pēc tam dzēsti. Respektīvi, datu saņēmēja savienotājs veic faktisku nevis loģisku sinhronizējamās datu kopas ierakstu dzēšanu.

5 Datu devēja integrācija

5.1 Pieteikšanās par datu devēju

Lai uzsāktu darbu ar DAGR, ir jāaizpilda DAGR veidlapa un jāiesniedz DAGR pārzinim, sūtot uz VRAA oficiālo elektronisko adresi caur Latvija.lv portālu vai pa pastu. Vairāk informācijas:

<https://www.vdaa.gov.lv/lv/datu-izplatisanas-un-parvaldibas-platforma>

Veidlapas apstrādes ietvaros DAGR pārzinis reģistrēs datu devēja pieteikumu DAGR sistēmā un veidlapā norādītajai personai piešķirs “pilnvarotā kontaktpersona” lomu. Tālāk šī persona varēs autentificēties DAGR pašapkalpošanās portālā un darboties datu devēja iestādes vārdā. DAGR pašapkalpošanās portāla adreses:

- Testa videi: <https://test.dagr.gov.lv/>
- Produkcijas videi: <https://dagr.gov.lv/>

Loma “pilnvarotā kontaktpersona” ļaus lietotājam iestādes pieteikumam pievienot un noņemt citus lietotājus tādējādi pārvaldot iestādes lietotāju sarakstu:

DAGR > Pieteikumu pārvaldība > Pieteikums

DAGR

Pieteikumu pārvaldība

Savienotāji

Iestāžu lietotāji

Zināšanu bāze

Monitorings

Audita notikumi

Izplatīšana

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Anonimizācijas uzdevumi

Sistēmas sertifikāti

Patērēšana

VIRSIIS atļaujas

Izmaiņu monitorings

Datu devējs

REĢISTRĀCIJAS NUMURS 90000024008

NOSAUKUMS Salaspils novada pašvaldība

STATUSS ✓ Apstiprināts

VĀRDS

KI

UZVĀRDS

Aļ

PERSONAS KODS

14

TĀLRUNIS

E-PASTA ADRESE

kli s.lv

☒ pilnvarotā kontaktpersona
☐ tehniskā kontaktpersona

Pārbaudīt un saglabāt

Dzēst

Pievienot jaunu kontaktpersonu

Bez lomas “pilnvarotā kontaktpersona” pastāv vēl arī loma “tehniskā kontaktpersona”, kas ļauj veikt visas pārējās darbības, bet neļauj pārvaldīt iestādes lietotājus.

5.2 Iestāžu lietotāju saraksts

DAGR pašapkalpošanās portālā ir izveidota sadaļa “Iestāžu lietotāji”, kurā var veikt vienotu lietotāju pārvaldību. Sadaļā ir pieejama tikai DAGR pašapkalpošanās portāla administratoriem un lietotājiem ar pilnvarotās personas tiesībām.

Par DAGR

DAGR
Pieteikumu pārvaldība
Savienotāji
Iestāžu lietotāji
Zināšanu bāze

Statistika

Administratora informācija

Datu devēji, pavisam 65
- ar vismaz vienu datu kopu 54
Datu saņēmēji, pavisam 50
- aktīvi pēdējā mēnesī 2
Reģistrēto datu kopu skaits 243

Izsaukumi
- pēdējās 24h 18
- pēdējā stundā 14
- pēdējā minūtē 2
- pēdējā sekundē 0

Rādīt tehniskos lauku nosaukumus

Sadaļā “Iestāžu lietotāji” var redzēt sarakstu ar iestāžu tehniskām un pilnvarotām kontaktpersonām. Par katru lietotāju ir šāda informācija:

- Iestādes veids (datu devējs vai datu saņēmējs).
- Vārds, Uzvārds un maskēts personas kods;
- Iestāde;
- Loma (pilnvarotā vai tehniskā).

Iestāžu lietotāji

+ Pievienot lietotāju

MEKLĒJAMĀIS TEKSTS

DATU SAŅĒMĒJS/DEVĒJS

KONTAKTPERSONAS VEIDS

ORGANIZĀCIJA

Datu devējs/saņēmējs	Vārds, uzvārds ↓	Iestāde	Kontaktpersona
saņēmējs	An *****	Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	tehniskā
saņēmējs	Jār *****	"Identity Solutions" SIA 40103958482	tehniskā
devējs	Jār *****	"Identity Solutions" SIA 40103958482	tehniskā
devējs	Jār *****	Iestāde "Elmex" 40900050831	pilnvarotā
saņēmējs	Jār *****	Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	pilnvarotā
saņēmējs	Jūl *****	Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	tehniskā

Iestāžu lietotāju sarakstā var redzēt kādas personas ir “neaktīvas”.

ORGANIZĀCIJA			
Sabiedrība ar ierobežotu atbildību "UNISO" X			
Datu devējs/saņēmējs	Vārds, uzvārds	Iestāde	Kontaktpersona
 saņēmējs	Ji **	Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	pilnvarotā
 devējs	Ic **	UNISO 40003562863	tehniskā
 saņēmējs (neaktīvs)	Ji **	Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	pilnvarotā

Iestāžu lietotāju sarakstu var filtrēt pēc šādiem parametriem:

- vārds un uzvārds,
- loma,
- iestāde,
- iestādes veids (datu devējs vai saņēmējs).

Iestāžu lietotāji

 Pievienot lietotāju

MEKLĒJAMĀIS TEKSTS

DATU SAŅĒMĒJS/DEVĒJS

saņēmējs ▼

KONTAKTPERSONAS VEIDS

tehniskā ▼

ORGANIZĀCIJA

Datu devējs/saņēmējs	Vārds, uzvārds	Iestāde	Kontaktpersona
 saņēmējs		Valsts policija 90000099040	tehniskā
 saņēmējs		Valsts sociālās apdrošināšanas aģentūra 90001669496	tehniskā

Lietotājs ar lomu “pilnvarotā kontaktpersona” var pievienot jaunus lietotājus, kā arī rediģēt un dzēst esošos iestādes lietotājus.

Pievienojot jaunu lietotāju, laukā “Datu saņēmējs/devējs” var izvēlēties iestādi tikai no tām iestādēm, kurās lietotājs ir pilnvarotā kontaktpersona.

Kontaktpersonas pievienošana

☒ Datu saņēmējs
☐ Datu devējs

DATU SAŅĒMĒJS

VĀRDS

"Identity Solutions" SIA 40103958482
Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863

Rediģēšanas ekrānformā var pievienot atzīmi, ka lietotājs ir "aktīvs", "neaktīvs", "aktīvs līdz".

DAGR > Iestāžu lietotāji > Jauns lietotājs

DAGR

Pieteikumu pārvaldība

Savienotāji

Iestāžu lietotāji

Zināšanu bāze

Monitorings

Audita notikumi

Izplatīšana

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Anonimizācijas uzdevumi

Sistēmas sertifikāti

Patērēšana

VIRSIS atļaujas

Izmaiņu monitorings

Kontaktpersonas pievienošana

DATU SAŅĒMĒJS Test Cepumi

VĀRDS

UZVĀRDS

PERSONAS KODS

TĀLRUNIS

E-PASTA ADRESE

☐ pilnvarotā kontaktpersona
☒ tehniskā kontaktpersona
☒ Aktīvs lietotāja konts
☐ Neaktīvs lietotāja konts

DATUMS, LĪDZ KURAM AKTĪVS

mm/dd/yyyy

☒ Pievienot kontaktpersonu [Atgriezties pie saraksta](#)

5.3 Izplatāmo datu kopu definēšana

Caur DAGR izplatāmie dati ir grupēti datu kopās. Pirms uzsākt izplatīt jauna veida datus ir jādefinē jaunu datu kopu. Datu kopas definēšana sastāv no diviem soļiem:

- Vispirms datu devējs reģistrē datu kopu priekšlikumu vidē (sandbox). Priekšlikumu vide ir izmantojama datu nodošanas testēšanai un atklādošanai. Priekšlikumu vidē reģistrētu datu kopu datiem var piekļūt pats datu devējs un DAGR administrators.
- Pēc tam DAGR administrators pārceļ datu kopu uz pamata vidi (dataset). Informācija par pamata vidē esošajām datu kopām tiek nodota uz VIRSIS un šajās kopās esošie dati pieejami datu ņēmējiem izmantojot VIRSIS piešķirtās datu piekļuves atļaujas.

5.3.1 Datu kopas pievienošana priekšlikumu videi (sandbox)

Caur DAGR izplatāmās datu kopas definīciju priekšlikumu vidē veido:

- **Nosaukums.** Unikāls datu kopas nosaukums. **Stingra rekomendācija** ir datu kopas nosaukumu rakstīt angļiski ar mazajiem burtiem, vārdus atdalīt ar apakš strīpu un prefiksā norādīt iestādes nosaukumu. Daži piemēri:
 - **ur_legal_entity** (Uzņēmumu reģistra tiesību subjekti);
 - **vzd_address** (Valsts zemes dienesta adreses);
 - **vbtai_orphan_certificate** (Valsts bērnu tiesību aizsardzības inspekcijas bāriņu apliecības).
- **Datu devējs.** Datu devēja reģistrācijas numurs un nosaukums;
- **Struktūra.** Datu struktūras definīcija YAML formātā. Datu struktūras definīcija apraksta datu kopas datu laukus, to formātus un to savstarpējo sasaisti. DAGR atbalsta daudzlīmeņu datu struktūras.
- **Komentārs.** Brīvā formā norādīts komentārs, kas apraksta datu kopu;

Caur DAGR dati tiek izplatīti tos formējot *key-value* paradīgmā, kur *key* ir ieraksta atslēga un *value* ir jebkuras sarežģītības datu struktūra. DAGR kodolā ieraksta *value* jeb vērtības daļa tiek uzglabāta JSON formātā.

Definēt jaunas datu kopas vai pieteikt izmaiņas eksistējošai datu kopai var DAGR pašapkalpošanās portāla sadaļā “Izplatīšana” -> “Priekšlikumu vide (sandbox)”.

DAGR > Priekšlikumu vide (sandbox) > teacher

Sandbox datu kopā teacher

[Atgriezties pie saraksta](#)

DATU KOPAS NOSAUKUMS: **teacher**

REĢISTRĀCIJAS NUMURS: 90000054163

SERVISI: <https://dagr.uniso.lv/api-docs/sandbox/90000054163>

LEJUPIELĀDE: [Lejupielādēt JSON](#) [Lejupielādēt CSV](#) [Pārlikt datus](#)

AUGŠUPLĀDE: [Augšuplādēt datus](#)

STRUKTŪRA

```

name: teacher
key: PersonCode, Id
data_description: Pedagogu aktuālās piederības (smilškastes dati)
fields:
  - Id: long: Piederības ID
  - PersonCode: long: Personas kods
  - PersonName: string: Personas vārds
  - InstitutionName: string: Iestādes nosaukums
  - IsGoodTeacher: boolean: Ir labs skolotājs
  - DateFrom: date: Piederības datums no
  - Citizenship: string: Pilsnība
  - comments: string: Pilsnības kods
  - CitizenshipCode: string: Pilsnības kods
  - Citizenship: string: Pilsnība
  - Positions: * :
    fields:
      - Id: long: Amata ID
      - ContractNumber: long: Līguma Nr
      - ContractDate: date: Līguma datums
  
```

Datu kopas definīcija priekšlikumu vidē (sandbox), detalizēti lasiet [wiki](#).

[Pārņem struktūru no pamata vides](#)

[Saglabāt](#) [Apstiprināt priekšlikumu](#) [Aplūkot pamatvidē](#) [Dzēst sandbox kopu](#)

Datu struktūras definīcijas veidojama atbilstoši šādai sintaksei:

```
name: {nosaukums}
```

DAGR datu kopas definīcijai jā sākas ar datu kopas nosaukumu. Nosaukumam datu struktūras definīcijā jā sakrīt ar datu kopas nosaukumu, kurš norādīts datu kopas ievades kartiņā.

```
key: {atslēgas lauks nr.1}, {atslēgas lauks nr.2}, {utt.}
```

Tālāk jā norāda datu lauki, kas veido ieraksta atslēgu. Atslēga var sastāvēt no viena vai vairākiem datu laukiem. Ņemot vērā, ka ieraksta atslēga pēc tam būs izmantojama ierakstu meklēšanai un filtrēšanai, par atslēgu jā izvēlas lauki ar biznesa nozīmi, piemēram, personas kods, reģistrācijas numurs, adreses kods, dokumenta numurs, u.c. Gadījumā, ja ieraksta unikalitātes nodrošināšanai nepieciešams izmantot kādu lauku, kuram nav biznesa nozīmes, tad tas liekams atslēgas beigās un iekļaujams iekavās. Iekavās iekļautie lauki meklēšanā nav obligāti. Atslēgā norādītajiem laukiem tālāk jā parādās datu lauku sarakstā.

```
data_description: {datu kopas apraksts}
```

Apraksta elementā norādāms datu kopas īss apraksts. Šim aprakstam ir informatīva nozīme.

```
search:
```

```
- name: {meklēšanas atslēgas nosaukums mazajiem burtiem}
  fields: {viens vai divi ar komatiem atdalīti pirmā līmeņa datu
lauku nosaukumi}
  description: {meklēšanas atslēgas apraksts}
- name: {meklēšanas atslēgas nosaukums mazajiem burtiem}
  fields: {viens vai divi ar komatiem atdalīti pirmā līmeņa datu
lauku nosaukumi}
  description: {nākamās meklēšanas atslēgas apraksts}
```

Datu kopai var būt viena vai vairākas papildus meklēšanas atslēgas. Meklēšanas atslēgu nosaukumi veidojami ar mazajiem burtiem un apakšsvītru, bet bez citiem speciālajiem simboliem un diakritiskajām zīmēm. Katra meklēšanas atslēga var izmantot vienu vai maksimums divus datu kopas pirmā hierarhijas līmeņa datu laukus. Šīs ir atslēgas, kuras papildus primārajai atslēgai, datu saņēmēji varēs izmantot, datu kopas ierakstu atļasei. Stingra rekomendācija ir izvēlēties tādas meklēšanas atslēgas, kur viena vērtība neatlasa ļoti daudz ierakstus. Iespēja izmantot definētās papildus atslēgas datu saņēmējiem tiek kontrolēta ar VIRSIS atļaujām.

```
fields:
```

```
- {lauka_nosaukums} {lauka_formāts} : {lauka apraksts}
- {salikta elementa nosaukums} * :
  comments : {salikta elementa komentārs}
  fields:
  | - {lauka_nosaukums} {lauka_formāts} : {lauka apraksts}
```

Datu kopas definīcijai jā noslēdzas ar hierarhisku datu lauku sarakstu. Šajā struktūrā var norādīt:

- datu laukus, kur katram laukam ir nosaukums, tips apraksts. Ja nenorāda datu tipu, tad tiek pieņemts, ka tas ir burtciparu kombinācija (string). Tiek atbalstīti šādi tipi:
 - **int** jeb **integer** veseliem skaitļiem;
 - **decimal** peldošā punkta decimālajiem daļskaitļiem - alternatīvi ar diviem skaitļiem decimal vietā var norādīt specifisku maksimālo simbolu skaitu pirms un pēc komanta (piemēram 8 2);
 - **string** burtciparu virknēm vispārīgā gadījumā, bet ar datumu un laiku saistītos gadījumos:
 - **date** burtciparu virknēm, kas satur datuma vērtību formētu atbilstoši RFC 3339 5.6 sadaļai;

- `dateTime` jeb burtciparu virknēm, kas satur datuma un laika vērtību formātu atbilstoši RFC 3339 5.6 sadaļai;
- `timestamp` līdzīgi kā `dateTime`, tikai bez laika zonas;
- `time`, burtciparu virknēm, kas satur laika vērtību.
- `boolean` patiesuma ("true" vai "false") vērtībām.
- saliktus elementus, kas sastāv no datu laukiem un citiem saliktiem elementiem. Ja ieraksti varēs saturēt vairāk kā vienu saliktā elementa vērtību (array), tad saliktā elementa definīcijā tas jāatzīmē ar reizinājuma zīmi `"**"`.

Praktiski datu kopu definīciju piemēri skatāmi pielikumā "8.1 Datu kopu definīcijas".

Pēc datu kopas definīcijas reģistrēšanas datu kopa ir pieejama priekšlikumu vidē smilškastēs režīmā. Respektīvi servisi datu nodošanai un apskatei pieejami DAGR REST API (<https://dagr.gov.lv/api>) sadaļā "sandbox". Smilškastēs režīms izmantojams datu nodošanas pārbaudei un atklūdošanai. Kad tas paveikts, tad jālūdz DAGR pārvaldniekam pārcelt datu kopa no smilškastēs uz pamata sadaļu.

Pēc datu kopas definīcijas reģistrēšanas priekšlikumu vidē datu devējam parādās pieeja šādām portāla darbībām:

- Augšupielādēt datu kopas datus izmantojot failu;
- Pārlūkot datu kopas datus;
- Lejupielādēt datu kopas datus JSON un CSV formātos.

5.3.1.1 Datu struktūras pielāgošana filtrēšanai

DAGR atbalsta ierakstu līmeņa filtrus VIRSIS datu piekļuves atļaujās pēc ieraksta atslēgā iekļautajiem datu laukiem. Apskatīsim piemēra datu struktūru:

```
name:                vdaa_pet_report
key:                 Code(, Status, Year, Submitted, Id)
data_description:    Atskaite par personas mājdzīvniekiem, labās
prakses piemērs
fields:
- Id                 : Pārskata
identifikators
- Code               : Personas kods
- Name               : Vārds
- Surname             : Uzvārds
- Year                int           : Gads
- Submitted           timestamp    : Atskaites
iesniegšanas datums un laiks
- Status (melnraksts, iesniegts, apstiprināts) : Pārskata statuss
- Pets * :
  comments           : Mājdzīvnieku
saraksts
  fields:
  - Species           : Suga
  - Breed             : Šķirne
  - Age                int       : Vecums
  - Name               : Vārds
```

Šī datu struktūra pieļauj VIRSIS sistēmā izsniegtajās atļaujās iekļaut filtrus ierakstiem pēc treknrakstā izceltajiem datu laukiem, respektīvi pēc personas koda, gada, statusa un iesniegšanas datuma. Gadījumā, ja kādā nofiltrēšanai pakļaujamajiem datu laukiem ir vēlme norādīt pieļaujamās vērtības (piemēram, ja tas ir klasifikatora kods), tad tās var uzskaitīt iekavās aiz datu lauka kā tas redzams piemērā datu laukam “Status”.

5.3.2 Datu kopas pārceļšana uz pamata sadaļu (dataset)

Lai pārceļtu datu kopu uz pamata sadaļu DAGR administrators DAGR portāla pārvaldības daļā atver sadaļu “Izplatīšana” -> “Priekšlikumu vide (sandbox)”. Šajā sadaļā ir saraksts ar visu datu devēju reģistrētajām datu kopām priekšlikumu vidē (sandbox).

DAGR > Priekšlikumu vide (sandbox)

DAGR ^

Pieteikumu pārvaldība

Savienotāji

Lietotāju pārvaldība

Portāla teksti

Iestāžu lietotāji

Zināšanu bāze

Monitorings ^

Audita notikumi

Nosūtītie paziņojumi

Izplatīšana ^

Datu kopu katalogs

Priekšlikumu vide (sandbox)

Datu kopas

Priekšlikumu vide (sandbox)

Saraksts ar datu kopu priekšlikumiem — izmēģinājuma versijām. Detalizētu aprakstu skatiet mūsu [wiki](#).

subjectOfLaw	90000151859 Veselības un darb spēju ekspertīzes ārstu valsts komisija
vdeavk_person_disability_test	90000151859 Veselības un darb spēju ekspertīzes ārstu valsts komisija
vdeavk_person_disability_test_keyless	90000151859 Veselības un darb spēju ekspertīzes ārstu valsts komisija
network_test	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
teacher_2	90000028508 Viedās administrācijas un reģionālās attīstības ministrija
pt_test_SAEIMA	90000028300 Latvijas Republikas Saeima
PT_UNISO_vraa_pep_test_v3	90001733697 Valsts digitālās attīstības aģentūra
bss_airship_register	90001733697 Valsts digitālās attīstības aģentūra
pt_akceptestšanas_uzdevums_parbaude_v2	90001733697 Valsts digitālās attīstības aģentūra
pt_bss_airship_register	90001733697 Valsts digitālās attīstības aģentūra
pt_test_anonimiz_VRAA	90001733697 Valsts digitālās attīstības aģentūra
pt_test_anonimiz_VRAA_v2	90001733697 Valsts digitālās attīstības aģentūra
pt_test_anonimiz_VRAA_v3	90001733697 Valsts digitālās attīstības aģentūra
pt_test_anonimiz_VRAA_v3_1111	90001733697 Valsts digitālās attīstības aģentūra
vdaa_pet_report	90001733697 Valsts digitālās attīstības aģentūra
vdaa_pt_testa_fails_gaisa	90001733697 Valsts digitālās attīstības aģentūra
vraa_PT_augusta_testam	90001733697 Valsts digitālās attīstības aģentūra
vraa_PT_augusta_testam_anon	90001733697 Valsts digitālās attīstības aģentūra
vraa_PT_augusta_testam_anonv2	90001733697 Valsts digitālās attīstības aģentūra

Jauna datu kopa

No šī saraksta ir jāizvēlas uz pamata sadaļu pārceļamā datu kopa, tā jāatver un jāspiež poga “Apstiprināt priekšlikumu” – tas pārceļ datu kopu no priekšlikumu vides (sandbox) uz pamata sadaļu (dataset).

DAGR > Priekšlikumu vide (sandbox) > teacher

Sandbox datu kopā teacher

[Atgriezties pie saraksta](#)

DATU KOPAS NOSAUKUMS: teacher

REĢISTRĀCIJAS NUMURS: 90000054163

SERVISI: <https://dagr.uniso.lv/api-docs/sandbox/90000054163>

LEJUPIELĀDE: [Lejupielādēt JSON](#) [Lejupielādēt CSV](#) [Pārlūkot datus](#)

AUGŠUPLĀDE: [Augšuplādēt datus](#)

STRUKTŪRA

```

name:      teacher
key:       PersonCode, Id
data_description: Pedagogu aktuālās piederības (smilšskates dati)
fields:
  - Id                long                : Piederības ID
  - PersonCode        : Personas kods
  - PersonName        : Personas vārds
  - InstitutionName    : Iestādes nosaukums
  - IsGoodTeacher      boolean           : Ir labs skolotājs
  - DateFrom          date              : Piederības datums no
  - Citizenship        :
  - Citizenship        : Pilsnība
  - comments           :
  - fields:
    - CitizenshipCode : Pilsnības kods
    - Citizenship     : Pilsnība
  - Positions * :
    - fields:
      - Id                long                : Amata ID
      - ContractNumber    : Līguma Nr
      - ContractDate      date              : Līguma datums

```

Datu kopas definīcija priekšlikumu vidē (sandbox), detalizēti lasiet [wiki](#).

