

Programmatūras integrācijas instrukcija

Valsts digitālās attīstības aģentūra

DATU IZPLATĪŠANAS UN PĀRVALDĪBAS PLATFORMA (DAGR)

VERSIJA 1.12

2024

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	Pievienota sadaļa "Datu izmaiņu monitorings"
	09.07.2024	Aktualizēta sadaļa "Audita apskate portālā"

Satura rādītājs

1	Ievads	5
1.1	Dokumenta nolūks	5
1.2	Projekta mērķi.....	5
1.3	Termini un saīsinājumi	5
2	DAGR vispārīgā arhitektūra.....	6
3	Integrācijas soļu plāns.....	7
3.1	Datu patērētāja integrācijas soļi	7
3.2	Datu devēja integrācijas soļi	8
4	Datu patērētāja integrācija	19
4.1	Iepazīšanās ar pieejamajiem datiem un servisiem	19
4.2	Pieteikšanās par datu patērētāju.....	20
4.2.1	Datu patērētāja pieteikums	20
4.2.2	Datu patērētāja savienotāja reģistrēšana	22
4.3	Datu izgūšana.....	25
4.3.1	Autentifikācija (drošības talona pieprasīšana).....	25
4.3.2	Atļauju noskaidrošana.....	27
4.3.3	Datu objekta izgūšana pēc objekta identifikatora	29
4.3.4	Datu izmaiņu izgūšana un pilnas datu kopas pārļasišana	30
4.3.5	Datu izmaiņu monitorings.....	33
4.3.6	Lielpjoma datu (failu) izgūšana	37
4.3.7	Pielāgoti servisi	37
4.3.8	Pēdējo izmaiņu datums.....	39
4.4	Datu pieprasījumu audits.....	39
4.4.1	Audita apskate portālā.....	39
4.4.2	Audita izgūšana caur REST API	39
4.5	Datu pieprasījumu audits.....	40
4.5.1	Audita apskate portālā.....	40
4.5.2	Audita izgūšana caur REST API	41
5	Datu devēja integrācija	42
5.1	Pieteikšanās par datu devēju	42
5.2	Izplatāmo datu kopu definēšana	44
5.2.1	Datu kopas pievienošana priekšlikumu videi (sandbox)	45
5.2.2	Datu kopas pārceļšana uz pamata sadaļu (dataset)	47

5.2.3	Specializētās datu kopas	49
5.3	Datu nodošanas mehānisma izvēle.....	52
5.4	Datu devēja savienojuma reģistrēšana	54
5.5	Datu nodošana uz DAGR	56
5.5.1	DAGR REST API izmantošana.....	56
5.5.2	Datu devēja savienotāja uzstādīšana	63
5.5.3	Debeziem savienotājs	72
5.6	Personas datu izmantošanas žurnālpieraksti.....	73
5.7	Datu anonimizācija.....	74
5.7.1	Datu anonimizācija izmantojot datu devēja savienotāju.....	74
5.7.2	Datu anonimizācija caur DAGR pašapkalpošanās portālu	77
6	Pielikums – papildus paraugi.....	79
6.1	Datu kopu definīcijas.....	79
6.2	Datu struktūras kartējums (connector.yaml).....	92

1 Ievads

Šis dokuments sagatavots saskaņā ar darba uzdevumu Nr.2 "Izmaiņu monitoringa un auditācijas pierakstu uzlabojumi" pie vispārīgās vienošanās ID Nr. VRAA EIS 2021/14/AK/CI-118-3-PPK_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 datu patērētājiem
Datu devēja savienotājs	DAGR reģistrēta tehnoloģiska komponente ar unikālu identifikatoru (<i>client id</i>), kas datu devēja vārdā veic datu ielādi DAGR kodolā
Datu patērētājs	tiesību subjekts, kas tiek identificēts ar reģistrācijas numuru un saņem datus no datu devēja caur DAGR
Datu patērētāja savienotājs	DAGR reģistrēta tehnoloģiska komponente ar savu identifikatoru (<i>client id</i>), kas datu patērētāja 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, datu patērētāja vai DAGR vārdā, atbilstoši lietotājam piešķirtajām sistēmas tiesībām
VIRISIS	Valsts informācijas resursu, sistēmu un sadarbības informācijas sistēma
VPM	Vienotais pieteikšanās modulis

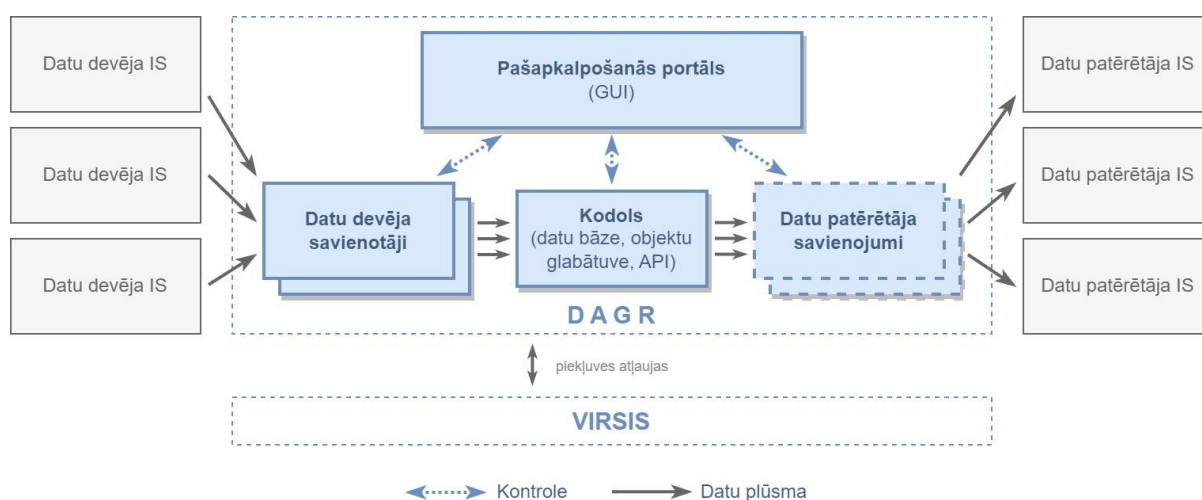
2 DAGR vispārīgā arhitektūra

DAGR nodrošina datu izplatīšanu datu patērētājiem no datu devējiem. Lai nodrošinātu datu patērētāju 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 varianta un datu devēja IS iespējām, datu devēja savienotāji tiek veidoti tā, lai varētu nodrošināt caur DAGR izplatāmo datu aktualitāti, kas ir mērāma sekundēs.

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

Pašapkalpošanās portāls ir DAGR lietotāja saskarne, kas ļauj lietotājiem pieteikties par datu patērētājiem 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 VIRSIS. DAGR ļauj datu patērētājiem iegūt datus atbilstoši VIRSIS reģistrētajām datu piekļuves atļaujām. Datu piekļuves atļaujas ir ar granularitāti līdz konkrētiem datu laukiem.

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

3 Integrācijas soļu plāns

3.1 Datu patērētāja integrācijas soļi

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