[Pārņemt struktūru no pamata vides](#)

[Saglabāt](#) [Apstiprināt priekšlikumu](#) [Aplūkot pamatvidē](#) [Dzēst sandbox kopu](#)

Pārceltās datu kopas parādās sadaļā “Izplatīšana” -> “Datu kopas”. Šajā sadaļā var aplūkot, rediģēt un dzēst pamata daļas datu kopas.

DAGR > Pamata vide (dataset)	
DAGR ^ Pieteikumu pārvaldība Savienotāji Iestāžu lietotāji Zināšanu bāze	
Monitorings ^ Audita notikumi Paziņojumi lietotājiem Nosūtītie paziņojumi	
Izplatīšana ^ Datu kopu katalogs Priekšlikumu vide (sandbox) Pamata vide (dataset) Anonimizācijas uzdevumi Sistēmas sertifikāti	
Pamata vide (dataset) Saraksts ar datu kopu definīcijām, detalizētāk skatīt wiki.	
PTTEST_csv_fails	Valsts digitālās attīstības aģentūra Gaisa kuģu reģistrs, labās prakses piemērs
PTTEST_csv_fails2	Valsts digitālās attīstības aģentūra Gaisa kuģu reģistrs, labās prakses piemērs
PT_UNISO_vraa_pep_test_v2	Valsts reģionālā attīstības aģentūra Emergn PEP smilšskastes personas politiski nozīmīgas personas (angliski PEP) un tās pirmās pakāpes radnieku, kuri atbilst PEP statusam, dati
VRAA_pt_test_anonimiz_vraa_TEST1	Valsts reģionālās attīstības aģentūra Gada atskaite par uzņēmuma sniegtajiem pakalpojumiem
debeziem_example	
fpr_classifier	Pilsonības un migrācijas lietu pārvalde (PMLP) PMLP klasifikatori
fpr_person	Pilsonības un migrācijas lietu pārvalde (PMLP) PMLP dati par fizisko personu
pt_akcepttestesanas_uzdevums_pa...	
pt_test_anonimiz_VRAA_anon	VRAA TESTA vajadzībām Par testa vajadzībām datu kopa
pt_test_anonimiz_VRAA_v3	VRAA TESTA vajadzībām Par testa vajadzībām datu kopa
pt_test_tst_company_service_report	
tst_company_service_report	SIA UNISO Testa datu kopa - uzņēmuma atskaite par sniegtajiem pakalpojumiem
uniso_filter_test	
uniso_filter_test_v2	SIA UNISO datu kopu piemērs filtru vajadzībām
ur_annual_report	Uzņēmumu reģistrs (UR) UR dati - uzņēmumu gada pārskati
ur_commercial_pledge	Uzņēmumu reģistrs (UR) UR komercķīlu reģistra dati (commercialPledge)

Lai pabeigtu datu kopas pārcelšanu nepieciešams atvērt pārceļtās datu kopas definīciju un to papildināt norādot šādus parametrus:

- **data_owner_code** norāda datu devēja reģistrācijas numuru pēdīnās;
- **data_owner_name** norāda datu devēja nosaukumu;
- **data_namespace** norāda datu devēja saīsinājumu, kurš tiek izmantots servisu definīciju grupēšanai sadaļā <https://dagr.gov.lv/api>;
- **subject_type** norāda datu kopas ieraksta subjekta tipu jeb to uz ko norāda ieraksta primārā atslēga. Dokumenta rakstīšanas brīdī DAGR reģistrētajām datu kopām pastāv šādi subjekta tipi “persona”, “uzņēmums” un “adrese”. Jaunām datu kopām var būt nepieciešams reģistrēt jaunus subjekta tipus. Datu kopām kurām subjekta tips ir “persona” tiek piemērota paplašināta auditācija – papildus datu pieprasījuma metadatiem tiek saglabāts arī saraksts ar ierakstu identifikatoriem (personas kodiem), kuri bijuši pieprasījuma atbildē;
- **subject_id_fieldname** ir jānorāda datu lauka nosaukums no datu struktūras kurš satur subjekta identifikatoru;
- **monitor_missing_keys** var norādīt vai datu kopai var monitorēt ierakstus, kas nav datu kopā. Pēc noklusējuma šī vērtība ir *false*.
- **is_public_dataset** var norādīt vai datu kopa ir publiska un pieejama bez VIRSIS datu atļaujas. Pamatā paredzēts izmantot testa anonimizētajām datu kopām DAGR testa vidē.

DAGR > Pamata vide (dataset) > sif_honourable_family_card_anon

sif_honourable_family_card_anon [Atgriezties pie saraksta](#)

LEJUPIELĀDE [Lejupielādēt JSON](#) [Lejupielādēt CSV](#) [Pārlūkot](#)

AUGŠUPIELĀDE [Datu augšupielāde](#)

DATU KOPAS NOSAUKUMS sif_honourable_family_card_anon

Latvijas Goda ģimenes apliecību dati

🟢 Ši datu kopa ir publiski pieejama!

STRUKTŪRA

```

name: sif_honourable_family_card_anon
comments: "Latvijas Goda ģimenes apliecību dati"
data_owner_code: "90001237779"
data_owner_name: Sabiedrības integrācijas fonds
data_namespace: sif
data_description: Latvijas Goda ģimenes apliecību dati
subject_type: persona
subject_id_fieldname: personas_kods
is_public_dataset: true
key: personas_kods
fields:
  - numurs: Apliecības numurs
  - vārds: Apliecības turētāja vārds
  - uzvārds: Apliecības turētāja uzvārds
  - personas_kods: Apliecības turētāja personas kods
  - tips: Apliecības tips
  - izveidošanas_datums date: Apliecības izveidošanas datums
  - derīguma_termiņš date: Apliecības derīguma termiņš
  - ir_anulētaboolean: Pazīme, ka apliecība ir anulēta

```

Datu kopas definīcija pamata vidē (dataset), detalizētāk skatīt [wiki](#).

[Pāņemt struktūru no priekšlikumu vides](#)

✓ Saglabāt [Aplūkot priekšlikumu vidē](#) [Dzēst datu kopu](#)

Piebilde: Rediģējot datu struktūru datu kopai, kas ir aprakstīta VIRSIS sistēmā, drīkst pievienot jaunus datu laukus, bet nedrīkst mainīt eksistējošo datu lauku (sadaļa *fields*) nosaukumus vai to vietu hierarhijā. Nedrīkst arī dzēst eksistējošus datu laukus. Ja kāda no šīm darbībām tiks veikta, tad VIRSIS izsniegtās atļaujas vairs nedarbosies pilnā mērā (nevarēs izgūt mainītos vai dzēstos datu laukus). Ja tomēr nepieciešams dzēst vai mainīt nosaukumus datu laukiem, tad rekomendējams veidot jaunu datu kopu.

5.3.3 Datu kopas reģistrēšana Latvijas atvērto datu portālā

DAGR datu kopas var publicēt arī Latvijas atvērto datu portālā un izplatīt bez VIRSIS atļaujas. Pirmie soļi šajā procesā jāveic DAGR administratoram:

- Jāsasaista DAGR datu devējs ar LADP organizāciju. DAGR administrators to var paveikt DAGR portāla pieteikumu sadaļā norādot datu devējam LADP izmantoto organizācijas nosaukumu;
- Datu kopa jāatzīmē kā publiska pamata vides (dataset) sadaļā datu kopas struktūru papildinot ar pazīmi `is_public_dataset: true`.

Kad priekšnosacījumi paveikti, datu devējs DAGR portāla sadaļā "Pamata vides (dataset)" datu kopai var aizpildīt LADP metadatus:

DAGR > Pamata vide (dataset) > vraa_test_pet_report

DAGR

- Pieteikumu pārvaldība
- Savienojumi
- Iestāžu lietotāji
- Zināšanu bāze

Monitorings

- Audita notikumi
- Paziņojumi lietotājiem
- Nosūtītie paziņojumi

Izplatīšana

- Datu kopu katalogs
- Priekšlikumu vide (sandbox)
- Pamata vide (dataset)**
- Anonimizācijas uzdevumi

vraa_test_pet_report

Atgriezties pie saraksta

Testa datu kopa - atskaite par personas mājdzīvniekiem

Izpētīt kopu Lejupielādēt JSON Lejupielādēt CSV

Augšupielādēt kopas saturu LADP Metadati

Šī datu kopa ir publiski pieejama!

STRUKTŪRA

```
name: vraa_test_pet_report
data_owner_code: "40003562863"
data_owner_name: UNISO
data_namespace: uniso
is_public_dataset: true
key: Code(, Year)
data_description: Testa datu kopa - atskaite par personas mājdzīvniekiem
fields:
  - Code: Personas kods
  - Name: Vārds
```

Datu kopas definīcija pamata vidē (dataset), detalizētāk skatīt [wiki](#).

Pārņemt struktūru no priekšlikumu vides

Saglabāt struktūras izmaiņas Aplūkot priekšlikumu vidē Dzēst datu kopu

Metadatu norādīšanas ekrānformā var reģistrēt vai nu jaunu LADP datu kopu vai sasaistīt DAGR datu kopu ar eksistējošu LADP datu kopu:

- Lai reģistrētu jaunu LADP datu kopu datu laukā "LADP datu kopa" jāatstāj vērtība "Izveidot jaunu";
- Lai sasaistītu DAGR datu kopu ar eksistējošu LADP datu kopu datu laukā "LADP datu kopa" jāizvēlas kāda no LADP jau reģistrētajām datu kopām.

DAGR > Pamata vide (dataset) > vraa_test_pet_report > LADP Metadati

DAGR

Pieteikumu pārvaldība
Savienojumi
Iestāžu lietotāji
Zināšanu bāze

Monitorings

Audita notikumi
Paziņojumi lietotājiem
Nosūtītie paziņojumi

Izplatīšana

Datu kopu katalogs
Priekšlikumu vide (sandbox)
Pamata vide (dataset)
Anonimizācijas uzdevumi
Sistēmas sertifikāti

Patērēšana

VIRSIS atļaujas
Monitoringa kopas
Pielāgotie servisi

LADP Metadati:

LADP DATU KOPA
Izveidot jaunu

DATU KOPAS NOSAUKUMS

URL

APRAKSTS UN ATSLĒGVĀRDI
Normal
B
I
U

BIRKAS

ATJAUNUŠANAS BIEŽUMS

ATKALIZMANTOŠANAS NOSACĪJUMI

PUBLICĒTĀJS
Valsts digitālās attīstības aģentūra

REDZAMĪBA
Privāts

SAZIŅAS E-PASTS DATU JAUTĀJUMIEM

VALODA
Latviešu

DATU IZDOŠANAS DATUMS
DD.MM.YYYY

DATU PĒDĒJO IZMAIŅU DATUMS
DD.MM.YYYY

AUGSTVĒRTĪGAS DATU KOPAS KATEGORIJA

PIEMĒROJAMĀ LIKUMDOŠANA

Saglabāt
Notīrīt formu
Atgriezties pie datu kopas

Kad metadati pievienoti jānospiež poga "Saglabāt" un datu kopa būs pieejama Latvijas atvērto datu portālā. Tajā skaitā datu kopai Latvijas atvērto datu portālā būs reģistrēti resursi, kas vedīs uz DAGR datu apskates servisiem.

5.3.4 Specializētās datu kopas

5.3.4.1 Publiskie klasifikatori

Īpašs uz DAGR nododamās datu kopas paveids ir publiskie klasifikatori. Publisko klasifikatoru dati ir pieejami visiem datu ņēmējiem izmantojot standartizētu servisu bez VIRSIS atļaujas.

Publisko klasifikatoru datus var nodot uz DAGR kā atsevišķu datu kopu. Lai atzīmētu datus kā publisko klasifikatoru, datu struktūras definīcija jāpapildina ar šādu konstrukciju:

```

public_classifier:
{klasifikatora_koda_lauks}:          classifier_code
{klasifikatora_ieraksta_koda_lauks}: code
{klasifikatora_ieraksta_vecaka_koda_lauks}: parent_code
{vecaka_klasifikatora_koda_lauks}:   parent_classifier_code
{klasifikatora_ieraksta_nosaukums}:  name

```

{klasifikatora_ieraksta_apraksts}:	description
{klasifikators_no}:	valid_from
{klasifikators_līdz}:	valid_to

Šī konstrukcija kartē klasifikatora struktūru ar datu kopas datu laukiem.

Pilns datu kopas, kas ir publisks klasifikators, definīcijas piemērs:

```

name:      fpr_classifier
key:
- (ClassifierCode)
public_classifier:
  ClassifierCode: classifier_code
  Code:          code
  ParentCode:    parent_code
  ParentClassifierCode: parent_classifier_code
  Name:          name
  Description:    description
  ValidFrom:     valid_from
  ValidUntil:    valid_to
fields:
- Code                [!]          : Kods
- Name                [!]          : Nosaukums
- Id                  [!] long      : Identifikators
- Description          [!]          : Apraksts
- ParentId            [!] long      : Klasifikatora
virsvērtības identifikators
- ValidFrom           [!] dateTime : Spēkā no
- ValidUntil          [!] dateTime : Spēkā līdz
- LastChangeDateTime  [!] dateTime : Pēdējo izmaiņu
datumlaiks
- CustomJson          [!] :
  comments : Papildu atribūti (JSON formātā)
  table:
    fields:
      - IsEU           [!] boolean  : pazīme vai ir Eiropas
Savienības dalībvalsts
      - IsCitizenship  [!] boolean  : pazīme, vai ir
valstiskā piederība
      - CitizenshipName [!]          : Valstiskās piederības
nosaukums
      - OORG_FULL_TITLE [!]          : pilnais nosaukums
      - OORG_ATTRIBUTES [!] :
        comments: Saliktas struktūras OORG_ATTRIBUTES atribūts "Ir
neklasificēta iestāde" ("AttributeCode":"NI")
        table:
          fields:
            - Attributes * [!] :
              table:
                fields:
                  - AttributeCode [!]          : Atribūta kods
                  - Value          [!]          : Atribūta vērtība
- ClassifierCode          [!] (CCNT, CORD, CPAR, CPDK, CPET) :
Klasifikatora veida kods (CCLS)
- ParentCode              [!]          :
Klasifikatora virsvērtības kods
- ParentClassifierCode    [!] (CCNT, CORD, CPAR, CPDK, CPET) :
Klasifikatora virsvērtības klasifikatora veida kods

```

5.3.4.2 Biznesa notikumi

Vēl viens DAGR datu kopas paveids ir biznesa notikumi. Biznesa notikumi tiek izplatīti kā izmaiņu rindas, bet tiem nav iespējas atlasīt atsevišķus ierakstus pēc identifikatora. Tipiski biznesa notikumi ir saistīti ar kādu pamata datu kopu un signalizē noteikta veida izmaiņas. Piemēram, PMLP fizisko personu datu kopai ir gan kopējā izmaiņu rinda, gan vairāki saistītie biznesa notikumi – vārda maiņa, personas koda maiņa, personas apliecinājošo dokumentu maiņa, dzīvesvietas maiņa, utt.

Lai norādītu, ka datu kopa ir biznesa notikums tās definīcijai ir jāpievieno pazīme `events_only: true`. Piemērs:

```
name: test_custom_events
comments: Hello for events
key: code, year
events_only: true
fields:
- code
- year int
- info
```

Datu struktūras kartējums (5.6.2.3 "Datu struktūru kartējums") biznesa notikumiem realizējams tāpat kā citām datu kopām.

5.3.5 Datu pievienošana caur failu portālā

Atverot sadaļu **Priekšlikumu vide (sandbox)** vai **Pamata vide (dataset)**, datu devēja konaktpersonai ir pieejama darbība **Augšupielādēt datus**. Noklikšķinot uz tās, tiek atvērta augšupielādes forma.

Datu augšupielāde: uniso_filter_test

[Atgriezties](#)

FAILS

No file chosen

Fails ar ierakstiem CSV vai JSON formātā

IELĀDES REŽĪMS

☒ **Datu kopas ielāde**

Ātra testa ielāde

Dati netiek mainīti, esošie ieraksti netiek apskatīti.

Rūpīga testa ielāde

Dati netiek mainīti. Tiek analizēti esošie ieraksti.

Importa vēsture

Datums	Datu kopa	Statuss
24.09.2025 16:07	vraa_test_pet_report Unexpected content type: application/x-compressed. Expecting 'application/json' or 'text/csv'	klūda
24.09.2025 15:30	vraa_test_pet_report Sync done vraa_test_pet_report (from-dagr): 0 inserts, 1 update, 0 deletes, 9 skips (ver. 3.52710.0)	izpildīts
24.09.2025 15:28	vraa_test_pet_report Sync done vraa_test_pet_report (from-dagr): 0 inserts, 9 updates, 0 deletes, 0 skips (ver. 3.52710.0)	izpildīts

Šajā formā lietotājs var:

- izvēlēties failu,
- noteikt ielādes režīmu (pilna ielāde vai tikai ierakstu papildināšana/pārrakstīšana),
- uzklīkšķinot uz zobratīņa var norādīt papildu iestatījumus (piemēram, laika zonu datumiem, CSV atdalītāju, datu kvalitātes pārbaudes režīmu)

IELĀDES REŽĪMS

✓ **Datu kopas ielāde**

Ātra testa ielāde
 Dati netiek mainīti, esošie ieraksti netiek apskatīti.

Rūpīga testa ielāde
 Dati netiek mainīti. Tiek analizēti esošie ieraksti.

TEHNISKĀS DETALĀS

DATUMU LAIKAZONA

Europe/Riga

CSV LAUKU ATDALĪTĀJS

,

DATA_VALIDATION

off: validācija atslēgta

DATETIME_TRUNCATE

sekundes

☒ fail_on_missing_fields

☒ fail_on_unknown_fields

☒ dry_run

☒ dry_run_full

☒ report_duplicates

Kad iestatījumi ir sagatavoti, ir jānospiež poga:

- **Ielādēt pilnu datu kopu** - šajā gadījumā tiks aizstāti visi ieraksti ar failā eksistējošiem ierakstiem, vai
- **Ielādēt kā papildinājumu** - šajā gadījumā tiks pievienoti/pārrakstīti datu kopai failā esoši ieraksti.

Kad datu ielāde ir palaista, sistēma izveido uzdevumu, kuram var aplūkot izpildes statusu un rezultātu sadaļā **Importa vēsture**.

Importa vēsture



Datums	Datu kopa	Statuss
24.09.2025 18:48	uniso_filter_test	pievienots
24.09.2025 16:07	vraa_test_pet_report Unexpected content type: application/x-compressed. Expecting 'application/json' or 'text/csv'	kļūda
24.09.2025 15:30	vraa_test_pet_report Sync done vraa_test_pet_report (from-dagr): 0 inserts, 1 update, 0 deletes, 9 skips (ver. 3.52710.0)	izpildīts
24.09.2025 15:28	vraa_test_pet_report Sync done vraa_test_pet_report (from-dagr): 0 inserts, 9 updates, 0 deletes, 0 skips (ver. 3.52710.0)	izpildīts
19.09.2025 15:01	vraa_test_pet_report Sync done vraa_test_pet_report (from-dagr): 9 inserts, 1 update, 0 deletes, 0 skips (ver. 3.52516.0)	izpildīts

Datu importa notikumu vēsture

Izvēlieties importa mēģinājumu no saraksta.

Importa vēsturē lietotājs aplūko detalizētos augšupielādes rezultātus un importa statusu (pievienots, kļūda, izpildīts).

5.3.6 Datu lejupielāde caur REST API

Pieslēdzoties pašapkalpošanās portālam kā datu devēja kontaktpersona, lietotājs var atvērt **REST API** lapu (<https://dagr.gov.lv/api>) un izvēlēties sadaļu **Portal**:

← → ↺ 🌐 dev.dagr.gov.lv/api

DAGR IS

API dokumentācija

Speciālie servisi

[dagr](#)
[file](#)
[portal](#)
[sandbox](#)
[session](#)
[system](#)
[update](#)

Datu servisi

[data](#) 90001733697
[dataset](#) 90001733697 VRAA TESTA vajadzībām
[fpr](#) 90000054163 Pilsonības un migrācijas lietu pārvalde (PMLP)
[sopa](#)
[uniso](#) 40003562863 SIA UNISO
[ur](#) 90000270634 Uzņēmumu reģistrs (UR)
[vbtai](#) 90002056949 Valsts bērnu tiesību aizsardzības inspekcija (VBTAI)
[ydaa](#) 90001733697 Valsts digitālās attīstības aģentūra
[ydeavk](#) 90000151859 Veselības un darbaspēju ekspertīzes ārstu valsts komisija (VDEĀVK)
[vraa](#) 90001733697 VRAA TESTA vajadzībām
90001733697 Valsts reģionālā attīstības aģentūra
90001733697 Valsts reģionālās attīstības aģentūra

Lietotājam tiek piedāvāta iespēja lejupielādēt datu kopas, izmantojot servisu **GET /data_download/{dataset}**.

GET	/data_consumer_or_producer	Saraksts: Datu devēju un patērētāju	🔒
GET	/data_consumer_or_producer/{registration_number}/{consumer_or_producer}	Atgriezt pēc lauku 'registration_number' un 'consumer_or_producer' vērtībām	🔒
PUT	/data_consumer_or_producer/{registration_number}/{consumer_or_producer}	Pievienot/atjaunot 'data_consumer_or_producer' datus pēc lauku 'registration_number' un 'consumer_or_producer' vērtībām	🔒
DELETE	/data_consumer_or_producer/{registration_number}/{consumer_or_producer}	Dzēst 'data_consumer_or_producer' datus pēc lauku 'registration_number' un 'consumer_or_producer' vērtībām	🔒
GET	/data_download/{dataset}	Saraksts: Datu devēja datu nokraušanas kvalitātes kontrolei. Ierakstu secība nav definēta.	🔒
GET	/data_download/{dataset}/{municipality_code}	Saraksts: Datu devēja datu nokraušanas kvalitātes kontrolei. Ierakstu secība nav definēta.	🔒
POST	/data_import_task/{target_dataset}	Pievienot jaunu datu importa uzdevumu	🔒

Nospiežot pogu **Try it out**, tiek atvērta ievades forma, kur jānorāda datu kopas nosaukums un jāizvēlas formāts - **JSON** vai **CSV**.

GET /data_download/{dataset}
Saraksts: Datu devēja datu nokraušanas kvalitātes kontrolei. Ierakstu secība nav definēta.
Try it out

Parameters

Name	Description
dataset * required string (path)	dataset dataset

Responses

Code	Description	Links
200	Media type application/json Controls Accept header. Example Value Schema [{}]	No links

GET /data_download/{dataset}
Saraksts: Datu devēja datu nokraušanas kvalitātes kontrolei. Ierakstu secība nav definēta.
Cancel

Parameters

Name	Description
dataset * required string (path)	dataset Nosaukums

Servers

These operation-level options override the global server options.

https://dev.dagr.gov.lv

Execute

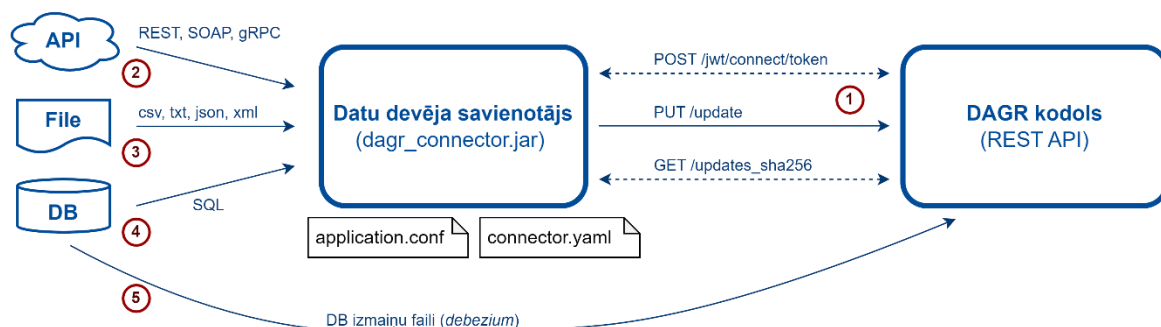
Responses

Code	Description	Links
200	Media type application/json application/json text/csv [{}]	No links
400	Bad request	No links
403	Nav aktuālas atbilstošas atļaujas	No links
500	Internal server error	No links
503	Service Unavailable	No links

Kad lietotājs apstiprina izvēli, sistēma sāk datu lejupielādi.

- Ja datu kopa ir hierarhiska, tā būs pieejama tikai JSON formātā.
- Ja datu kopa ir plakana, tā būs pieejama gan CSV, gan JSON formātā.

5.4 Datu nodošanas mehānisma izvēle



Ilustrācija 2 - datu nodošana uz DAGR

Pastāv vairāki varianti kā nodot datus no datu avota uz DAGR kodolu:

6. **Izmantojot DAGR REST API.** Datu devējs var organizēt datu nodošanu uz DAGR izmantojot DAGR REST API pieejamos datu nodošanas servissus. Serviss *POST /jwt/connect/token* izmantojams drošības talona pieprasīšanai. Tālāk serviss *PUT /update* izmantojams datu nodošanai. Papildus serviss *GET /updates_sha256*, ļauj izgūt sarakstu ar DAGR kodolā esošo ierakstu identifikatoriem un ierakstu vērtību kontrolsummām, kas var tikt izmantots nododamo izmaiņu atpazīšanai. Šis datu nodošanas variants ir salīdzinoši vienkāršs, bet prasa izmaiņas datu devēja informācijas sistēmā. Detalizētāks datu nodošanas mehānisma apraksts atrodams “5.6.1 DAGR REST API izmantošana”.
7. **Izmantojot datu devēja eksistējošos REST, SOAP vai gRPC datu izplatīšanas servissus.** Ja datu devējam ir eksistējoši datu izplatīšanas servisi, kas nodrošina iespēju atklāt datu kopas ierakstus un to izmaiņas, tad datu nodošanai uz DAGR var izmantot šos servissus. Šajā gadījumā datu nodošana uz DAGR tiks organizēta izmantojot datu devēja savienotāju, kurš nolasīs datus no eksistējošajiem datu devēja servisiem un nodos uz DAGR kodolu caur DAGR REST API. Šis variants neprasa izmaiņas datu devēja pusē, bet prasa pielāgotas izstrādes darbus DAGR projekta pusē, lai iestrādātu datu devēja eksistējošo servisu atbalstu datu devēja savienotāja komponentē.
8. **Nolasot datus no failiem.** Datus uz DAGR kodolu var nodot no csv, txt, json vai xml tipa failiem. Šajā gadījumā datu nodošana uz DAGR organizējama izmantojot datu devēja savienotāju, kurš nolasīs datus no failiem un nodos tos uz DAGR kodolu caur DAGR REST API. Pieņemot, ka datu avota sistēmai eksistē mehānisms kā eksportēt datus uz failiem, šis datu nodošanas mehānisms neprasa izstrādes pielāgojumus ne datu devēja ne DAGR pusē – viss kas nepieciešams ir uzstādīt un nokonfigurēt datu devēja savienotāju.
9. **Nolasot datus no DBVS ar SQL.** Datus uz DAGR kodolu var nodot izmantojot lasīšanas režīma SQL pieslēgumu avota datu bāzei. Šajā gadījumā datu nodošana uz DAGR organizējama izmantojot datu devēja savienotāju, kurš nolasīs datus no datu bāzes izmantojot JDBC pieslēgumu un nodos tos uz DAGR kodolu caur DAGR REST API. Lai nebūtu nepieciešams atvērt piekļuvi avota datu bāzei no ārējā tīkla, šajā gadījumā datu devēja savienotāju rekomendējams uzstādīt datu devēja infrastruktūrā. Šis datu nodošanas mehānisms neprasa izstrādes pielāgojumus ne datu devēja ne DAGR pusē - viss kas nepieciešams ir uzstādīt un nokonfigurēt datu devēja savienotāju.
10. **Nolasot datus no DBVS transakciju datnēm.** Datu nodošanu uz DAGR iespējams organizēt izmantojot *Debezium* (<https://debezium.io/>) tehnoloģiju, kas ļauj datus nolasīt no datu bāzu transakciju failiem. Šajā gadījumā uzstādāms *Debezium* pieslēgums datu avota datu bāzei, kurš datus nodos uz DAGR kodola Kafka rindu. Šis datu nodošanas mehānisms neprasa

izstrādes pielāgojumus ne datu devēja ne DAGR pusē, bet tas var prasīt būtiskas izmaiņas avota DBVS konfigurācijā.

Ja datu nodošana tiek organizēta izmantojot datu devēja savienotāju (otrais līdz ceturtais datu nodošanas mehānisms), tad pastāv vēl jautājums kur šo savienotāju izvietot. Pēc noklusējuma datu devēja savienotājs uzstādāms datu devēja infrastruktūrā. Taču vienojoties ar DAGR pārvaldnieku un atverot nepieciešamās piekļuves datu avotam, ir iespējama arī datu devēja savienotāja izvietošana DAGR centrālajā infrastruktūrā.

No esošās pieredzes sevi vislabāk ir pierādījusi datu izgūšana no DBVS ar SQL pieprasījumiem (ceturtais variants) vai failiem (trešais variants) izmantojot datu devēja savienotāju. Salīdzinoši vienkārša ir arī datu tieša ierakstīšana DAGR REST API (pirmais variants).

5.5 Datu devēja savienojuma reģistrēšana

Lai izplatītu datus ir jāreģistrē datu devēja savienojums. Ja tiek izplatīta viena datu kopa no viena avota, tad izmantojams viens savienojums. Bet ja datu devējs izplata vairākas datu kopas no vairākiem avotiem, tad katram datu avotam rekomendējams reģistrēt savu datu devēja savienojumu.

Datu devēja savienojumu iespējams reģistrēt pašapkalpošanās portālā sadaļas “DAGR” apakš sadaļā “Savienotāji”. Šajā sadaļā ir pieejams saraksts ar visiem savienojumiem:

DAGR > Savienotāji > Savienotāju pārskats

DAGR

Pieteikumu pārvaldība

Savienotāji

Iestāžu lietotāji

Zināšanu bāze

Monitorings

Audita notikumi

Izplatīšana

Datu kopu katalogs

Savienotāji

DATU DEVĒJA IZVĒLE

Autotransporta direkcija

Jauns Testa	Neaktīvs
Ptaugusta Ss	Neaktīvs
PT_testATD PtrudensSAV	Neaktīvs
pt_atd_test Sertifikāti: 1. 18.10.2024	5cc9621e964 Aktīvs
VDEĀK test vdeak_test_II Sertifikāti: 1. 2.10.2025.	8a0df3465d Aktīvs

Jauns savienotājs

Lai reģistrētu jaunu savienojumu jāaizpilda forma:

DAGR > Savienotāji > Jauns savienotājs

DAGR

Pieteikumu pārvaldība

Savienotāji

Zināšanu bāze

Monitorings

Audita notikumi

Izplatīšana

Datu kopas

Anonimizācija

Patērēšana

VIRŠIS atļaujas

Izmaiņu monitorings

Datu savienotāja reģistrācija

NOSAUKUMS: Savienojuma nosaukums pilniem vārdiem

IDENTIFIKATORS: VDAA_DEMO_SAV_1

DATU SAVIENOTĀJA TIPS: consumer

☒ Aktīvs?

TALONA DZĪVES LAIKS, S: 900

APRAKSTS: Savienojums, kas tiek reģistrēts DAGR zināšanu bāzes ekrānšāviņu vajadzībām

Saglabāt

Atgriezties pie saraksta

Šajā formā jānorāda portālā savienojuma nosaukums (informatīva nozīme), identifikators (tiks izmantots datu devēja savienotāja konfigurācijā), pazīme vai savienojums aktīvs, drošības talona dzīves cikla ilgums sekundēs un opcionāls apraksts.

Kad savienojums ir saglabāts, tam iespējams pievienot X.509 standarta drošības sertifikātu.

DAGR

Pieteikumu pārvaldība

Savienotāji

Zināšanu bāze

Monitorings

Audita notikumi

Datu savienotāja informācija

VispārīgiSertifikāti

Pievienot jaunu sertifikātu

1.	SHA256withRSA for CN=DAGR, OU=IT, O=UNISO, L=Rīga, ST=Rīga, C=LV 66d16e0a5c952e443b79b3d9ffa61018e3a9ec30	2.10.2024.	Dzēst
----	--	------------	-----------------------

Drošības sertifikāta pievienošana datu devēja savienotāja reģistrācijas scenārijā neatšķiras no datu ņēmēja savienotāja reģistrācijas scenārija – detalizētāku aprakstu par sertifikātiem skatīt sadaļā “4.2.1 Datu ņēmēja savienojuma reģistrēšana”.

5.6 Datu nodošana uz DAGR

Datu nodošanu uz DAGR konceptuāli iespējams organizēt vienā no trim variantiem:

- Izmantojot **DAGR REST API**. Šajā gadījumā datu nodošanu uz DAGR kontrolē datu avota sistēma;
- Izmantojot **datu devēja savienotāju**. Šajā gadījumā datus izgūšanu un nodošanu uz DAGR kontrolē datu devēja savienotājs. Datu devēja savienotājs datus var izgūt no SQL tipa datu bāzes, failiem (CSV, TXT, XML, JSON) vai eksistējošajiem datu devēja datu izplatīšanas servisiem;
- Izmantojot **Debezium** savienotāju. *Debezium* savienotājs ļauj izgūt datus no datu bāzu žurnālfailiem.

Tālākajās nodaļās aprakstīts katrs no variantiem.

5.6.1 DAGR REST API izmantošana

Datu nodošanu uz DAGR datu devējs var organizēt pats savā sistēmā izmantojot DAGR REST API.

Vispirms tiek pieprasīts drošības talons un tālāk ar to tiek darbināti šādi servisi:

- **PUT {dagr_adrese}/update/{service_name}**. Šis serviss nosūta strukturētus datus uz DAGR JSON formātā;
- **PUT {dagr_adrese}/file/{namespace}/{path_and_filename}**. Šis serviss uz DAGR nosūta datnes. Datu devējam pieejams arī DELETE serviss datnes dzēšanai un GET serviss datnes lejupielādei;
- **GET {dagr_adrese}/updates_sha256/{service_name}**. Šis serviss ļauj izgūt uz DAGR nosūtīto strukturēto datu ierakstu identifikatorus un kontrolsummas. Serviss izmantojams izmaiņu atpazīšanai kā arī atklūdošanas nolūkiem;
- **GET {dagr_adrese}/updates_last_timestamp/{service_name}**. Šis serviss ļauj noteikt pēdējo uz DAGR nosūtīto izmaiņu datumu un laiku. Servisu var izmantot izmaiņu atpazīšanai, ja datu avotā tiek uzturēti izmaiņu veikšanas datumi un laiki.

Minēto servisu apraksti OpenAPI standartā pieejami: <https://dagr.gov.lv/api-docs/update>

5.6.1.1 Drošības talona pieprasīšana

DAGR REST API autentifikācija ir organizēta saskaņā ar *OAuth 2.0 OpenId Connect* standarta *Client Credentials Flow* plūsmu.

Drošības talona pieprasīšanas adrese: <https://dagr.gov.lv/jwt/connect/token>

Drošības talona izsaukuma piemērs:

```
curl --location 'https://dagr.gov.lv/jwt/connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--header 'Cache-Control: no-cache' \
--data-urlencode 'client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer' \
--data-urlencode 'client_assertion=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJJEQUdSX0FLQ19URVNU'
```

```
XzIiLCJqdGkiOiJkOTQwZTliNC0yYjgwLTQ4NzUtYmEyMS1mYmVmYmVkYzUzZjIiLCJpc3MiOiJEQUdSX0F
LQ19URVNUXzIiLCJhdWQiOiJlcm46VEVTVClMTkI6REFHUi5TVFMvY29ubmVjdC90b2t1biIsImV4cCI6Ij
E3MDkyOTQ5NjQiLCJuYmYiOiIxNzA5MjkxMzY0In0.Xq3bZNI62hS9OFYx-
gNBEbh2SwQt72to_gL3Xwa9yh_uyUqd5__tDJuXkz-
F03Jn0CdcRKcdeitrKTtSCxIlP3WqQpbotsPHZvuF0ek26koyhtGhK2lw6RNF8mPkIktb4L6F0GuTjwc9Zm
Q1OlYo86u6_ZZWe2sS8aISJAUBmClW5cIt_qc3xY8no_UUJwD1yRGfxntGw3T_IyvIv0bPOLlOvyfmswpcI
7I5S2qR5EIJlKggM4SninbGfBlj_u7HGSziO14kIXmOikdhDviyP0je9kYBcwZgwYeja19dBxo1NkDd6A3B
G3jVPlvxJG2QQ1HdWNkHbXSonyJajDhGnA' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=DAGR_AKC_TEST_2' \
--data-urlencode 'scope=producer'
```

Šajā izsaukumā kā *client_id* ir jānorāda datu devēja savienojuma identifikators (skatīt sadaļu “0

Datu devēja savienojuma reģistrēšana”) un `client_assertion` ir jā sagatavo atbilstoši [RFC 7515 JSON Web Signature\(JWS\)](#) standartam.

Respektīvi, *client_assertion* sastāv no 'Header' un 'Payload', kas parakstīti ar datu ņēmēja savienotāja privāto atslēgu. 'Header' veidojams kā JSON objekts, kas satur paraksta parametrus, piemēram:

```
{
  "typ" : "JWT",
  "alg" : "RS256"
}
```

'Payload' arī veidojams kā JSON objekts un tajā ir iekļaujama šāda informācija:

```
{
  "sub": {datu devēja savienojuma id},
  "jti": {uuid},
  "iss": {datu devēja savienojuma id},
  "aud": urn:LNB:DAGR.STS/resources,
  "exp": {derīguma termiņš, max nbf + 1 stunda},
  "nbf": {parakstīšanas datums un laiks}
}
```

Drošības talona izsaukuma atbildes piemērs:

```
{
    "access_token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcml4TE5COKRBR1IuU1RTIiwic3ViIjoioTAWMDAwMTcwNzgiLCJhdWQlOiolsicHJvZHVjZXIIiLCJ1cm46TE5COKRBR1IuU1RTL3Jlc291cmNleCYjdCJLeHAiOjE3MDkyOTI1NjQsIm5iZiI6MTcwOTI1MTY2NCwiaWF0IjoxNzA5MjkxNjY0LCJsZWdhbGVudGl0eSI6IjkwMDAwMDE3MDC4IiwiaY2xpZw50X2lkIjoivlZEXOREU18xiwic2NvcGUiOlsicHJvZHVjZXIIiXX0.u832OMjhZZojv2wIpfie9ol9gkN0AVmi_YACeHxmYsk4JC-_XC3ad4tjp9mOjlfnGM4u8OXwX2WEhbc_i_3YtUMBUCsojdJ70u9-tuggggZivKmlXbWvld6Ui1FrKRPTdnhp5FCbtW6KgYTh_jqtqxqGiMT09ELPH0ts_-8DXlhAnnBfIbewznUiUuWeNWpsejd2zVD0hsancdPww7w9dzNjhvE6QlTrjn3Gk5Hn6b2Yj8yfvD3bNn6r7uOwo_o79mJo4LdWDNy3MX8he54OGd93EoMt0UoiW9FETdfitti3iioAFHo_sizkjHNGUY48o9xglV7-o41A3d0fyXtAWK9Q",
    "expires_in": 900,
    "token_type": "Bearer",
    "scope": "producer",
    "expires_epoch_utc": null
}
```

Izsniegtais drošības talons atbilst [RFC 7519](#) standartam un drošības talona satura pārlūkošanai ērti izmantot: <https://jwt.io/>

Postman piemērs, kas iekļauj arī drošības talona izsaukumu:

https://api.postman.com/collections/9631194-e7289644-5aa8-4639-8d4f-80c8591f2a49?access_key=PMAT-01GPE6A92NR3T5PNH5MTB9KMWD


```

zMtFWL0gBUIIT2rwIG7bcdZLkd8NqEsYtq2qOqkfnnccwm-iSn9uYV-1OM6q02NHcXVA68NU56PQkX_DvM-
Vl251GFtZFR8z9Tu64J1EaBWI2DR7x6BG7ULxO_bRzUaBSn6VrBkGxNe0YoerJ5uMJZex7q461ed5UxCk0Hl8-BE4eWSE-
OJCnaTAHcDnLaCEiIQgiwL9jbbq4hmWTe_O_0EO5zwRv0QiuieLhHQ7whBallI5fx0WSH0cXINT7GRwcqP5UuMbpQ_-
dwnBcEtcS-Iaj3Auz9ycrU3mIkldld4EnVs1OeaLmXakBk8AvR8mRHIq7A' \
--data '[
{
  "key": "40003562863/2023/5634634",
  "value": "{\n  \"Id\": 5634634,\n  \"CompanyName\": \"UNISO SIA\",\n
  \"RegistrationNumber\": \"40003562863\",\n  \"IsLicensed\": true,\n  \"Year\": 2023,\n
  \"Status\": \"apstiprināts\",\n  \"Submitted\": \"2024-03-06T09:22:11.593Z\",\n  \"Clients\":
  [\n    {\n      \"CompanyName\": \"VARAM\",\n      \"RegistrationNumber\": \"90000028508\",\n
      \"ClientSince\": \"2022-01-12\",\n      \"Services\": [\n        {\n          \"Name\":
          \"Programmatūras izstrāde\",\n          \"DateFrom\": \"2023-01-01\",\n          \"DateTo\":
          \"2023-12-31\"\n        },\n        {\n          \"Name\": \"Konsultācijas\",\n
          \"DateFrom\": \"2022-02-12\",\n          \"DateTo\": \"2022-03-12\"\n        }\n      ],\n
      \"VerifiedBy\": {\n        \"Name\": \"Jānis\",\n        \"Surname\": \"Sprukts\",\n
        \"Position\": \"Sistēmu arhitekts\"\n      }\n    }\n  ]\n}"
},
{
  "key": "40003562863/2022/5634234",
  "value": "{\n  \"Id\": 5634234,\n  \"CompanyName\": \"UNISO SIA\",\n
  \"RegistrationNumber\": \"40003562863\",\n  \"IsLicensed\": true,\n  \"Year\": 2022,\n
  \"Status\": \"apstiprināts\",\n  \"Submitted\": \"2023-03-06T09:22:11.593Z\",\n  \"Clients\":
  [\n    {\n      \"CompanyName\": \"VARAM\",\n      \"RegistrationNumber\": \"90000028508\",\n
      \"ClientSince\": \"2022-01-12\",\n      \"Services\": [\n        {\n          \"Name\":
          \"Programmatūras izstrāde\",\n          \"DateFrom\": \"2022-01-12\",\n          \"DateTo\":
          \"2022-12-31\"\n        },\n        {\n          \"Name\": \"Konsultācijas\",\n
          \"DateFrom\": \"2022-02-12\",\n          \"DateTo\": \"2022-03-12\"\n        }\n      ],\n
      \"VerifiedBy\": {\n        \"Name\": \"Jānis\",\n        \"Surname\": \"Sprukts\",\n
        \"Position\": \"Sistēmu arhitekts\"\n      }\n    }\n  ]\n}"
},
{
  "key": "40003562863/2022/56332",
  "value": "{ \"Id\": 56332, \"RegistrationNumber\": \"40003562863\", \"Year\": 2022 }"
}
]'

```

Piemērā redzams, ka galvenē (*header*) tiek padots drošības talons un tālāk saturā (*data*) tiek padotas divas ierakstu vērtības. Drošības talona pieprasīšana aprakstīta sadaļā “5.6.1.1 Drošības talona pieprasīšana”, savukārt viens no piemērā nododamajiem ierakstiem neeskiepotā formātā izskatās šādi:

```

{
  "Id": 5634634,
  "CompanyName": "UNISO SIA",
  "RegistrationNumber": "40003562863",
  "IsLicensed": true,
  "Year": 2023,
  "Status": "apstiprināts",
  "Submitted": "2024-03-06T09:22:11.593Z",
  "Clients": [
    {
      "CompanyName": "VARAM",
      "RegistrationNumber": "90000028508",
      "ClientSince": "2022-01-12",
      "Services": [
        {
          "Name": "Programmatūras izstrāde",
          "DateFrom": "2023-01-01",
          "DateTo": "2023-12-31"
        }
      ]
    }
  ],
  "VerifiedBy": {
    "Name": "Jānis",
    "Surname": "Sprukts",
    "Position": "Sistēmu arhitekts"
  }
}

```

1 }

Atbilstošā datu kopas definīcija (skatīt sadaļu “5.3 Izplatāmo datu kopu definēšana”), izskatās šādi:

```

name:      tst_company_service_report
key:       RegistrationNumber, Year(, Id)
data_description: Gada atskaite par uzņēmuma sniegtajiem pakalpojumiem
fields:
- Id                long                : Atskaites ID
- CompanyName       : Uzņēmuma nosaukums
- RegistrationNumber : Uzņēmuma reģistrācijas numurs
- IsLicensed        boolean            : Ir licenzēts pakalpojumu sniedzējs
- Year              int                 : Gads
- Status            : Atskaites statuss
- Submitted         timestamp          : Atskaites iesniegšanas datums un laiks
- Clients * :
  comments          : Uzņēmuma klienti
  fields:
  - CompanyName     : Uzņēmuma nosaukums
  - RegistrationNumber : Uzņēmuma reģistrācijas numurs
  - ClientSince      date              : Datums kopš kura ir klients
  - Services * :
    comments        : Sniegtie pakalpojumi
    fields:
    - Name          : Pakalpojuma nosaukums
    - DateFrom      date              : pakalpojuma sniegšanas sākuma datums
    - DateTo        date              : Pakalpojuma sniegšanas beigu datums
- VerifiedBy :
  comments    : Pārskata pārbaudītājs
  fields:
  - Name      : Vārds
  - Surname   : Uzvārds
  - Position  : Amats

```

Datu nodošanas servisa izsaukuma piemērs (ieraksta dzēšana):

```
curl --location --request PUT
'https://dagr.gov.lv/sandbox/40003562863/update/tst_company_service_report' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcm46VEVTVClMTkI6REFHUl5TVFMiLCJzdWIiOiI0MDAwMzU2Mjg2MyIsImF1ZCI6WjYwcm9kdWNlciIsInVyb2pURVNUU0xvOQpEQudSLLNUUy9yZXNvdXJjZXMiXSwiZXhwIjoxNzA5N2IxNjIyLCJmYm9yOiE3MDk3MjA3MjIsImhhdCI6MTcwOTcyMDcyMiwiibGVnYWxlb3RpdHkiOiI0MDAwMzU2Mjg2MyIsImNsaWVudF9pZCI6ImRlbnW9fZGRzX2NvbW51Y3Rvc18xIiwic2NvcGUlOlsicHJvZHVjZSIiXX0.Qm0lUSjblBRoX9Xkt8H
PeERjHf9U1_e6CYsZLsZYj7d5OHHLSOb8EH-
Sz8apIGbK8tQKbdtBKHKUKbgVMPiOsqnDVIguKNTq4drZn_I0gkhvucHIO4yHtYsNE240HlvIctfLzi2_7vCYB5r816XeS_
Y3PdS7CwQWlH5I_w407Uzp71FGS8hNuhBBBe7RcScL4TVZjMEuxVh67BeMNHsrNREumeuPoPHEGABrDTK2YEn_4biphbGW
5iYy5a4PnoKtOAN_pChs0Ug0Ojm18e8AK8fahDF7c4nNlmemNzyidOmwhOSCCIwJW_N82N1qX70PrFluszC7AYm0RLGk5s
Gmd2Q' \
--data '{
    {
        "key": "40003562863/2023/5634634",
        "value": null
    }
}'
```

5.6.1.3 Izmainu atpazīšana

Ja nepieciešams konstatēt kādi ieraksti ir nodoti uz DAGR, var izmantot servisu **GET**

{dagr_adrese}/updates_sha256/{service_name}. Šis serviss atgriež sarakstu ar ierakstu atslēgām un SHA256 ierakstu vērtību kontrolsummām uz DAGR nodotajiem datiem. Ilglaiīgi katram unikālajam ierakstam jeb ieraksta atslēgai DAGR uzglabā pēdējo nodoto vērtību un kontrolsummu, bet īslaicīgi un izmēģinājuma vidē (*sandbox*) tiek uzglabātas visas nodotās ierakstu vērtības un kontrolsummas.

updates_sha256 servisa atbildes piemērs:

```
[
  {
    "key": "40003562863/2023/5634634",
    "offset": 0,
    "sha256": "b4b5f8353f1d6bc7ab1db2637afa28cb1ce33fb67e4e0cef7f0d785884c8eac5"
  },
  {
    "key": "40003562863/2022/5634234",
    "offset": 1,
    "sha256": "a6d2b783425ca15c86fbb88c3a05fe643c2e78e7bb57bd581e8fedffc0a4bdcc"
  },
  {
    "key": "40003562863/2023/5634634",
    "offset": 2,
    "sha256": null
  },
  {
    "key": "40003562863/2022/56332",
    "offset": 3,
    "sha256": "42a70d82a114352e4af97610d8b4c756ff6586cc7557c5ac12af583dd18d5346"
  }
]
```

Piemērā redzam, ka ierakstam ar atslēgu 40003562863/2023/5634634 uz DAGR vispirms nodota viena vērtība, tad cita vērtība un visbeidzot ieraksts dzēsts. Pēc tam uz DAGR nodots ieraksts ar atslēgu 40003562863/2022/56332.

SHA256 kontrolsummu rēķināšanai DAGR izmanto OpenSSL bibliotēku. Online rīks, kas pārbaudīts, ka sakrīt ar DAGR aprēķinātajām SHA256 vērtībām: <https://tools.keycdn.com/sha256-online-generator>

Lai nodrošinātu, ka uz DAGR tiek nodotas tikai izmaiņas, izmantojot *updates_sha256* servisu, realizējams šāds mehānisms:

- Jāsaņem saraksts ar datu avotā esošo ierakstu atslēgām un aprēķinātajām ierakstu vērtībām sha256 kontrolsummām;
- Jāņem saraksts ar DAGR ierakstīto ierakstu atslēgām un sha256 kontrolsummām no DAGR izmantojot *updates_sha256* servisu;
- Jāsalīdzina saraksti un jāveic šādas darbības:
 - Visi ieraksti ar atslēgām, kuras ir avota sistēmā, bet nav DAGR ir jauni ieraksti. Tie ir nododami uz DAGR;
 - Visi ieraksti ar sakrītošām atslēgām un sakrītošām kontrolsummām ir ieraksti kuru vērtības nav mainījušās. Šos ierakstus uz DAGR nodot nav nepieciešams;
 - Visi ieraksti ar sakrītošām atslēgām un dažādām kontrolsummām ir modificēti ieraksti. Šo ierakstu jaunās vērtības ir nododamas uz DAGR;
 - Visi ieraksti ar atslēgām, kuras ir DAGR, bet nav avota sistēmā ir dzēsti ieraksti. Šo ierakstu dzēšanas fakts ir nododams uz DAGR (līdzīgi kā tiek nodots jauns vai modificēts ieraksts, tikai ieraksta vērtības laukā norādot *null*).

Piebilde: Ja DAGR atgriež vairākus ierakstus ar vienādām atslēgām, tad faktiskā pēdējā ierakstītā vērtība būs ar lielāko *offset* vērtību.

Augstāk minētais mehānisms ir tāds pats kā tiek izmantots datu devēja savienotājā.

5.6.1.4 Failu pievienošana

Failu nodošanai uz DAGR izmantojams serviss **PUT /file/{namespace}/{path_and_filename}**, kur {namespace} ir datu devēja datu kopām izmantotā vārdtelpa.

Datnes pievienošanas piemērs:

```
curl -X 'PUT' \
  'https://dagr.gov.lv/file/vsaa/test_file_name.txt' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcm46VEVTVClMTkI6REFHU55TVFMI6JzdWIiOiI5MDAwMTY2OTQ5NiIsImFlZCI6WyJwcm9kdWNlciIsInVyb2pURVNULUxOQjpeQUdSLlNUUy9yZXNvdXJjZXMiXSwiZXhwIjoxNzEwMzI4MDI4LCJmYyIjOiE3MTAzMjc5ImhhdCI6MTcxMDMyNzEyOCwibGVnYWxlb2RpdHkiOiI5MDAwMTY2OTQ5NiIsImNsaWVudF9pZCI6I1ZTUUFFVEVTVF9ERFNfMSIsInNjb3B1IjpbInByb2R1Y2VyI119.gZgNI6451nrLAMYp4UHRWLbTGIvPjIBxPpPlj2xiU5RyUf2lraD3CWdmyFyBirlhzSjmQ5ITvrtJrvlMm0ID4j3BPoDiUsrOtr-60FEbR3rwYLRuQyu_CiHDBLbyfyWr8KKSbyov834-37x1wrZjablC-t_tncs7Uyd9-pvQp_RjkVmEYgwfPLiq-edOyczQQ4WwcYwVoYKgH05yA5TX3LQ7n4oOGvJvMa9KoZenU2a49y4bLxwQGKtxV_kox75ABMXroaGT5111FpTRz1AnzCBh6n4D3s3D0novL2IppYgBxA5jhybkWJPdlynD_n4SVQDJN1Lp_UiRYq-gnbCGw' \
  -H 'Content-Type: application/octet-stream' \
  --data-binary '@test_file.txt'
```

Papildus PUT servisam, datu devējam pieejams arī **DELETE /file/{namespace}/{path_and_filename}** serviss, kas ļauj datni dzēst un **GET /file/{namespace}/{path_and_filename}**, kas ļauj datni lejupielādēt.

5.6.2 Datu devēja savienotāja uzstādīšana

Datu devēja savienotājs ir JAVA aplikācija, kas izgūst datus no datu avota, tos transformē un nodod uz DAGR kodolu izmantojot iepriekšējā nodaļā aprakstīto DAGR REST API. Datu devēja savienotāju ir rekomendējams izvietot datu devēja infrastruktūrā, lai nebūtu jāatver piekļuve datu avotam ārpus datu devēja tīkla infrastruktūras un datu devējam būtu pilna kontrole pār datu devēja savienotāju un tā konfigurāciju.

Datu devēja savienotāja sekmīgai darbināšanai ir nepieciešamas šādas datnes:

- **Datu devēja savienotāja aplikācija (dagr-connector.jar).** Datu devēja savienotāja izpildkods, kuru iespējams lejupielādēt DAGR pašapkalpošanās portālā (<https://dagr.gov.lv/download/dagr-connector>)
- **Datu devēja savienotāja konfigurācija (application.conf).** Datu devēja savienotāja konfigurācijas fails, kurš satur informāciju par pieslēgumu DAGR kodolam (datu devēja savienotāja identifikators, privātā atslēga, DAGR kodola adreses), par datu avotu (avota veids, avota pieslēguma informācija), ceļu uz datu avota definīciju, savienotāja darbināšanas biežumu.
- **Datu kartējums (connector.yaml).** Datu avota struktūras kartējums pret mērķa struktūru.
- **Žurnālēšanas uzstādījumi (logback.xml).** Definē datu devēja savienotāja žurnāla pierakstu veidošanas apjomu. Atklādošanas nolūkiem uzstādāms lielāks apjoms, produkcijas darbināšanas režīmam mazāks.

5.6.2.1 Datu devēja savienotāja uzstādīšanas instrukcija

Datu devēja savienotāja tehniskās uzstādīšanas instrukcijas (piemērs:

1. Instalēt java 17
2. Lejupielādēt dagr-connector.jar

3. Lejupielādēt postgres JDBC draiveri

<https://jdbc.postgresql.org/download/>

4. Izveido mapi, novieto dagr-connector.jar un postgres JDBC draiveri

5. Turpat novieto application.conf un connector.yaml

6. Koriģē application.conf - JDBC pielēguma parametri, ceļš uz metadata_file (connector.yaml)

7. Koriģē connector.yaml - db struktūra

***Papildus konfigurācijas maiņa LNB videi

target_server_uri = "https://dagr.gov.lv"

jwt.service.uri = "https://dagr.gov.lv/jwt/connect/token"

jwt.payload-audience = "urn:LNB:DAGR.STS/resources"

jwt.scope = "producer"

ģenerētam RSA atslēgu pārim (private key/pub cert)

piem., `openssl req -newkey rsa:2048 -nodes -keyout test.key.pem -x509 -days 365 -out utest.certificate.pem`

provider.key = privātā atslēga (single line) - saturs starp -----BEGIN PRIVATE KEY----- un -----END PRIVATE KEY-----

provider.client_id = "{DAGR portālā reģistrēta savienotāja identifikators}"

kamēr pub cert jābūt reģistrētam DAGR portālā pie atbilstošā savienotāja (ar atbilstošo client_id)

8. Palaiž datu devēja savienotāju ar komandu:

java -Dconfig.file=./application.conf -Dlogback.configurationFile=./logback.xml -cp ./dagr-connector.jar dagr.Connector

5.6.2.2 *Datu devēja savienotāja vispārīgā konfigurācija*

Datu devēja savienotāja vispārējā konfigurācija norādāma failā application.conf. Zemāk references konfigurācija, kurā uzskaitīti visi parametri un aprakstīts to pielietojums.

application.conf references konfigurācija:

DAGR CONNECTOR CONFIG

data.datetime.truncate_to = seconds

Datuma laika saīsināšana, ja te nav null. Iespējamās vērtības: seconds, millis, micros, nanos

data.timezone = "Europe/Riga"

tiek izmantots, lai formatētu dateTime datus

```

data.validation = enabled
# Datu kvalitātes pārbaudes iestatījumi. Vērtības: enforced, enabled, off. Ja enforced vai enabled, pārbaudīs
# atbilstību metadatu failā norādītajam obligājumam un šablonam. Enabled režīmā neatbilstības neapturēs
# ielādes procesu, enforced režīmā apturēs.

dataset_names      = my_service
# datu kopu (servisu) nosaukumi DAGR sistēmā

akka.quartz.schedules.connector.expression = "*/10 * * ? * *"
# Datu aktualizēšanas biežums. Atkarīgs no viena cikla izpildes ātruma. Piemērā, izpilde notiek katrā 10. sekundē.

datasource_type    = csv
# csv, jdbc, u.c.

do_connection_test  = false
# Konektivitātes atklādošanai - vai pirms datu sūtīšanas vispirms sūtīt
# tukšu datu kopu, lai pārlicinātos par konekcijas darīgumu

connection_test_interval= 55 s
# Konektivitātes atklādošanai - laiks, kas pagājis kopš iepriekšējāš datu
# sūtīšanas, kad jāizpilda konekcijas tests, (sk. augstāk)

dry_run            = false
# Testēšanas nolūkiem. Ja true - neveiks izmaiņas uz servera

dry_run_full       = false
# Testēšanas nolūkiem. Ja true – neprasis statusu un neveiks izmaiņas uz servera

report_duplicates   = true
# Ienākošo datu kvalitātes pārbaude

delete_unpaired_records = true
# Vai dzēst ierakstus, kuri neatradās ienākošajā ("avota") datu kopā

target_server_uri   = null
# DAGR serveris, kas saņems datus, piemēram "https://dagr.gov.lv/". Gadījumā, ja dati tiek nodoti uz
# izmēģinājuma vidi (sandbox), tad adrese jānorāda formā
# "https://dagr.gov.lv/sandbox/{datu_devēja_reģistrācijas_numurs}".

provider.key        = null
# Privātā atslēga Base64 kodējumā bez -----BEGIN PRIVATE KEY----- un -----END PRIVATE KEY----- elementiem
# Piemērs,
# "MIIEvQIBADANBgkqhkiw0BAQEFAASCBKcwggSjAgEAAoIBAQDfRfIz6URnDrmITZAestc7qUk37y3wp2KJhUJG1+4
# mFt5fQTkYm/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYD
# MaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgV
# cNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJp
# okwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkH
# awmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo
# 90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIX
# zHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj
# 7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjtQ/PyWVRtkO
# xDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCSQKBgA1bWjt
# Q/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDmaPGnCS
# QKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEgIAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJ
# YDmaPGnCSQKBgA1bWjtQ/EPaFqC7o8jNR+"

provider.client_id  = null

```



```

# Datu devēja savienotāja identifikators, piemēram, "MY_ORG_connector_1"

jwt.service.uri      = null
# Drošības talona pieprasījuma adrese, piemēram "https://dagr.gov.lv/jwt/connect/token "

#jwt.service.with-host-header = true

jwt.payload-audience = null
# Drošības talona audience, piemēram " urn:LNB:DAGR.STS/resources"

jwt.scope = "producer"
# Datu devēja savienotājam "producer", datu ņēmējam "consumer"

metadata_file      = "./connector.yaml"
# Ceļš uz datu struktūras kartējumi

progress_report_interval= 10 s
# Cik bieži atskaitīties par tekošo statusu

# Automātiskie paziņojumi tehniskajām kontaktpersonām
report_problems.by_email = false
report_problems.interval = 30 m
report_problems.timezone = "Europe/Riga"
# tiek izmantots datuma laika formatēšanai automātiskajos paziņojumos

# Šī sadaļa attiecas tikai uz csv konektoru =====
# codifs.json - outer object key is codif name, inner object represents codif keys and values
csv.codifs_file      = null    # ${csv.files.directory}/codifs.json
csv.files {
  directory          = "."
  filename_pattern   = ${dataset_names}.csv
  fields_delimiter    = ","
  encoding            = "UTF-8" # https://docs.oracle.com/javase/8/docs/technotes/guides/intl/encoding.doc.html
  date_format        = "[d.M.u.][M/d/u][u-M-d]"
  datetime_format     = "[[u-M-d][u.M.d][' ']['T']H:m[:s][.SSS][.SS][.S]['X']['Z']]"
}

# Šī sadaļa attiecas tikai uz jdbc konektoru =====
jdbc {
  use_connection_pool = true
  data {
    jdbcUrl = "jdbc:hsqldb:mem:main"
    username = "SA"
    password = ""
    # Ja use_connection_pool ir true, tad tiks ņemti vērā papildus iestatījumi:
    # https://github.com/brettwooldridge/HikariCP#gear-configuration-knobs-baby
    maximumPoolSize = 1
    readOnly = true
  }
}

```

Ja konfigurācija, kas attiecas uz pieslēgumu DAGR kodolam ir korekta, tad aktuālo konfigurācijas informāciju, atskaitot paroles un atslēgas, var apskatīties arī DAGR pašapkalpošanās portāla sadaļā "Savienotāji".

5.6.2.3 Datu struktūru kartējums

Datu devēja savienotāja konfigurācijā ir jāiekļauj uz DAGR kodolu nododamo datu kopu vai biznesa notikumu mērķa datu struktūras kartējums pret avota datu struktūrām. Šī kartējuma rezultātam ir jāatbilst DAGR pašapkalpošanās portālā definētajai mērķa datu struktūrai.

Mērķa datu struktūras kartējums pret datu avotu norādāms datu devēja savienotāja konfigurācijas failā **connector.yaml**, YAML formātā.

Datu struktūras kartējuma definīcija vispārīgā formā ir šāda:

```
name:          {DAGR datu kopas nosaukums}
table:         {SQL avota pamata tabulas nosaukums} {tabulas saīsinājums}
key:           {saraksts ar datu laukiem, kas veido ieraksta atslēgu}
joins:         {saišu saraksts, ja datu avots sastāv no vairākām tabulām}
fields:        {datu lauku saraksts / struktūra}
filter:        {filtri}
```

Datu kopas nosaukums norāda kurā datu kopā dati ielādējami. Tam ir jāsakrīt ar DAGR pašapkalpošanās portālā reģistrētas datu kopas nosaukumu (skatīt “5.3 Izplatāmo datu kopu definēšana”). Piemēram:

```
name: testa_kopa
table: testa_tabula
key: kods
...
```

SQL avota pamata tabulas nosaukums norādāms gadījumos, ja dati tiek ielādēti no SQL datu bāzes. Dati var tikt ielādēti arī no vairākām tabulām reizē, kā tas būs parādīts pie elementa “joins:” apraksta, bet šeit jānorāda pamata tabula. Ja dati tiek ielādēti no faila vai kādā citā veidā, tad šī vērtība jāatstāj tukša. Piemērs:

```
name: testa_kopa
table: testa_tabula
key: kods
...
```

Atslēgas sadaļā ir jānorāda datu lauki no elementa “fields:”, kas veidos ierakstu primāro atslēgu. Ierakstu primārajai atslēgai ir jābūt unikālai. Piemēram:

```
name: person_pension
table:
key: persk, sak, pensveids
fields:
- persk          11
- uzvards        34
- vards          34
- sak            date
- beig           date
- pensveids      2
```

Saišu (joins:) sadaļa ir paredzēta lai apvienotu datus no vairākām SQL tabulām. Šajā sadaļā norāda saites formātā: {pamata tabula} [{sasaistes vienādība}] {piesaistāmā tabula} {saīsinājums}. Piemērs:

```

name:          vbtai_orphan_certificate
table:         t_Cards c
key:           PersonId, DocNr
comments:      Bāreņu apliecības
joins:
- c [c.Id = cc.Card_id ] t_CanceledCards cc?
- c [c.fk_Person = p.Id] t_Persons p
fields:
- PersonId          = p.PK                : Personas kods
- DocNr             int = c.Number          : ASGN numurs - apliecības numurs
- Name              = p.Name              : Personas vārds un otrs vārds kopā
- Surname           = p.Surname           : Personas uzvārds
- CreateDate        date = p.Decree_date    : ASGN izdošanas datums
- EndDate           date = c.Valid_till     : ASGN termiņš
- StopDate          date = cc.Cancel_date   : ASGN anulēšanas datums
- ASGN_TYPE         int = c.Status          : Braukšanas maksas

```

Datu lauku sarakstā norāda datu kopā iekļaujamos datu laukus. Ja datu avots ir fails, tad datu lauku nosaukumi ir patvaļīgi un norādāmi secībā, kas datu laukiem ir failā. Ja datu avots ir SQL tabulas, tad datu lauku nosaukumiem jāsakrīt ar tabulu datu lauku nosaukumiem vai jābūt ar patvaļīgu nosaukumu, kurš kartēts pret tabulas datu lauku nosaukumu. Datu laukiem opcionāli var norādīt datu formātu kuram tiem jāatbilst un komentāru.

Piemērs, kur dati tiek ielādēti no faila:

```

name:  vzd_address
table:
key:
- adr_code
fields:
- adr_code          : Adreses kods (indekss)
- address           : Adrese
- code              int   : Adreses kods
- cieCode           long  : Ciema / mazciema kods
- cieName           : Ciema / mazciema nosaukums
- dzvCode           long  : Telpu grupas kods
- dzvName           : Telpu grupa
- ielCode           long  : Ielas kods

```

Piemērs, kur dati tiek ielādēti no vienas tabulas saglabājot lauku nosaukumus:

```

name:  sopa_needy
table:  needy
comments: Trūcīgie
key:    person_code, district_atvk, status_from
fields:
- id
- person_code
- district_atvk
- submit_date
- status_from
- status_till
- decision_date
- decision_nr
- household_id
- address_ar_code
- address_text
- last_date

```

Piemērs, kur dati tiek ielādēti no vairākām tabulām ar jauniem datu lauku nosaukumiem:

```

name:      vbtai_orphan_certificate
table:     t_Cards c
comments:  Bāreņu apliecības
joins:
- c [c.Id = cc.Card_id ] t_CanceledCards cc?
- c [c.fk_Person = p.Id] t_Persons p
key:      PersonId, DocNr
fields:
- PersonId          = p.PK                : Personas kods
- DocNr             int = c.Number          : ASGN numurs - apliecības numurs
- Name              = p.Name                : Personas vārds un otrs vārds kopā
- Surname           = p.Surname            : Personas uzvārds
- CreateDate        date = p.Decree_date    : ASGN izdošanas datums
- EndDate           date = c.Valid_till     : ASGN termiņš
- StopDate          date = cc.Cancel_date   : ASGN anulēšanas datums
- ASGN_TYPE         int = c.Status          : Atvieglojumu statuss

```

Filtru sadaļā var norādīt nosacījumus kuriem ierakstiem ir jāatbilst, lai tie tiktu atlasīti un iekļauti uz DAGR nododamo ierakstu izlasē. Piemērs:

```

name:      sopa_needy
table:     needy
comments:  Trūcīgie
key:      person_code, district_atvk, status_from
fields:
- id
- person_code
- district_atvk
- submit_date
- status_from
- status_till
- decision_date
- decision_nr
- household_id
- address_ar_code
- address_text
- last_date
filter:
- active = -1
- person_code != null
- status_from != null
- district_atvk = substr(:dataset_name, instr(:dataset_name, '/') + 1)

```

Pilns **connector.yaml** satura piemērs (JDBC gadījumā vispirms norādāmas datu avota tabulas, pēc tam DAGR mērķa datu struktūra pēc nosaukuma):

```

connector.yaml

    table: tst_tabula

    columns:

    - code          50

    - name          250

```

```
- address_code
- issued      date
- application_type
- application_num
- valid_from   dateTime
- valid_to     dateTime
```

```
name: tst_serviss
```

```
table: tst_tabula
```

```
key: code
```

```
fields:
```

```
- code
- name
- address_code
- issued
- application_type
- application_num
- valid_from
- valid_to
```

Ja konfigurācija, kas attiecas uz pieslēgumu DAGR kodolam ir korekta, tad aktuālo datu kartējumu, var apskatīties arī DAGR pašapkalpošanās portāla sadaļā “Savienotāji”.

5.6.2.4 *Datu kvalitātes kritēriji*

Datu kvalitātes kritēriju pārbaudi nodrošina datu devēja savienotājs.

Obligātie datu kvalitātes kritēriji ir tie datu kvalitātes kritēriji kuriem ir jāizpildās, lai DAGR tehniski spētu izgūt datus no datu avota un ielādēt datus DAGR kodolā. Obligātie datu kvalitātes kritēriji ir:

- **Datu struktūras kontrole.** Uzsākot datu nodošanu datu devēja savienotājs pārlicinās, ka datu devēja konfigurācijas failā **connector.yaml** norādītais datu avota struktūras kartējums pret mērķa datu struktūru ir korekts. Ja kartējums ir korekts, tālāk datu devēja savienotājs pārlicinās, ka dati avotā ir pieejami konfigurācijas failā **connector.yaml** norādītajā struktūrā. Visbeidzot datu devēja savienotājs izgūst datus no avota un nodod tos DAGR kodolam, kurš savukārt pārbauda saņemto datu atbilstību DAGR pašapkalpošanās portālā

definētajai mērķa datu struktūrai. Ja kāda no pārbaudēm ir neveiksmīga, datu devēja savienotājs izvada atbilstošu kļūdas paziņojumu un tālāk darbu neturpina.

- **Obligātuma kontrole.** Izgūstot datus no datu avota, datu devēja savienotājs pārliedz, ka ierakstiem ir aizpildīti visi obligātie datu lauki. Priekš DAGR darbības par obligātiem uzskatāmi tie datu lauki, kas ir iekļauti ieraksta primārās atslēgas meklēšanas daļā. Ja kādam ierakstam šie datu lauki ir tukši, tiek izvadīts atbilstošs kļūdas paziņojums un konkrētais ieraksts uz DAGR nodots netiek.
- **Datu formātu kontrole.** Nododot datus, datu devēja savienotājs katram ierakstam pārliedz, ka ieraksta parametri atbilst mērķa datu struktūrā definētajiem datu formātiem. Ja kādam ierakstam dati neatbilst noteiktajam formātam, tad tiek izvadīts atbilstošs kļūdas paziņojums un konkrētais ieraksts uz DAGR nodots netiek.

Papildus datu kvalitātes kritēriji. Šie ir datu kvalitātes kritēriji, kurus iespējams norādīt datu devēja savienotāja konfigurācijā **connector.yaml**. Ja kāds no papildus datu kvalitātes kritērijiem netiek izpildīts, tad datu devēja savienotājs izvada atbilstošu kļūdas paziņojumu, bet kļūdainie ieraksti joprojām uz DAGR tiek nodoti. Ir iespējams norādīt šādus papildus datu kvalitātes kritērijus:

- **Vērtību saraksts.** Datu lauka vērtībai ir jāsakrīt ar kādu no konfigurācijā definētajām vērtībām. Piemērs:

```
name: validate_enum
table:
fields:
- name (Jānis, Pēteris)
```

- **RegEx izteiksme.** Datu lauka vērtība atbilst konfigurācijā norādītajai [RegEx](#) izteiksmei. Šādā veidā var norādīt visdažādākā veida datu kvalitātes pārbaudes. Piemērs:

```
name: validate_pattern
table:
fields:
- person_code :
  pattern: ^\d{6}-\d{5}$
```

- **Obligātums.** Datu lauka vērtībai jābūt obligāti norādītai. Šajā gadījumā datu lauka obligātums ir biznesa likums un tas netraucē nodot datus uz DAGR. Piemērs:

```
name: validate_not_null
table:
fields:
- code !
```

Papildus datu kvalitātes kritēriju pārbaudi var pieslēgt un atslēgt datu devēja savienotāja vispārīgā konfigurācijā parametrā “data.validation”.

Datu kvalitātes kritēriju pārbaudes rezultāti tiek izvadīti datu devēja savienotāja žurnālpierakstos. Obligāto datu kvalitātes kritēriju kļūdas tiek izvadītas ar žurnālēšanas līmeni “ERROR”, savukārt neobligāto kritēriju kļūdas tiek izvadītas ar žurnālēšanas līmeni “WARNING”. Informācija par sekmīgi izpildītiem datu kvalitātes kritērijiem žurnālēta netiek.

5.6.2.5 Žurnālēšanas uzstādījumi

Datu devēja savienotājs veido žurnālpierakstus izmantojot *Logback* (<https://logback.qos.ch/>) ietvaru. Pēc noklusējuma žurnāla pieraksti tiek izvadīti konsolē (STDOUT) ar vidēju detalizācijas līmeni (INFO) un tiek nodoti DAGR portālam ar detalizācijas līmeni (INFO). Žurnālēšanas uzstādījumus var mainīt ar konfigurācijas faila (logback.xml) palīdzību - detalizētāka instrukcija atrodama: <https://logback.qos.ch/manual/configuration.html>.

Papildus tam, ja datu devēja savienotāja vispārīgajā konfigurācijā (application.conf) parametrā "report_problems.by_email" norādīta vērtība "true", tad informācija par kritiskajām kļūdām tiks nosūtīta arī datu devēja pieteikumā reģistrētajām kontaktpersonām uz e-pastu.

logback.xml noklusētā satura piemērs:

```
<configuration>

  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

    <!-- encoders are assigned the type
         ch.qos.logback.classic.encoder.PatternLayoutEncoder by default -->

    <encoder>

      <pattern>%d %highlight(%-5level) %-15logger{15} -%kvp- %msg%n%ex{0}</pattern>

    </encoder>

  </appender>

  <appender name="DAGR" class="dagr.DagrLogbackAppender">

    <filter class="ch.qos.logback.classic.filter.ThresholdFilter">

      <level>INFO</level>

    </filter>

    <encoder>

      <pattern>%d %highlight(%-5level) %-15logger{15} -%kvp- %msg%n%ex{0}</pattern>

    </encoder>

  </appender>

  <logger name="com.zaxxer.hikari" level="warn"/>

  <!-- <logger name="dagr.jdbc.sql" level="debug"/> -->

  <logger name="org.quartz" level="warn"/>

  <logger name="dagr.data" level="off"/>
```

```
<root level="info">

  <appender-ref ref="STDOUT" />

</root>

</configuration>
```

Uz DAGR portālu nodotos žurnāla pierakstus var apskatīties portāla sadaļā “Savienotāji”.

5.6.3 Debezium savienotājs

Alternatīva iepriekšējās nodaļās aprakstītajam datu devēja savienotājam ir datu nodošana uz DAGR izmantojot [Debezium](https://debezium.io/) tehnoloģisko platformu, kas ļauj datus izgūt no datu bāzu transakciju datnēm. *Debezium* tehnoloģija atbalsta populārākās SQL DBVS, tajā skaitā PostgreSQL, MySQL, Microsoft SQL Server, Oracle un Db2. Izmantojot *Debezium* savienotāju izmaiņas no avota datu bāzes tiek izgūtas reālajā laikā un nodotas uz DAGR kodola Kafka izmaiņu rindu.

Debezium savienotāja uzstādīšana ir veicama sekojot tehnoloģijas izstrādātāja instrukcijām (<https://debezium.io/documentation/reference/stable/connectors/index.html>). Katrai DBVS ir savas īpatnības, taču kopumā process sastāv no šādiem soļiem:

1. **Datu bāzes sagatavošana.** Jāpārlicinās, ka avota datu bāzes versija ir *Debezium* atbalstīto versiju sarakstā. Ja nav, tad jāpāriet uz jaunāku DBVS programmatūras versiju. Tālāk datu bāzei jāieslēdz izmaiņu fiksēšanas (*change data capture*) režīms. Visbeidzot jāpiešķir *Debezium* savienotājam nepieciešamās piekļuves tiesības.
2. **Jāuzstāda Debezium savienotājs.** Jālejupielādē *Debezium* savienotājs no tehnoloģijas izstrādātāja mājas lapas (<https://debezium.io/>) un jāuzstāda atbilstoši tehnoloģijas izstrādātāja instrukcijām.
3. **Jākonfigurē piekļuve avota datu bāzei.** *Debezium* savienotāja konfigurācijā jānorāda pieslēguma parametri datu bāzei, tajā skaitā IP adrese, ports, lietotājvārds, parole, u.c. DBVS specifiskie parametri. Konfigurācijā jānorāda saraksts ar tabulām, kuru izmaiņu dati nododami uz DAGR kodolu.
4. **Jākonfigurē piekļuve DAGR Kafka rindai.** *Debezium* savienotāja konfigurācijā jānorāda DAGR kodolā izvietotā Kafka servera pieslēguma parametri un lietotāja rekvizīti. Šī informācija ir pieprasāma no DAGR administratora.

Nemot vērā, ka *Debezium* savienotājs strādā reālā laika režīmā, izmaiņu nodošanas biežums atsevišķi nav konfigurējams.

5.7 Personas datu izmantošanas žurnālpieraksti

DAGR nodrošina iespēju datu ņēmējiem informācijas sistēmās veikto personu datu izmantošanas (atbilstoši VDAR prasībām) žurnālpierakstus nodot uz DAGR, lai tos tālāk padarītu pieejamus Latvijas iedzīvotājiem Latvija.lv portālā.

Lai šo iespēju izmantotu ir jāuzstāda datu devēja savienotājs un jāraksta ieraksti datu kopā “external_audit”. Šajā datu kopā ierakstus var ierakstīt jebkurš datu devējs, bet ierakstu tālākās piekļuves tiesības kontrolē DAGR pārvaldnieks.

Personas datu izmantošanas žurnālpierakstu datu kopas struktūra ir apskatāma:

<https://dagr.gov.lv/api-docs/vraa>

Datu devēja savienotāja datu struktūras kartējum (connector.yaml) piemērs personas datu izmantošanas žurnālpierakstu nodošanai:

```
table: source_table_example
columns:
- person_code          12
- reg_nr               11
- data_issuer          250
- action_time          dateTime
- facility_address
- data_processing_reason 250
- data_processing_purpose 250
- data_processing_type   250
- data_category         250

name: external_audit
table: source_table_example
key: person_code, reg_nr, data_system, action_time
fields:
- person_code
- reg_nr
- data_issuer
- action_time
- data_processing_reason
- data_processing_purpose
- data_processing_type
- data_category
```

5.8 Datu anonimizācija

Lai testēšanas nolūkiem sagatavotu datus, kas nesatur personas datus, DAGR nodrošina rīkus datu anonimizācijai. Izmantojot šos rīkus iespējams sagatavot datus, kas pēc struktūras ir vienādi ar faktiskajiem datiem, bet vairs nesatur ne tiešas ne netiešas atsauces uz reālām personām.

5.8.1 Datu anonimizācija izmantojot datu devēja savienotāju

Datu anonimizāciju veic datu devēja savienotājs atbilstoši datu anonimizācijas nosacījumiem (anonimizācijas veidnei), kas iekļauti datu devēja savienotāja konfigurācijas failā **connector.yaml**.

Datu anonimizācijas nosacījumi konfigurācijas failā iekļaujami šādā formā:

```
anonymizer: {datu_kopas_nosaukums}
min-batch-size: {izlases_izmers_vērtību_jaukšanai}
limit: {maksimālais_anonimizējamo_ierakstu_skaitis}
rules:
- {datu_lauka_nosaukums}: {anonimizācijas_veids} {veida_parametri}
- {datu_lauka_nosaukums}: {anonimizācijas_veids} {veida_parametri}
- ...
```

Parametrs min-batch-size nedrīkst būt mazāks nekā datu kopas ierakstu skaits.

DAGR atbalsta šādus anonimizācijas veidus:

- **Trokšņa pievienošana** (*add noise x y*). Skaitliska rakstura datu laukiem (skaitļiem, datumiem, laikiem) var pievienot gadījuma lieluma nobīdes. Metodei var norādīt vienu vērtību (*x*), tad nobīde tiks pievienota intervālā no $-x$ līdz $+x$. Vai divas vērtības (*x* un *y*), tad nobīde tiks pievienota intervālā no *x* līdz *y*. Šādā veidā modificēti dati neļauj identificēt personu pēc netiešajām pazīmēm. Piemēram, ja personu dzimšanas datumiem ir pievienots trokšnis, tad pēc personas dzimšanas datuma vairs nevar orientēties personas identificēšanai;
- **Apaļošana** (*round x*). Līdzīgi kā trokšņa pievienošana, arī apaļošana skaitliska rakstura datus (skaitļiem, datumiem, laikiem) padara mazāk izmantojamus personas precīzai identificēšanai. Apaļošanai parametrs *x* norāda apjomu līdz kuram vērtība apaļojama – vieniem, desmitiem, simtiem, utt;
- **Maskēšana** (*mask ***00000****). Lauku vērtības var aizklāt ar noteiktiem simboliem. Šī metode izmantojama teksta laukiem. Maskā simbols “*” nozīmē, ka simbola vērtība ir saglabājama, bet pārējie simboli pārraksta maskējamo vērtību. Piemērā norādītā maska ***00000*** vienpadsmit zīmju burtciparu kombinācijai vidējās četras zīmes aizvieto ar nullēm.
- **Personas koda ģenerēšana** (*generate person code* VAI *generate short person code*). Personas koda laukiem ir pielietojama jauna personas koda ģenerēšana, kas aizvieto datus esošo personas kodu ar neeksistējošu, bet korekti veidotu citu personas kodu. Ģenerētie personas kodi vienmēr sākas ar 32. Atkarībā no tā vai tiek norādīts atslēgas vārds “short”, personas kods tiek ģenerēts vai nu pilnajā formā ar domu zīmi (32****-****) vai saīsinātajā formā bez domu zīmes (32*****).
- **Jaukšana** (*shuffle*). Jebkura datu lauka vērtības var gadījuma kārtā samainīt starp datu kopas ierakstiem. Piemēram, personas datu kopai samainīt vietām dažādu personu vārdus. Vērtību jaukšana notiek nevis starp visas datu kopas ierakstiem, bet gan pa parametru *izlases_izmērs_vērtību_jaukšanai* norādīta apjoma grupām. Šis ļauj ietaupīt jaukšanas darbībai nepieciešamo operatīvās atmiņas apjomu. Normāls izlases izmērs ir daži tūkstoši ierakstu.
- **Aizvietošana**. Lauku vērtības var aizvietot ar noteiktu predefinētu vērtību vai kādu no vērtībām predefinētā sarakstā. Šī metode izmantojama gan skaitliskiem, gan teksta laukiem. Sintakse atšķiras no citiem anonimizācijas veidiem un tā pierakstāma šādā formā:
 - {datu_lauka_nosaukums}:
 - {vērtība_ar_kuru_aizvietot_1}
 - {vērtība_ar_kuru_aizvietot_1}
 - {vērtība_ar_kuru_aizvietot_n}

Datu anonimizāciju datu devēja savienotājs veic kā daļu no datu nodošanas procesa, anonimizētos datus ierakstot datu kopā, kas norādīta anonimizācijas nosacījumā (anonimizācijas veidnes) parametrā {datu_kopas_nosaukums}.

Praktisks piemērs datu devēja savienotāja konfigurācijas failam **connector.yaml** ar anonimizācijas veidni:

```
name:      tst_company_service_report_v2
table:     report r
key:       RegistrationNumber, Year, Id
fields:
- Id              = r.id
- Year            = r.year
- Year_anon_noise = r.year
```

```

- RegistrationNumber          = company.registration_number
- RegistrationNumber_anon_mask = company.registration_number
- CompanyName                 = company.name
- IsLicensed                   = company.is_licenced
- Status                       = r.status
- Status_anon_value           = r.status
- Submit                       = r.submitted
- Submitted_anon_value        = r.submitted
- VerifiedBy * [r.verifier_id=p.id]:
    table: person p
    fields:
    - Name                      = p.surname
    - Surname                   = p.name
    - Surname_anon_shuffle     = p.name
    - PersonCode                = p.code
    - PersonCode_anon_pk       = p.code
    - Client [s.receipient_id=c.id]:
        table: company c
        fields:
        - CompanyName           = c.name
        - RegistrationNumber     = c.registration_number

anonymizer: tst_company_service_report_v2
min-batch-size: 500
limit: 10000
rules:
- Year_anon_noise:                add noise -20 100
- RegistrationNumber_anon_mask:   mask ***00000***
- Submitted_anon_value:          round 1000000
- Status_anon_value:
    - anonymizēts saskaņots
    - anonymizēts iesniegts
    - anonymizēts cits
- VerifiedBy.PersonCode_anon_pk: generate person code
- VerifiedBy.Surname_anon_shuffle: shuffle

table: report
columns:
- id
- company.id
- year                                int
- status
- submitted                           dateTime
- verifier.id                         person.id

table: company
columns:
- id
- registration_number
- name

```

```
- is_licenced
```

```
table: person
```

```
columns:
```

```
- id
```

```
- code
```

```
- name
```

```
- surname
```

```
- created
```

```
dateTime
```

```
- last_modified
```

```
dateTime
```

Piemērā, sākot no apakšas, redzama definīcija trim tabulām no kurām tiek ņemti dati, tālāk seko anonimizācijas veidne uz visbeidzot sākumā ir tabulu datu lauku kartējums pret datu kopas datu struktūru.

Pēc nepieciešamības DAGR testa vidē anonimizētos datus var ierakstīt vai nu pamata datu kopā vai izveidot jaunu atsevišķu datu kopu anonimizētajiem datiem.

5.8.2 Datu anonimizācija caur DAGR pašapkalpošanās portālu

Datu anonimizāciju datu devēji var veikt arī DAGR pašapkalpošanās portālā sadaļas “Izplatīšana” apakš sadaļā “Anonimizācija”. Šajā sadaļā var reģistrēt datu anonimizācijas uzdevumus, kurus fonā izpilda specializēts DAGR centrālajā infrastruktūrā izvietots datu devēja savienotājs.

Anonimizācijas uzdevumi		
2024-05-23T09:15:30.156Z	sif_honourable_family_card_anon	pievienots
2024-05-23T09:13:16.404Z	sandbox/40003562863/tst_company_service_report_anon	pievienots
2024-05-20T12:15:24.283Z	sif_honourable_family_card_anon	izpildīts
2024-05-20T09:10:37.855Z	sif_honourable_family_card_anon	izpildē
2024-05-17T13:02:59.388Z	sif_honourable_family_card_anon	izpildē
2024-05-15T13:16:56.144Z	tst_company_service_report_anon	izpildē
2024-05-15T13:11:27.351Z	tst_company_service_report_anon	izpildē
2024-05-15T13:08:35.964Z	tst_company_service_report_anon_changes	izpildē
2024-05-15T10:30:43.646Z	rightsOfRepresentation	izpildē

Reģistrējot datu anonimizācijas uzdevumu DAGR pašapkalpošanās portālā ir jānorāda šāda informācija:

- **Datu devēja izvēle.** Ņemot vērā, ka viena fiziskā persona var pārstāvēt vairākus datu devējus, vispirms ir jāizvēlas datu devējs kura datu kopā tiks anonimizēta;
- **Avota datu kopa.** Izmaiņu servisa nosaukums datu kopai no kuras ņemami anonimizējamie dati. Šī vērtība ir jāizvēlas no saraksta;
- **Mērķa datu kopa.** Nosaukums datu kopai kurā ierakstāmi anonimizētie dati. Šī vērtība ir jāizvēlas no saraksta;
- **Anonimizācijas veidne.** Anonimizācijas nosacījumu saraksts. Detalizētāk skatīt sadaļā “5.8.1 Datu anonimizācija izmantojot datu devēja savienotāju”;

- **Lauku saraksts mērķa kopas ieraktu atslēgai.** Saraksts ar datu laukiem, kas veido mērķa datu kopas ierakstu primāro atslēgu.

DAGR > Anonimizācija > Jauns pieteikums

Jauns anonimizācijas uzdevums

DATU DEVĒJA IZVĒLE: Sabiedrības integrācijas fonds

DATU AVOTA KOPA: sif_honourable_family_card_changes

MĒRĶA DATU KOPA: sif_honourable_family_card_anon Latvijas Goda ģimenes apliecinājuma dati

ANONIMIZĀCIJAS VEIDNE:

```
anonymizer: sif_honourable_family_card_anon
min-batch-size: 500
rules:
  - vārds: shuffle
  - uzvārds: shuffle
  - izveidošanas_datums: round 1000
  - derīguma_termiņš: add noise -20 40
```

IDENTIFIKATORU LAUKI: personas_kods

Iesniegt anonimizācijas uzdevumu Atgriezties pie saraksta

Kad uzdevums reģistrēts, DAGR to izpilda fonā asinhronā režīmā. Standarta noslodzes gadījumā anonimizācijas uzdevums tiek paveikts dažu minūšu laikā. Anonimizācijas uzdevuma statuss aplūkojams anonimizācijas uzdevumu sarakstā.

Pirms reģistrēt anonimizācijas uzdevumu ir jāizveido datu kopa kurā anonimizētie dati tiks ierakstīti. Anonimizētie dati var tikt ierakstīti:

- **Priekšlikumu vides datu kopā (sandbox).** Priekšlikumu vidē datu kopas reģistrēt var pats datu devējs sadaļā “Izplatīšana” -> “Datu kopas”. Priekšlikumu vidē datu kopas ir pieejamas tikai pašam datu devējam, tajā skaitā datu devējam ir pieejami servisi gan datu ierakstīšanai, gan datu izgūšanai un apskatei. Priekšlikumu vide ir izmantojama anonimizācijas procesa testēšanai un anonimizēto datu priekšskatīšanai.
- **Pamata datu kopā (dataset).** Uz pamata vidi datu kopu no priekšlikumu vides pārceļ DAGR administrators pēc datu devēja lūguma. Pamata vidē anonimizēto datu kopai var piemērot pazīmi “*is_public_dataset: true*” – tādā gadījumā datu kopas dati būs pieejami reģistrētajiem datu ņēmējiem bez VIRSIS atļaujas.

5.9 Datu pievienošana un lejupielāde portālā

Datu devējiem pievienot, apskatīt un lejupielādēt savu datu kopu datus iespējams arī DAGR pašapkalpošanās portālā.

5.9.1 Datu lejupielāde

Savu datu kopu datus datu devējs var lejupielādēt vai nu Portālā vai izmantojot DAGR REST API.

Lai lejupielādētu savu datu kopu datus, jāautenticējas DAGR pašapkalpošanās portālā un jāatver DAGR pašapkalpošanās portāla REST API servisu aprakstu sadaļa: <https://dagr.gov.lv/api> Šajā sadaļā

jāatver apakš-sadaļa “Portāls” un jāizvēlas serviss GET /data_download/{dataset}. Tālāk šim servisam:

- jānospiež pogu “Try it out”,
- jāievada datu kopas nosaukumu datu laukā “dataset”,
- jāizvēlas JSON vai CSV datu laukā “Media Type”.

Datu kopas, kas ir ar hierarhisku datu struktūru, varēs lejupielādēt tikai JSON formātā, bet datu kopas ar viena līmeņa datu struktūru varēs lejupielādēt gan JSON, gan CSV formātā.

5.10 Datu nodošanas procesa atklūdošana

Ja datu nodošanas integrācijas testos datu devējs pamana DAGR darbības kļūdas vai neatbilstības dokumentācijai, lūgums, sazināties ar DAGR pārzini.

DAGR pārziņa rīcība ir DAGR pašdiagnostikas rīki, kas ļauj izsekot datu saņemšanas, datu sinhronizācijas starp DAGR datu centriem un datu izplatīšanas procesiem. Detalizētāka informācija par pašdiagnostikas rīku izmantošanu iekļauta DAGR administratora rokasgrāmatā.

6 Auditācijas uzlabojumi (DU-19)

6.1 Audita notikumi

DAGR auditē visus notikumus DAGR REST API pieprasījumu griezumā, gan tos kas veikti ar datu devēja vai datu saņēmēja teshiskajiem lietotājiem jeb savienojumiem, gan tos, kas veikti lietotājiem darbojoties DAGR pašapkalpošanās portālā.

DAGR datu devējiem un datu saņēmējiem ir iespēja aplūkot:

- **DAGR REST API auditu** par pieprasījumiem, kas veikti ar datu devēja vai datu saņēmēja savienojumiem jeb tehniskajiem lietotājiem;
- **datu pieprasījumu auditu** par datu izgūšanas pieprasījumiem, kas veikti ar datu saņēmēja savienojumu jeb tehnisko lietotāju. Šis ir DAGR REST API audita apakšgadījums. Atšķirībā no vispārējā gadījuma datu audita pieprasījumi ir redzami ne tikai pieprasījuma veicējam (datu saņēmējam), bet arī datu devējam kura datu kopas dati ir tikuši pieprasīti.

DAGR pārvaldniekiem ir iespēja aplūkot visus audita notikumus, gan DAGR REST API auditu, gan datu pieprasījumu auditu, gan lietotāju DAGR portālā veikto darbību auditu.

Audita dati ir pieejami DAGR pašapkalpošanās portālā un izmantojot DAGR REST API.

Audita notikumi tiek uzglabāti divus gadus (24 mēnešus).

6.1.1 Audita izgūšana caur REST API

Audita datus iespējams izgūt izmantojot servisu GET /audit ([servisa definīcija](#)) audita notikumu atlasei un servisu GET /audit_entity/{id} ieraksta satura apskatei. Abiem servisiem galvenē "Authorization" ir jānorāda datu saņēmēja vai datu devēja drošības talons (skatīt [autentifikācijas sadaļu](#)).

Audita notikumu atlasē serviss ir šādi datu saņēmēju un datu devēju izmantošanai paredzēti parametri:

- **timestamp_from (neobligāts)**, datums un laiks sākot no kura atlasīt audita notikumus;
- **timestamp_to (neobligāts)**, datums un laiks līdz kuram atlasīt audita notikumus;
- **request_ip_inet (neobligāts)**, IP adrese no kuras nākušo pieprasījumu audita notikumi atlasāmi;
- **request_ip_cidr (neobligāts)**, IP adreses CIDR filtrs. Piemēram, 1.2.0.0/16 atradīs visas adreses, kas sākas ar 1.2;
- **sender_id (daļēji obligāts)**, datu saņēmēja reģistrācijas numurs. Obligāti norādāms, ja serviss netiek izsaukts ar administratora drošības talonu un nav norādīts parametrs 'owner_id'. Ja servisu izmanto datu saņēmējs, tad paredzams, ka datu saņēmējs šeit norādīs pats savu reģistrācijas numuru.
- **owner_id (daļēji obligāts)**, datu devēja reģistrācijas numurs. Obligāti norādāms, ja serviss netiek izsaukts ar administratora drošības talonu un nav norādīts parametrs 'owner_id'. Ja servisu izmanto datu devējs, tad paredzēts, ka datu devējs šeit norādīs savu reģistrācijas numuru;
- **sender_type (neobligāts)**, pieprasījuma sūtītāja veids (iespējamās vērtības - portāla lietotājs, tehniskais lietotājs, tehnoloģiskais lietotājs, anonīms). Parametrs pamatā paredzēts administratoriem, jo datu devējiem un datu saņēmējiem pieejamajiem audita notikumiem sūtītāja veids vienmēr būs "tehniskais lietotājs". ***Piebilde:** Vēsturiski parametram bijušas citas iespējamās vērtības;*
- **subject_id (neobligāts)**, viens vai vairāki subjekta identifikatori (piemēram, personas kodi) pēc kuriem atlasīti dati, ja izmantots datu izgūšanas pēc identifikatora serviss. Ja norāda vairākas vērtības, tās jāatdala ar komatu;
- **client_id (neobligāts)** savienojuma jeb tehniskā lietotāja nosaukums ar kuru veikts pieprasījums;
- **is_success (neobligāts)**, vai pieprasījums ir bijis sekmīgi izpildīts;
- **permission_number (neobligāts)**, ar kādu atļauju izsaukts datu pieprasījums;
- **service_name (neobligāts)**, nosaukums servisam kurš ticis izsaukts;
- **action (neobligāts)**, darbības veids. Iespējamās vērtības ("pieslēgšanās", "atslēgšanās", "datu pieprasīšana (DP)", "datu atklāšana (DP)", "datu nodošana (DD)", "metadatu pieprasīšana (DD)", "atlasīšana", "izveidošana vai labošana", "dzēšana", "tehniskā procesa uzsākšana", "darbības priekšlikumu vidē", "kopskaita izgūšana", "list", "get", "delete", "insert", "update"). ***Piebilde:** Vēsturiski parametram bijis cits nosaukums - "action_type";*
- **key (neobligāts)** parametrs ar kādu ticis izsaukts serviss. Datu izgūšanas servisam pēc ID šī būs atslēga pēc kuras ieraksts meklēts.

Audita notikumu saraksta servisa izsaukuma piemērs:

```
curl --location
'https://dagr.gov.lv/audit?sender_id=40003562863&timestamp_to=2026-05-06T16%3A40%3A13.248Z&timestamp_from=2026-01-01T15%3A59%3A13.248Z'
\
--header 'Authorization: .....'
```

Audita notikumu saraksta servisa atbildes piemērs:

```
[
  {
    "id": 7701,
    "timestamp": "2026-03-05 13:24:30.104",
    "action_type": "atlasīšana",
    "request_ip4_addr": "127.0.0.6",
    "sender_id": "40003562863",
    "sender_type": "tehniskais lietotājs",
    "subject_id": "pievienots",
    "subject_type": "portāls",
    "owner_id": null,
    "owner_name": null
  },
  {
    "id": 236797,
    "timestamp": "2026-03-05 13:24:30.104",
    "action_type": "atlasīšana",
    "request_ip4_addr": "127.0.0.6",
    "sender_id": "40003562863",
    "sender_type": "tehniskais lietotājs",
    "subject_id": "pievienots",
    "subject_type": "portāls",
    "owner_id": null,
    "owner_name": null
  }
]
```

Audita notikuma ieraksta servisa izsaukuma piemērs:

```
curl --location 'https://dagr.gov.lv/audit_entity/41769336' \
--header 'Authorization: .....'
```

Audita notikuma ieraksta servisa atbildes piemērs:

```
{
  "id": 41769336,
  "timestamp": "2026-03-24 10:39:42.732",
  "related_activity_id": null,
  "session_id": "40003562863",
  "endpoint_id": "dcl.dagr-http-747b5b64f5-ffjnm",
  "action_type": "datu pieprasīšana (DP)",
  "request_ip4_addr": "159.148.76.1",
  "service_metadata":
  "{\n\"uri\": \"http://dagr.gov.lv/access_control_template/fpr_person_changes\", \"userAgent\": \"PostmanRuntime/7.52.0\", \"xForwardedFor\": \"159.148.76.1, 192.168.177.0\", \"action\": \"datu pieprasīšana (DP)\", \"service\": \"access_control_template\", \"key\": [\"fpr_person_changes\"], \"params\": null}\",
  "event_details":
  "{\n\"successful\": false, \"err_class\": \"java.lang.Exception\", \"err_msg\": \"Not Found fpr_person_changes\"}\",
  "sender_id": "40003562863",
  "sender_type": "tehniskais lietotājs",
  "sender_metadata":
  "{\n\"consumer_id\": \"40003562863\", \"consumer_name\": \"UNISO_test_connector_1\", \"subject\": \"S abiedrība ar ierobežotu atbildību \\\"UNISO\\\"\"}\",
```



```

"sender_connection_metadata": "{ \"user_type\": \"datu
patērētājs\", \"scope\": \"consumer\", \"permission_number\": null, \"purpose_of_use\": null }",
"subject_id": "fpr_person_changes",
"subject_type": "nenoteikts",
"subject_metadata":
"{ \"service_name\": \"access_control_template\", \"subject_keys\": [ \"fpr_person_changes\" ] }",
"owner_id": null,
"owner_name": null,
"client_id": "UNISO_test_connector_1",
"is_authority_of_inquiries": false
}

```

Atbildes datu skaidrojums:

- **action_time** – notikuma laiks;
- **related_identity** – saistītais reģistrācijas numurs (tipiski sakrīt ar datu ņēmēja reģ.nr.);
- **consumer_identity** – datu ņēmēja reģistrācijas numurs;
- **consumer_name** – datu ņēmēja savienotāja nosaukums;
- **consumer_id** – datu ņēmēja reģistrācijas numurs;
- **request_ip** – izsaukuma IP adrese;
- **request_session_id** – sesijas identifikators (drošības talona HASH);
- **request_info** – izsaukuma dati;
- **producer_reg_nr** – datu devēja reģistrācijas numurs;
- **producer_name** – datu devēja nosaukums;
- **permission_number** – atļaujas numurs;
- **purpose_of_use** – datu izmantošanas mērķis (atbilstoši atļaujai);
- **successful** – pazīme vai pieprasījums bijis veiksmīgs;
- **subject_id** – subjekta identifikators (piemēram, konkrēts personas kods);
- **subject_type** – datu subjekta tips (piemēram, “persona”, “dokuments”, “uzņēmums”);
- **endpoint_id** – DAGR komponente, kas izsniedza atbildi (atkļūdošanas nolūkiem).

6.2 Audita aplūkošana portālā

Audita datus iespējams apskatīties DAGR portāla sadaļā “Monitorings” -> “Audita notikumi”.

Līdzīgi kā izmantojot iepriekš aprakstīto servisu arī DAGR portālā ir šādas audita filtrēšanas iespējas:

- **Datu devējs** - filtrs ļauj meklēt iestādi sarakstā no iestādēm, kurām datu devēja pieteikumā lietotājs norādīts kā kontaktpersona, vai norādīt ierakstot iestādes reģistrācijas numuru;
- **Datu saņēmējs** – filtrs ļauj meklēt iestādi sarakstā no iestādēm, kurām datu ņēmēja pieteikumā lietotājs norādīts kā kontaktpersona, vai norādīt ierakstot iestādes reģistrācijas numuru;
- **Datums un laiks, no** – filtrs ļauj norādīt datumu un laiku sākot no kura audita ieraksti atlasāmi;
- **Datums un laiks, līdz** – filtrs ļauj norādīt datumu un laiku līdz kuram audita ieraksti atlasāmi;
- **Notikuma rezultāts** - filtrs ļauj norādīt vai atlasāmi sekmīgie, nesekmīgie vai visi notikumi;

- **VIRSIS atļaujas numurs** - filtrs ļauj meklēt tikai tos datu pieprasījumus, kas veikti izmantojot norādīto atļaujas numuru;
- **Servisa nosaukums** - filtrs ļauj meklēt tikai norādītā servisa pieprasījumus. Noderīgs lai filtrētu konkrētas datu kopas / servisa pieprasījumus;
- **Notikuma veids** – filtrs ļauj norādīt notikuma veidu par kuru atlasāmi audita ieraksti;
- **Lietotāja veids** – filtrs ļauj norādīt kāda veida lietotāju veiktās darbības atlasāmas;
- **Atslēgas** - filtrs ļauj norādīt atslēgu ar kādu serviss izsaukts. Datu izgūšanai pēc ID tas sakrītīs ar identifikatoru meklētajam ierakstam;
- **Pieprasījuma IP adrese** – filtrs ļauj norādīt IP adresi no kuras veikto pieprasījumu audita notikumi atlasāmi.
- **Klienta ID** – filtrs ļauj norādīt identifikatoru datu ņēmēja vai datu devēja savienotājam, kurš tika izmantots darbības veikšanai;
- **Subjektu identifikatori** – filtrs ļauj norādīt vienu vai vairākus subjekta identifikatorus, piemēram, personas kodus, par kuriem datu pieprasījumus datus atlasījis. Šis filtrs strādā tikai uz datu pieprasījuma veida audita notikumiem.

Obligāti aizpildāmie datu lauki - lai atlasītu datus, datu devēja pārstāvim jānorāda sava iestāde kā datu devējs un datu ņēmēja pārstāvim jānorāda sava iestāde kā datu ņēmējs.

Personas datu apstrādes gadījumu atlase. Viens no tipiskākajiem audita izmantošanas scenārijiem ir personas apstrādes gadījumu atlase. Lai to paveiktu ir izmantojami šādi filtri:

- Obligāti filtrā jānorāda sevis pārstāvētā iestāde – atbilstoši datu devējs, ja audita notikumus atlasa datu devēja pārstāvis vai datu ņēmējs, ja audita notikumus atlasa datu ņēmēja pārstāvis. Personām, kuras pārstāv gan datu ņēmēju/s gan datu devēju/s, jānorāda vismaz viens no abiem.
- Jānorāda personas kods subjekta identifikatora filtrā.

Atlasītos datus ir iespēja lejupielādēt faila formātā. Ir jāizvēlas faila formāts un jāspiež poga “Lejupielādēt”. Sistēma sagatavos un ļaus saglabāt datni, kas satur visus atlasītos ierakstus. Ja atlasīto ierakstu skaits pārsniedz 100 000 ierakstu (administratoram konfigurējams parametrs), tad faila lejupielāde pabeigsies ar atbilstošu kļūdu un ir jāsašaurina atlasē parametri, piemēram, samazinot atlasītā laika posma garumu.

7 DAGR satura centra izveide, atbilstoši lietotāju vajadzībām

Šī sadaļā ir izveidota atbilstoši Darba uzdevumam Nr. 4 “DAGR satura centra izveide, atbilstoši lietotāju vajadzībām” nodevumu prasībām pie vispārīgās vienošanās ID nr. VRAA EIS 2020/13/AK/CI-118-2-PKP pasūtījumam nr. VRAA/2024/35.

DAGR satura centra izveidei ir izvēlēts brīvi pieejams Wiki.JS produkts ar atvērto licenci, kas ļauj šo risinājumu izmantot gan komerciāliem mērķiem, gan pielāgot:

<https://github.com/requarks/wiki/blob/main/LICENSE>

Wiki.JS ražotāja dokumentācija: <https://docs.requarks.io/>

Wiki JS neietekmēs citu iesaistīto institūciju informācijas sistēmu piegādātāju saskarnes.

7.1 Paziņojumi datu saņēmējiem

Datu devējiem DAGR pašapkalpošanās portālā ir iespēja nosūtīt paziņojumus datu saņēmējiem, kuri tiks piegādāti uz saņēmēju kontaktpersonu e-pastiem no DAGR.

Paziņojumu izsūtīšanas funkcionalitāte ir pieejama portāla sadaļā “Paziņojumi datu saņēmējiem”. Šajā sadaļā ir aplūkojams saraksts ar nosūtītajiem paziņojumiem un ir iespēja pievienot jaunu vai rediģēt eksistējošu paziņojumu, ja tas vēl nav izsūtīts.

Paziņojumiem iespējams norādīt šādus datu laukus:

- **Adresāts.** Saraksts ar adresātiem, kur katru raksturo e-pasts/vārds/uzvārds. Adresātus var norādīt šādos veidos:
 - Izvēloties iestādi (datu ņēmēju). Šajā gadījumā paziņojums tiks nosūtīts visām iestādes kontaktpersonām;
 - Izvēloties datu kopu. Šajā gadījumā, izmantojot VIRSIS atļauju datus, paziņojums tiks nosūtīts visu to iestāžu kontaktpersonām, kam ir atļauja piekļūt datu kopas datiem;
 - Izvēloties lietotāju (fiziskās personas). Šajā gadījumā paziņojums tiks nosūtīts specifiskām izvēlētajām datu saņēmēja kontaktpersonām.
- **Pārstāvēto datu devēju.** Ja lietotājs var pārstāvēt vairākus datu devējus, tad jāizvēlas kura datu devēja vārdā paziņojums tiks nosūtīts.
- **Nosaukumu.** Paziņojuma nosaukumu jeb subject;
- **Saturu.** Paziņojuma saturs jeb body;
- **Izsūtīšanas datumu un laiku.** Ja datu lauks tiek atstāts tukšs, tad paziņojums tiks nosūtīts uzreiz. Ja datums un laiks tiek norādīts, tad paziņojums tiks nosūtīts tikai norādītajā laikā.
- **Datu kopa.** Opcionāli var norādīt uz kuru datu kopu paziņojums attiecināms.

7.2 Apkārtraksti

DAGR administratoram DAGR pašapkalpošanās portāla pārvaldības sadaļā ir iespēja nosūtīt apkārtrakstus DAGR lietotājiem, kuri tiks piegādāti uz datu devēju un datu saņēmēju kontaktpersonu e-pastiem no DAGR.

Apkārtrakstu izsūtīšanas funkcionalitāte ir pieejama portāla sadaļā “Apkārtraksti”. Šajā sadaļā ir aplūkojams saraksts ar nosūtītajiem apkārtrakstiem un ir iespēja pievienot jaunu vai rediģēt/dzēst eksistējošu apkārtrakstu, ja tas vēl nav izsūtīts.

Apkārtrakstiem iespējams norādīt šādus datu laukus:

- **Adresāts.** Saraksts ar adresātiem, kur katru raksturo e-pasts/vārds/uzvārds. Adresātus var norādīt šādos veidos:
 - Izvēloties iestādi (datu ņēmēju vai datu devēju). Šajā gadījumā apkārtraksts tiks nosūtīts visām iestādes kontaktpersonām;

- Izvēloties datu kopu. Šajā gadījumā, izmantojot VIRSIS atļauju datus, apkārtraksts tiks nosūtīts visu to iestāžu kontaktpersonām, kam ir atļauja piekļūt datu kopas datiem;
- Izvēloties lietotāju (fiziskās personas). Šajā gadījumā apkārtraksts tiks nosūtīts specifiskām izvēlētajām datu saņēmēja vai datu devēja kontaktpersonām.
- **Nosaukumu.** Paziņojuma nosaukumu jeb subject;
- **Saturu.** Paziņojuma saturu apkārtraksta šablonā;
- **Izsūtīšanas datumu un laiku.** Ja datu lauks tiek atstāts tukšs, tad paziņojums tiks nosūtīts uzreiz. Ja datums un laiks tiek norādīts, tad paziņojums tiks nosūtīts tikai norādītajā laikā.

Apkārtraksta saturs tiek ievietots HTML šablonā un pirms apkārtraksta izsūtīšanas ir iespēja priekšskatīt apkārtrakstu kāds tas izskatīsies e-pastā.

DAGR administratoram ir iespēja apkārtrakstu rediģēt DAGR pašapkalpošanās portāla pārvaldības sadaļā "Apkārtraksta šablons". Tajā skaitā ir iespējams atgriezties pie sākotnējā (iebūvētā) šablona.

8 Pielikums – papildus paraugi

8.1 Datu kopu definīcijas

Šajā sadaļā iekļauti datu kopu definīciju piemēri. Datu kopu definēšana aprakstīta sadaļā “5.3 Izplatāmo datu kopu definēšana”. Piemēriem ir tikai ilustratīva nozīme un lasīšanas brīdī tie var būt neaktuāli salīdzinājumā ar DAGR produkcijas vidi.

Piemēra datu kopa ar filtriem un papildus meklēšanas atslēgām

```
data_owner_code: "40003562863"
data_owner_name: SIA UNISO
data_namespace: uniso
name: uniso_filter_test
key: code, year, change_date, status
search:
- name: surname
  fields: surname
  description: Meklēšana pēc uzvārda
- name: date
  fields: change_date
  description: Meklēšana pēc datuma
data_description: datu kopu piemērs filtru vajadzībām
fields:
- code : Personas kods
- name : Personas vārds
- surname : Personas uzvārds
- year int : Ieraksta gads
- change_date dateTime : Ieraksta izmaiņu datums
- status (new, declined, accepted) : Ieraksta statuss
- detail : Detalizēts apraksts
```

Goda ģimenes apliecības turētāji no Sabiedrības integrācijas fonda

```
name: sif_honourable_family_card
data_description: "Latvijas Goda ģimenes apliecību dati"
key: personas_kods
fields:
- numurs : Apliecības numurs
- vārds : Apliecības turētāja vārds
- uzvārds : Apliecības turētāja uzvārds
- personas_kods : Apliecības turētāja personas kods
- tips : Apliecības tips
- izveidošanas_datums date : Apliecības izveidošanas datums
- derīguma_termiņš date : Apliecības derīguma termiņš
- ir_anulēta boolean : Pazīme, ka apliecība ir anulēta
- hash : Apliecības kontrolsumma. 30 ciparu skaitlis, kas sākas ar 1234, turpinās ar apliecības numuru un noslēdzas ar 14 ciparu gadījuma skaitli.
```

Dabas resursu nodokļa maksātāji

```
name: vvd_drn_maksataji
data_description: VVD DRN maksātāji
key: (code, permit_num, drn_rec_id, permit_rec_id)
fields:
- code 50 : Maksātāja reģistrācijas kods / personas kods
- non_resident int : Rezidents vai nerezidents
- name 250 : Maksātāja nosaukums/ vārds, uzvārds
- address_code
- facility_address : Maksātāja adrese Atļaujas objekta adrese
- drn_date_from date : DRN maksātājs no
```

- drn_date_to date : DRN maksātājs līdz
- permit_type : Atļaujas veids
- permit_num : Atļaujas numurs
- permit_issue_date dateTime : Atļaujas izsniegšanas datums
- permit_valid_from dateTime : Atļauja derīga no
- permit_valid_to dateTime : Atļauja derīga līdz
- permit_cancel_date dateTime : Atļaujas anulēšanas datums
- permit_status int : Atļaujas statuss sistēmā
- drn_rec_id long : Atbilst atbrīvojuma perioda ieraksta DB identifikatoram, jo vienai atļaujai var būt vairāk par vienu drn periodu
- permit_rec_id long : Atbilst atļaujas ieraksta DB identifikatoram, jo ir vairākas spēkā esošas atļaujas ar vienādiem numuriem.

Personas izglītība no Valsts izglītības informācijas sistēmas (VIIS)

```

name:      viis_person_education
key:       PersonCode
data_description: "Personas izglītība"
fields:
- Id int: izglītības dokumenta ID
- PersonCode: Personas kods
- PersonName: personas vārds
- PersonOtherNames: personas citi vārdi
- PersonSurname: personas uzvārds
- EducationDegreeId int: Izglītības pakāpes ID
- EducationDegree: Izglītības pakāpe
- EducationTypeId int: Izglītības veida ID
- EducationType: Izglītības veids
- EducationDocumentTypeId int: Dokumenta veida ID
- EducationDocumentType: izglītības dokumenta veids
- InstitutionName: Izglītības iestādes nosaukums
- DocumentIssueDate date: izglītības dokumenta izsniegšanas datums
- EducationProgramName: izglītības programma
- Specialities * :
  comments: Specialitāte
  table:
  fields:
  - Name
- Qualifications * :
  comments: Kvalifikācija
  table:
  fields:
  - Name
- Series: sērija
- Number: numurs
- CourseName: Kursu nosaukums
- LearnedLessonCount int: stundu skaits
- LearningStartDate dateTime: mācību sākuma datums
- MainEducationTypeId int:
- MainEducationType:
- ModifiedOn dateTime: izmaiņu datums

```

Personas aktuālās mācības no Valsts izglītības informācijas sistēmas (VIIS)

```

name:      viis_person_study
key:       PersonCode
data_description: "Personas aktuālās mācības"
fields:
- Id int : personas piederības ID
- PersonCode : Personas kods
- PersonName : personas vārds
- PersonOtherNames : personas citi vārdi
- PersonSurname : personas uzvārds
- PersonBirthDate dateTime : dzimšanas datums
- PersonGender : int dzimums
- PersonCitizenshipCode : pilsonības kods
- PersonCitizenship : pilsonība
- PersonIsDisabled boolean : ir invalīds
- PersonModifiedOn : izmaiņas

```

- ParticipationTypeId	int	: piederības veida ID
- ParticipationType		: piederības veids
- DateFrom	dateTime	: piederības datums no
- DateTill	dateTime	: piederības datums līdz
- ModifiedOn	dateTime	: piederības izmaiņas
- InstitutionRegistrationNumber		: iestādes reģistra numurs
- InstitutionName		: iestādes nosaukums
- StudyParticipations * :		
table:		
fields:		
- Id	int	: klases piederības ID
# Name		
- DateFrom	dateTime	: klases piederības datums no
- DateTill	dateTime	: klases piederības datums
līdz		
- AcademicGroupId	int	: Semestra ID
- AcademicGroupName		: Semestra nosaukums
- AcademicLearningSubStatusId	int	: Studiju apakšstatuss ID
- AcademicLearningSubStatusName		: Studiju apakšstatuss
- ClassId	int	: Klases ID
- ClassName		: Klases nosaukums
- ClassGrade	int	: klases pakāpe
- ClassParalel		: klases paralēle
- EducationProgramLicenceId	int	: klases piederības izglītības
programmas licences ID		
- EducationProgramLicenceNumber		: klases piederības izglītības
programmas licences numurs		
- InterestEducationProgramId	int	: Interese programmas ID
- InterestEducationProgramSubTypeCode		: Interese programmas
apakštipa kods		
- InterestEducationProgramSubTypeName		: Interese programmas
apakštipa nosaukums		
- KindergartenGroupId	int	: grupiņas ID
- KindergartenGroupName		: Grupiņas nosaukums
- LearningFormTypeId	int	: Izglītības ieguves formas ID
- LearningFormTypeName		: Izglītības ieguves forma
- StudyFormTypeId	int	: Studiju veida ID
- StudyFormTypeName		: Studiju veids
- UniversalGroupId	int	: interešu izglītības grupas
ID		
- UniversalGroupName		: Interese iqlitības grupas
nosaukums		
- UniversalGroupTypeId	int	: interešu izglītības grupas
veida ID		
- UniversalGroupTypeName		: interešu izglītības grupas
veida nosaukums		
- VocationalGroupId	int	: Grupas ID
- VocationalGroupName		: Grupas nosaukums

Adrešes no Valsts adrešu reģistra

```

name: vzd_address
data_description: "Valsts adrešu reģistra dati"
key:
- adr_code
fields:
- adr_code : Adrešes kods (indekss)
- address : Adrese
- code int : Adrešes kods
- cieCode long : Ciema / mazciema kods
- cieName : Ciema / mazciema nosaukums
- dzvCode long : Telpu grupas kods
- dzvName : Telpu grupa
- ielCode long : Ielas kods
- ielName : Ielas nosaukums
- lksCoordLong decimal : LKS ģeogrāfiskais garums
- lksCoordLat decimal : LKS ģeogrāfiskais platums
- nltCode long : Ēkas, apbūvei paredzēta zemes vienības (NLT) kods
- nltName : Ēka, apbūvei paredzēta zemes vienība (NLT)

```

```

- novCode long : Novada kods
- novName : Novada nosaukums
- pagCode long : Pagasta kods
- pagName : Pagasta nosaukums
- pilCode long : Pilsētas kods
- pilName : Pilsētas nosaukums
- typ long : Adresācijas objekta tips
- pilAtvk : Pilsētas ATVK kods
- novAtvk : Novada ATVK kods
- pagAtvk : Pagasta ATVK kods
- zipCode : Pasta indekss
- irAdrese boolean : Ir adrese (citas adreses neatsaucas uz šo adreses ierakstu)
- history * address_history:
  - readonly

```

NILLTPFN likuma subjektu dati no Uzņēmumu reģistra

```

name: subjectOfLaw
key: identityNumber
data_description: NILLTPFN likuma subjektu dati
fields:
- identityNumber : Personas kods
- forename : Personas vārds
- surname : Personas uzvārds
- organization * :
  comments : Organizācija, kuru persona pārstāv
  fields:
  - id integer : Organizācijas ID
  - registrationNumber : Organizācijas reģistrācijas numurs
  - name : Organizācijas nosaukums
  - userRole * string : Personas loma organizācijā
  - supervisoryAuthority * :
    comments : Organizācijas uzraugošā institūcija
    fields:
    - id integer : Uzraugošās institūcijas ID
    - registrationNumber : Uzraugošās institūcijas reģistrācijas numurs
    - name : Uzraugošās institūcijas nosaukums

```

Tiesību subjekts no Uzņēmumu reģistra

```

name: ur_legal_entity
key:
- registrationNumber
fields:
- activitiesDetails :
  table:
  comments : Darbības veidi/mērķi. Satur Nace darbības veidus
un/vai Darbības jomas.
  fields:
  - areasOfActivityOfAssociationsFoundations * :
    table:
    comments : Darbības jomas.
    fields:
    - id int : Ieraksta identifikators
    - dateFrom date : Datums, kad pievienots
    - dateTo date : Datums, kad izslēgts
    - isAnnulled boolean : Vai ieraksts anulēts
    - lastModifiedAt dateTime : Pēdējo izmaiņu datums un laiks
    - registeredOn date : Ieraksta spēkā stāšanās datums
    - removedOn date : Ieraksta izslēgšanas datums
    - type : Biedrību un nodibinājumu darbības jomas kods.
Klasifikatora tips - "area_of_activity".
  - goals * 250 : Darbības mērķi
  - naceTypes * :
    table:
    comments : Nace darbības veidi.
    fields:
    - id int : Ieraksta identifikators

```



```

- lastModifiedAt          dateTime : Pēdējo izmaiņu datums un laiks
- naceCode                5 : NACE kods
- removedOn              date : Ieraksta izslēgšanas datums
- territoryOfActivity     250 : Darbības teritorija (religiskām organizācijām).
- typesOfActivity         * 250 : Galvenie darbības veidi
- address                 ur_le_address :
  comments                : Juridiskā adrese.
- arbitrationCourtDetails :
  table:
  comments                : Šķīrējtiesas.
  fields:
  - dateOfEffectiveRulesOfProcedure date : Reglamenta reģistrācijas datums
- atvkCode                7 : Administratīvo teritoriju un teritoriālo vienību
klasifikatora (ATVK) kods
- beneficialOwners * :
  table:
  comments                : Patiesie labuma guvēji. Saraksts. Satur datus
par fiziskajām personām un kontroles veidiem.
  fields:
  - dateFrom              date : No datums
  - dateTo                date : Līdz datumam
  - id                    int : Ieraksta identifikators
  - isAnnulled            boolean : Vai ieraksts anulēts
  - isMinor               boolean : Pazīme, ja PLG ir nepilngadīgs vai persona,
kuras dati ir ierobežotas pieejamības. Ja vērtība 'true', tad dati par šo personu netiks
atgriezti.
  - lastModifiedAt        dateTime : Ieraksta pēdējo izmaiņu datums un laiks
  - meansOfControl * ur_le_meansOfControl :
    - Kontroles veidi.
  - naturalPerson         ur_le_naturalPerson :
    - Satur atbilstošos laukus no struktūras "Fiziska persona"
  - note                  : Piezīmes
  - registeredBecauseOf    250 : Sākuma datuma reģistrācijas veids. Klasifikatora
tips - "registration_removal_reason".
  - registeredOn           date : Ieraksta izdarišanas datums
  - removedBecauseOf       250 : Beigu datuma reģistrācijas veids. Klasifikatora
tips - "registration_removal_reason".
  - removedOn             date : Izslēgšanas datums
- beneficialOwnershipStatuses * :
  table:
  comments                : Patieso labuma guvēju speciālie statusi.
  fields:
  - dateFrom              date : No datums
  - dateTo                date : Līdz datumam
  - id                    intString : Ieraksta identifikators
  - isAnnulled            boolean : Vai ieraksts anulēts
  - lastModifiedAt        dateTime : Ieraksta pēdējo izmaiņu datums un laiks
  - registeredOn           date : Ieraksta izdarišanas datums
  - removedOn             date : Ieraksta izslēgšanas datums
  - type                  250 : "PLG veids. Klasifikatora tips -
"various_status". Iespējami šādi PLG veidi:
BENEFICIAL_OWNER_CONTROL,_RESULTS_FROM_,SHAREHOLDERS_STATUS,BENEFICIAL_OWNER_CANNOT,_BE_IDENTI
FIED"
- branches * :
  table:
  comments                : Filiāles.
  fields:
  - id                    intString : Ieraksta identifikators
  - dateFrom              date : Datums no
  - dateTo                date : Datums līdz
  - lastModifiedAt        dateTime : Pēdējo izmaiņu datums un laiks
  - legalEntity           ur_le_legalEntity :
    - Satur atbilstošos laukus no struktūras "Juridiska persona"
- branchOf * :
  table:
  comments                : Mātes uzņēmumi. Satur datus par Juridiskām
personām.
  fields:
  - id                    intString : Ieraksta identifikators
  - dateFrom              date : Datums no
  - dateTo                date : Datums līdz

```

```

- lastModifiedAt          dateTime : Pēdējo izmaiņu datums un laiks
- legalEntity             ur_le_legalEntity :
    - Satur atbilstošos laukus no struktūras "Juridiska persona"
- cleanedShortName        250 : Saīsināts (bez norādes uz tiesisko formu) un no
diakritiskajām zīmēm attīrīts nosaukums lieliem burtiem.
- commercialEntityDetails :
    table:
    comments                : Komercuzņēmuma dati. Satur datus par
ierobežotiem partneru ieguldījumiem, Pamatkapitālu, Izšķirošajām ietekmēm, kā arī papildus
informāciju ZS.
    fields:
    - decisiveInfluenceNotices * :
        table:
        comments                : Izšķirošā ietekme. Satur valdošos un atkarīgos
uzņēmumus.
        fields:
        - dateOfAgreement        date : Datums, kad ietekme izveidota
        - dateOfAmendments       date : Ietekmes izmaiņu datums
        - dependentEntity :
            table:
            comments                : Atkarīgais uzņēmums.
            fields:
            - registrationNumber    11 : Atkarīgā uzņēmuma reģistrācijas numurs
            - currentName          250 : Atkarīgā uzņēmuma pilnais nosaukums
            - links                ur_le_links :
                - Satur atbilstošos laukus (self) no struktūras "Saīsināts",
- dominantEntity :
    table:
    comments                : Valdošais uzņēmums.
    fields:
    - naturalPerson ur_le_naturalPerson :
        - Satur atbilstošos laukus no struktūras "Fiziska persona",
    - legalEntity ur_le_legalEntity :
        - Satur atbilstošos laukus no struktūras "Juridiska persona"
- id                        int : Ieraksta identifikators
- isAnnulled               boolean : Vai anulēts
- lastModifiedAt           dateTime : Ieraksta pēdējo izmaiņu datums un laiks
- note                     : Piezīme
- noteOnRemoval            : Līguma izbeigšanas pamats
- registeredOn             date : Ieraksta izdarīšanas datums
- removedOn               date : Ieraksta izslēgšanas datums
- type                    250 : Izšķirošās ietekmes veids. Klasifikatora veids -
"decisive_influence".
- typeDetails * : Konkurnu izšķirošās ietekmes veidu saraksts. Klasifikatora veids -
"decisive_influence_details".
- equityCapitals * :
    table:
    comments                : Pamatkapitāls.
    fields:
    - amount                decimal 14 2 : Summa
    - changeReason          250 : Ieraksta veids. Klasifikatora tips -
"change_reason".
- currency                 3 : Vērtības valūta
- id                      int : Ieraksta identifikators
- isAnnulled              boolean : Vai anulēts
- lastModifiedAt          dateTime : Ieraksta pēdējo izmaiņu datums un laiks
- numberOfShares          decimal 13 0 : Dalībniekam piederošo daļu skaits
- registeredOn            date : Ieraksta izdarīšanas datums
- removedOn              date : Ieraksta izslēgšanas datums
- shareNominalValue       decimal 14 1 : Daļas nominālvērtība
- type                    250 : Pamatkapitāla veids. Klasifikatora tips -
"capital".
- farmDetails :
    table:
    comments                : Zemnieku saimniecības dati. Satur zemes
īpašumtiesību izmaiņas.
    fields:
    - areaInHectares        decimal 15 3 : Zemnieku saimniecībām - zemes platība hektāros;
    - landOwnershipChanges * :
        table:
        comments                : Zemes īpašumtiesību izmaiņas.

```

```

        fields:
        - address                250 : Atrašanās vieta
        - areaChange              decimal 15 3 : Platības izmaiņas
        - deedIssuedOn            date : Akta datums
        - deedNumber              12 : Lēmuma numurs
        - id                      int : Ieraksta identifikators
        - isAnnulled              boolean : Vai ieraksts anulēts
        - lastModifiedAt          dateTime : Ieraksta pēdējo izmaiņu datums un laiks
        - rightsToLandUnderDeed   20 : Aktā noteiktās tiesības uz zemi. Klasifikatora
tips - "rights_to_land".
        - registeredOn            date : Ieraksta izdarišanas datums
        - removedOn              date : Ieraksta izslēgšanas datums
    - insolvencies * :
        table:
        comments                  : Maksātnespējas.
        fields:
        - dateFrom                date : Datums no
        - dateTo                  date : Datums līdz
        - governedBy              int : Regulējošā tiesību akta identifikators, kas
norāda, uz kuru likumu maksātnespējas ieraksti ir attiecināmi. Klasifikatora tips -
"legislation".
        - isAnnulled              boolean : Vai ieraksts anulēts
        - lastModifiedAt          dateTime : Ieraksta pēdējo izmaiņu datums un laiks
        - links                   ur_le_links :
            - Satur atbilstošos laukus (self) no struktūras "Saites"
        - registeredOn            date : Ieraksta izdarišanas datums
        - removedOn              date : Ieraksta izslēgšanas datums
    - limitedPartnerContribution :
        table:
        comments                  : Komandīta ieguldījums (Komanditsabiedrībām).
        fields:
        - amount                  decimal 14 2 : Summa
        - currency                3 : Vērtības valūta
- formationDetails :
    table:
    comments                     : Dibināšanas informācija.
    fields:
    - dateOfEffectiveArticlesOfAssociation date : Statūtu parakstīšanas datums
    - durationIndefinite          boolean : Pazīme, ka objekta darbības laiks ir beztermiņa
    - durationUntil               date : Atļauja līdz
    - durationUntilObjectsAchieved boolean : Pazīme, ka objekts nodibināts līdz mērķis
sasniegts
    - establishedOn               date : Dibināšanas datums (attiecas uz reģistriem
'register' -
ASSOCIATIONS_AND_FOUNDATIONS,RELIGIOUS_ORGANISATIONS_AND_INSTITUTIONS,POLITICAL_PARTIES).
    - objects                    250 : Dibināšanas/izveides mērķi
    - registrationRenewedOn       date : Pārreģistrēšanas datums (religiskām
organizācijām).
    - registrationTermUntil       date : Reģistrācijas termiņš (religiskām
organizācijām).
    - isAnnulled                 boolean : Pazīme, vai ieraksts ir anulēts.
    - lastModifiedAt             dateTime : Pēdējo izmaiņu datums un laiks
- liquidations * :
    table:
    comments                     : Likvidācijas.
    fields:
    - id                         int : Ieraksta identifikators
    - completionExpectedOn        date : PPI Likvidācijas datums (paredzamais). Pie
pabeigtas likvidācijas, šis ir null
    - endedOn                    date : Likvidācijas procesa izbeigšanas datums
    - groundsForContinuation      : Likvidācijas izbeigšanas pamatojums
    - groundsForLiquidation       : Likvidācijas procesa uzsākšanas pamatojums
    - isAnnulled                 boolean : Vai ieraksts anulēts
    - lastModifiedAt             dateTime : Ieraksta pēdējo izmaiņu datums un laiks
    - legalAct                   ur_le_legalAct :
        - (tikai PPI) Tiesību akts likvidācijas pamatā.
    - registeredOn                date : Likvidācijas reģistrācijas datums
    - removedOn                  date : Ieraksta izslēgšanas datums. (Ja šis datums ir
aizpildīts, tad ieraksts vairs nav aktuāls/ vairs nav spēkā.)
    - resultType                 250 : Likvidācijas rezultāts. Klasifikatora tips -
"liquidation_suspension_prohibition_result".

```

```

- startedOn                                date : Likvidācijas procesa uzsākšanas datums
- type                                    20 : Likvidācijas procesa veids. Klasifikatora tips -
"liquidation_suspension_prohibition".
- links                                  ur_le_links :
- LE izsaukumu saites.
- legalName                              250 : Pilnais nosaukums
- name                                    250 : Iestāžu nosaukums
- members * :
  table:
    comments                                : Dalībnieki. Satur datus par Fiziskām personām,
Juridiskām personām, dalībnieku ieguldījuma daļām un Eiropas ekonomisko interešu grupas (EEIG)
informāciju.
    fields:
      - id                                  int : Ieraksta identifikators.
      - dateFrom                            date : Datums, kad dalībnieku reģistra nodalījums
sastādīts (dalībnieks ir iekļauts dalībnieku reģistrā)
      - dateTo                              date : Datums, kad sastādīts dalībnieku reģistra
nodalījums, no kura izriet, ka daļas ir atsavinātas (vai arī mainījies daļu skaits, vai
nominālvērtība vai kādas citas izmaiņas)
      - eeigPartyDetails :
        table:
          comments                            : Eiropas ekonomisko interešu grupas (EEIG)
informācija. Attiecas tikai uz EEIG reģistru.
          fields:
            - isReleasedFromPriorLiabilities boolean : Pazīme - biedram piešķirtais atbrīvojums no
saistībām, kas radušās pirms viņa uzņemšanas grupā
            - mainLocationAddress            250 : Eiropas ekonomisko interešu grupas biedra
(juridiskas personas) vadības atrašanās vietas adrese
            - mainLocationCountry            3 : Vadības atrašanās vietas valsts
            - isAnnulled                      boolean : Vai dalībnieks anulēts
            - isPersonallyLiable              boolean : Vai dalībnieks ir personiski atbildīgs (PS, KS).
            - lastModifiedAt                  dateTime : Pēdējo izmaiņu datums un laiks
            - legalEntity                    ur_le_legalEntity :
              - Satur atbilstošos laukus no struktūras "Juridiska persona"
            - naturalPerson                  ur_le_naturalPerson :
              - Satur atbilstošos laukus no struktūras "Fiziska persona"
            - note                            : Piezīmes
            - registeredBecauseOf             250 : Iecelšanas veids. Klasifikatora tips -
"registration_removal_reason".
            - registeredOn                    date : Ieraksta izdarīšanas datums. Datums, kad valsts
notārs dalībnieku reģistru pievienoja lietai.
            - removedBecauseOf               250 : Atbrīvošanas veids. Klasifikatora tips -
"registration_removal_reason".
            - removedOn                      date : Ieraksta izslēgšanas datums. Datums, kad valsts
notārs pievienojis jaunu dalībnieku reģistra nodalījumu.
            - representative                  ur_le_naturalPerson :
              - Dalībnieka pārstāvja informācija. Satur laukus no struktūras "Fiziska persona"
            - shareholderDetails :
              table:
                comments                        : Dalībnieka ieguldījuma daļas. Satur Kopdaļu
īpašnieku sarakstu.
              fields:
                - numberOfShares              decimal 12 1 : Dalībniekam piederošo daļu skaits.
                - shareNominalValue          decimal 14 2 : Daļas nominālvērtība
                - shareCurrency               3 : Daļu vērtības valūta
                - inPercent                   decimal 10 5 : Daļu procentuālā daļa
                - jointOwners * :
                  table:
                    comments                    : Kopdaļas īpašnieki. Satur informāciju no Fizisku
vai Juridisku personu struktūras.
                  fields:
                    - id                      int : Ieraksta identifikators
                    - isAnnulled              boolean : Vai ieraksts anulēts
                    - lastModifiedAt          dateTime : Pēdējo izmaiņu datums un laiks
                    - legalEntity            ur_le_legalEntity :
                      - Satur laukus ('registrationNumber' - UR tiesību subjektam,
'externalObjectNumber'- ārvalstniekiem, 'foreignRegistrationNumber' - ārvalstniekiem, 'name',
'address' un 'country') no struktūras "Juridiska persona"
                      - naturalPerson        ur_le_naturalPerson :
                        - Satur laukus ('latvianIdentityNumber', 'name', 'country') no struktūras
"Fiziska persona"

```

```

- registeredOn          date : Ieraksta izdarīšanas datums
- removedOn            date : Ieraksta izslēgšanas datums
- officers *:
  table:
    comments              : Amatpersonas. Satur datus par Fiziskām personām,
Juridiskām personām, kā arī pārstāvību apjomu.
    fields:
      - appointedOn      date : Ieraksta izdarīšanas par amatpersonas iecelšanu
datums
      - governingBody    250 : Institūcija, piemēram, valde vai padome.
Klasifikatora tips - "governing_body".
      - id               int : Ieraksta identifikators
      - isAnnulled       boolean : Vai ieraksts anulēts
      - lastModifiedAt   dateTime : Pēdējo izmaiņu datums un laiks
      - legalEntity      ur_le_legalEntity :
        - Tiesību subjekta dati (Likvidācijas gadījumā likvidators var būt juridiska persona).
Elements. Satur atbilstošos laukus no struktūras "Juridiska persona"
      - naturalPerson    ur_le_naturalPerson :
        - Fiziskas personas dati. Elements. Satur atbilstošos laukus no struktūras "Fiziska
persona"
      - note              : Piezīmes
      - position          250 : Amats. Klasifikatora tips - "position".
      - positionDetails   250 : Amatpersonas amats detalizēti. Tas papildina
"position", īpaši gadījumos, ja amats nav klasifikatorā.
      - registeredBecauseOf 250 : Ierakstīšanas veids. Klasifikatora tips -
"registration_removal_reason".
      - registeredOn      date : Ieraksta izdarīšanas datums
      - removedBecauseOf  250 : Atcelšanas veids. Klasifikatora tips -
"registration_removal_reason".
      - removedOn         date : Ieraksta izslēgšanas datums
      - resignedOn        date : Atbrīvošanas datums
      - rightsOfRepresentation * ur_le_rightsOfRepresentation:
        - Pārstāvības tiesības.
      - termUntil         date : Pilnvaru termiņa gala datums (tikai PPI
subjektiem)
- politicalOrganizationDetails :
  table:
    comments              : Politiskās organizācijas. Satur Politisko
organizāciju teritoriālās struktūrvienības.
    fields:
      - fieldOffices * :
        table:
          comments        : Politisko organizāciju teritoriālās
struktūrvienības.
          fields:
            - address      250 : Struktūrvienības pilna adrese
            - country      3 : Struktūrvienības adreses valsts
            - id           int : Ieraksta identifikators
            - isAnnulled   boolean : Vai ieraksts anulēts
            - lastModifiedAt dateTime : Pēdējo izmaiņu datums un laiks
            - name         250 : Struktūrvienības nosaukums
            - registeredOn  date : Ieraksta izdarīšanas datums
            - removedOn    date : Ieraksta izslēgšanas datums
- previousAddresses * :
  table:
    comments              : Iepriekšējās adreses.
    fields:
      - id               int : Ieraksta identifikators.
      - isAnnulled       boolean : Vai ieraksts anulēts
      - lastModifiedAt   dateTime : Pēdējo izmaiņu datums un laiks
      - registeredOn      date : Ieraksta par adresi spēkā stāšanās datums
      - removedOn        date : Datums, ar kuru adrese vairs nav spēkā
      - address          ur_le_address :
        comments          : Adrese
- previousNames * :
  table:
    comments              : Iepriekšējie nosaukumi.
    fields:
      - id               int : Ieraksta identifikators.
      - name             250 : Iepriekšējais nosaukums (pilnais)
      - isAnnulled       boolean : Vai ieraksts anulēts

```

```

- lastModifiedAt          dateTime : Pēdējo izmaiņu datums un laiks
- registeredOn            date : Datums, kad nosaukums reģistrēts
- removedOn              date : Datums, kad nosaukums vairs nav aktuāls
- previousLegalForms * :
  table:
  comments                : Iepriekšējās komercdarbību formas.
  fields:
  - isAnnulled            boolean : Vai ieraksts anulēts
  - lastModifiedAt        dateTime : Pēdējo izmaiņu datums un laiks
  - register              100 : Iepriekšējais reģistrs. Klasifikators
  - registeredBecauseOf    250 : Ieraksta izdarišanas iemesls. Klasifikatora tips
- "registration_removal_reason".
  - registeredOn          date : Ieraksta izdarišanas datums
  - removedBecauseOf      250 : Ieraksta izslēgšanas iemesls. Klasifikatora tips
- "registration_removal_reason".
  - removedOn            date : Ieraksta izslēgšanas datums
  - type                 250 : Iepriekšējā tiesiskā forma. Klasifikatora tips
- "legal_form".
- procurations * :
  table:
  comments                : Prokūras. Satur datus par Tiesībām rīkoties ar
nekustamo īpašumu, Pārstāvības tiesībām un Prokūristiem.
  fields:
  - dateFrom              date : Datums No
  - id                    int : Ieraksta identifikators.
  - isAnnulled            boolean : Vai anulēts
  - lastModifiedAt        dateTime : Pēdējo izmaiņu datums un laiks
  - prokurists * :
    table:
    comments              : Prokūristi.
    fields:
    - dateFrom            date : Datums No
    - id                  int : Ieraksta identifikators
    - isAnnulled          boolean : Vai ieraksts anulēts
    - lastModifiedAt      dateTime : Pēdējo izmaiņu datums un laiks
    - naturalPerson ur_le_naturalPerson :
      - Satur atbilstošos laukus no struktūras "Fiziska persona"
      - registeredOn      date : Ieraksta izdarišanas datums
      - removedOn         date : Ieraksta izslēgšanas datums
  - registeredOn          date : Ieraksta izdarišanas datums
  - removedOn             date : Ieraksta izslēgšanas datums
  - rightsOnImmovableProperty :
    table:
    comments              : Tiesības rīkoties ar nekustamo īpašumu.
    fields:
    - mayAlienate          boolean : Pazīme par tiesībām atsavināt nekustamo īpašumu
    - mayPledge            boolean : Pazīme par tiesībām ieķīlāt nekustamo īpašumu
    - mayEncumber          boolean : Pazīme par tiesībām apgrūtināt ar lietu tiesībām
nekustamo īpašumu
  - rightsOfRepresentation * ur_le_rightsOfRepresentation :
    - Pārstāvības tiesības.
- prohibitionsOfActivities * :
  table:
  comments                : Darbības liegumi.
  fields:
  - id                    int : Ieraksta identifikators.
  - endedOn              date : Darbības lieguma atcelšanas datums
  - groundsForEndingProhibition : Darbības atjaunošanas pamatojums
  - groundsForProhibition : Lieguma piemērošanas pamatojums
  - isAnnulled            boolean : Vai ieraksts anulēts
  - lastModifiedAt        dateTime : Pēdējo izmaiņu datums un laiks
  - registeredOn          date : Ieraksta izdarišanas datums
  - removedOn            date : Ieraksta izslēgšanas datums
  - resultType            100 : Darbības lieguma rezultāts. Klasifikatora tips -
"liquidation_suspension_prohibition_result".
  - startedOn            date : Darbības lieguma reģistrēšanas datums
  - type                 100 : Darbības lieguma veids. Klasifikatora tips -
"liquidation_suspension_prohibition". Iespējams šāds veids - PROHIBITION_ON_PUBLIC_ACTIVITY
- publicAuthorityDetails :
  table:
  comments: PPI subjektam specifiskā informācija.

```

```

fields:
- contactDetails :
  table:
    comments: Kontaktinformācija.
    fields:
      - website                250 : Oficiālās tīmekļvietnes adrese
      - email                   250 : Elektroniskā pasta adrese
      - phone                   250 : Tālrunis
- registrationDetails :
  table:
    comments: Reģistrācijas informācija.
    fields:
      - authorityType           10 : Klasifikatora tips - "authority". (PARLIAMENT,
CHANCERY_OF_PRESIDENT, COURT, PROSECUTION_OFFICE, INSTITUTION_OF_DIRECT_ADMINISTRATION,
DERIVED_PUBLIC_PERSON, INSTITUTION_OF_INDIRECT_ADMINISTRATION, OTHER_AUTHORITY)
      - establishedOn           date : Datums, ar kuru iestāde ir izveidota, dibināta
      - isInstitutionOfHigherEducation boolean : Pazīme, vai ir augstākās izglītības iestāde
      - isIndependentTaxpayer   boolean : Pazīme, vai ir patstāvīgs nodokļu maksātājs
      - taxpayerUnitNumber       50 : Nodokļu maksātāja struktūrvienības reģistrācijas
kods. To ievada pašas iestādes
      - higherAuthorityNumber    11 : Augstākās iestādes reģistrācijas numurs
      - higherAuthorityCurrentName 250 : Augstākās iestādes aktuālais nosaukums
      - subordinationType        20 : Padotības forma. Klasifikatora tips -
"subordination" (CONTROL, SUPERVISION)
      - establishingAct          ur_le_legalAct :
        - Iestādes izveidošanas tiesību akta informācija.
      - delegatedEntities * :
        table:
          comments: Ziņas par citos reģistros ierakstītajām privātpersonām, kurām publiskā
persona vai iestāde deleģējusi valsts pārvaldes uzdevumu.
          fields:
            - id                 int : Ieraksta identifikators
            - registrationNumber  11 : Iestādes reģistrācijas numurs
            - name                250 : Iestādes nosaukums
            - registeredOn        date : Ieraksta izdarīšanas datums
            - removedOn           date : Ieraksta izslēgšanas datums
            - lastModifiedAt      dateTime : Pēdējo izmaiņu datums un laiks
      - schoolBranches * :
        table:
          comments: Augstskolas filiāles.
          fields:
            - id                 int : Ieraksta identifikators
            - name                250 : Nosaukums
            - address             ur_le_address :
              comments: Filiāles adrese.
            - generalManager :
              table:
                comments: Filiāles vadītājs
                fields:
                  - latvianIdentityNumber  11 : Personas kods
                  - latvianIdentityNumberChangedOn dateTime : Personas koda maiņas datums
                  - forename                34 : Vārds
                  - surname                  34 : Uzvārds
                  - registeredOn            date : Ieraksta izdarīšanas datums
                  - removedOn               date : Ieraksta izslēgšanas datums
                  - lastModifiedAt          dateTime : Pēdējo izmaiņu datums un laiks
                  - isAnnulled              boolean : Vai ieraksts anulēts
      - register                 100 : Reģistrs. Klasifikatora tips - "register".
      - registeredOn             date : Datums, kad ierakstīts reģistrā.
      - registrationNumber        11 : Reģistrācijas numurs
      - registrationNumberAssignedOn date : Datums, kad piešķirts esošais reģistrācijas
numurs.
      - removedOn                date : Datums, kad izslēgts no reģistra.
      - religiousEntityDetails :
        table:
          comments: Reliģiskās organizācijas.
          fields:
            - affiliation          250 : Konfesija. Klasifikatora tips -
"religious_affiliation".
            - previousRegistrations * :
              table:

```

```

        comments                                : Reliģisko organizāciju iepriekšējās
reģistrācijas.
        fields:
        - registrationNumber                    10 : Valsts reģistrācijas numurs; Uzsk. Reģistrācijas
numurs
        - registeredOn                          date : Reģistrācijas datums
- reregistrations * :
    table:
    comments                                    : Pārreģistrācijas.
    fields:
    - date                                      date : Pārreģistrācijas datums
    - register                                  250 : Iepriekšējā reģistra publiskais kods.
- reorganizations * :
    table:
    comments                                    : Reorganizācijas. Satur Reorganizācijas subjektus
un Reorganizācijas objektus.
    fields:
    - id                                        int : Ieraksta identifikators
    - date                                      date : (tikai PPI) Datums, ar kuru reorganizācija
stājas spēkā. Var būt datums nākotnē, kad reorganizācija tiek plānota.
    - entitiesInvolved * :
        table:
        comments                                : (tikai PPI) Reorganizācijā iesaistītie subjekti.
        fields:
        - registrationNumber                    11 : Subjekta reģistrācijas numurs
        - currentName                          250 : Subjekta nosaukums
        - links                                ur_le_links :
            - Satur atbilstošos laukus (self) no struktūras "Saites"
    - entitiesObjects * :
        table:
        comments                                : Sabiedrība, kas noslēdz reorganizācijas procesu.
        fields:
        - address                              120 : JP juridiskā adrese
        - country                              2 : JP reģistrācijas valsts
        - currentName                          250 : JP pilnais pašreizējais nosaukums
        - externalObjectNumber                 10 : Ārējā objekta reģistrācijas numurs.
        - foreignRegistrationNumber             50 : Ja country ='LV' - ārējā objekta VID piešķirtais
nodokļu maksātāja kods. Ja country !='LV' - ārējā objekta oriģinālais reģistrācijas numurs, ja
tāda nav, tad VID piešķirtais nodokļu maksātāja kods.
        - name                                250 : JP nosaukums
        - registrationNumber                   11 : Reģistrācijas numurs
        - links                                ur_le_links :
            - Noteiktas vērtības (self) no saišu saraksta.
    - entitiesSubjects * :
        table:
        comments                                : Sabiedrība, kas uzsāk reorganizācijas procesu.
        fields:
        - address                              120 : JP juridiskā adrese
        - country                              2 : JP reģistrācijas valsts
        - currentName                          250 : JP pilnais pašreizējais nosaukums
        - externalObjectNumber                 10 : Ārējā objekta reģistrācijas numurs.
        - foreignRegistrationNumber             50 : Ja country ='LV' - ārējā objekta VID piešķirtais
nodokļu maksātāja kods. Ja country !='LV' - ārējā objekta oriģinālais reģistrācijas numurs, ja
tāda nav, tad VID piešķirtais nodokļu maksātāja kods.
        - name                                250 : JP nosaukums
        - registrationNumber                   11 : Reģistrācijas numurs
        - links                                ur_le_links :
            - Noteiktas vērtības (self) no saišu saraksta.
    - isAnnulled                             boolean : Vai ieraksts anulēts (vēl netiek lietots).
    - lastModifiedAt                          dateTime : Pēdējo izmaiņu datums un laiks (pašlaik lietots
tikai PPI subjektiem).
    - legalAct                                ur_le_legalAct :
        - (tikai PPI) Reorganizācijas tiesību akts.
    - registeredOn                            date : Reorganizācijas ieraksta izdarišanas datums
    - removedOn                              date : Reorganizācijas ieraksta dzēšanas/izslēgšanas
datums (pašlaik lietots tikai PPI subjektiem)
    - result                                  : (tikai PPI) Reorganizācijas sekas
    - type                                    100 : Reorganizācijas veids. Klasifikatora tips -
"reorganization".
- representatives * :
    table:

```



```

        comments : Pārstāvji (ārvalsts komersantam vai
organizācijai). Satur datus par Fiziskām personām.
        fields:
        - appointedOn date : Ieraksta izdarišanas par amatpersonas iecelšanu
datums
        - id int : Ieraksta identifikators
        - isAnnulled boolean : Vai anulēts
        - lastModifiedAt dateTime : Pēdējo izmaiņu datums un laiks
        - naturalPerson ur_le_naturalPerson :
            - Satur atbilstošos laukus no struktūras "Fiziska persona"
        - note : Piezīmes
        - resignedOn date : Atbrīvošanas datums
        - registeredBecauseOf 250 : Iecelšanas veids. Klasifikatora tips -
"registration_removal_reason".
        - registeredOn date : Ieraksta izdarišanas datums
        - removedBecauseOf 250 : Atbrīvošanas veids. Klasifikatora tips -
"registration_removal_reason".
        - removedOn date : Ieraksta izslēgšanas datums
- securingMeasures *:
    table:
        comments : Nodrošinājuma līdzekļi. Satur Nodrošinājuma
līdzekļu piemērotāju un Liegumus uz kapitāla daļām.
        fields:
        - caseNumber 15 : Dokumenta, uz kā pamata tiek veikta darbība ar
nodrošinājuma līdzekli numurs
        - dateFrom date : Ieraksta izdarišanas datums par nodrošinājuma
uzlikšanu
        - dateTo date : Nodrošinājuma atcelšanas datums
        - id int : Ieraksta identifikators
        - imposedBy ur_le_legalEntity :
            - Nodrošinājuma līdzekļa piemērotājs - Juridiska persona.
        - isAnnulled boolean : Vai ieraksts anulēts
        - lastModifiedAt dateTime : Ieraksta pēdējo izmaiņu datums un laiks
        - noteOnRegistration : Dokumenta, uz kā pamata tiek veikta darbība ar
nodrošinājuma līdzekli (piemērots), saturs
        - noteOnRemoval : Dokumenta, uz kā pamata tiek veikta darbība ar
nodrošinājuma līdzekli (atcelts), saturs
        - prohibitionOnMemberShareCapital :
            table:
                comments : Liegumi uz kapitāldaļām.
                fields:
                - naturalPerson ur_le_naturalPerson :
                    - Satur atbilstošos laukus no struktūras "Fiziska persona"
                - legalEntity ur_le_legalEntity :
                    - Satur atbilstošos laukus no struktūras "Juridiska persona"
        - registeredOn date : Ieraksta izdarišanas datums
        - removedOn date : Ieraksta izslēgšanas datums
        - type 250 : Piemērotā (atceltā) nodrošinājuma līdzekļa
veids. Klasifikatora tips - "securing_measure"
        - typeDetails 250 : Piemērotā (atceltā) nodrošinājuma līdzekļa
paveids
        - sepaCreditorId 18 : SEPA identifikators
- specialStatuses *:
    table:
        comments: Objekta speciālie statusi.
        fields:
        - id intString : Ieraksta identifikators
        - dateFrom date : Datums No
        - dateTo date : Datums līdz
        - isAnnulled boolean : Vai ieraksts anulēts
        - lastModifiedAt dateTime : Pēdējo izmaiņu datums un laiks
        - registeredOn date : Ieraksta spēkā stāšanās datums
        - removedOn date : Ieraksta izslēgšanas datums
        - type : Speciālā statusa veids. Klasifikatora tips -
"various_status".
        - status 25 : Statuss
        - statusDetails 25 : Statusa detalizācija. Klasifikatora tips -
"status_details".
- suspensionsOfActivities *:
    table:
        comments : Darbības apturēšanas.

```

```

fields:
- id int : Ieraksta identifikators
- isAnnulled boolean : Vai ieraksts anulēts
- lastModifiedAt dateTime : Ieraksta pēdējo izmaiņu datums un laiks
- endedOn date : Darbības apturēšanas izbeigšanas vai darbības
atjaunošanas registrēšanas datums
- groundsForRenewal : Darbības atjaunošanas pamatojums
- groundsForSuspension : Darbības apturēšanas pamatojums
- registeredOn date : Ieraksta izdarīšanas datums
- removedOn date : Ieraksta izslēgšanas datums
- resultType 250 : Saimnieciskās darbības apturēšanas rezultāts.
Klasifikatora tips - "liquidation_suspension_prohibition_result".
- startedOn date : darbības apturēšanas datums
- type 20 : Likvidācijas procesa veids. Klasifikatora tips -
"liquidation_suspension_prohibition". Iespējami šādi veidi -
ACTIVITY_SUSPENDED,ACTIVITY_SUSPENDED_BY_COURT,ECONOMIC_ACTIVITY_SUSPENDED
- type : Tiesiskā forma. Klasifikatora tips -
"legal_form".
- variousNotices * :
table:
comments : Dažādas citi paziņojumi.
fields:
- date date : Dokumenta datums
- id int : Ieraksta identifikators
- isAnnulled boolean : Vai ieraksts anulēts
- lastModifiedAt dateTime : Ieraksta pēdējo izmaiņu datums un laiks
- registeredOn date : Ieraksta izdarīšanas datums
- removedOn date : Ieraksta izslēgšanas datums
- type 250 : Papildinformācijas veids. Klasifikatora tips -
various_notice".

```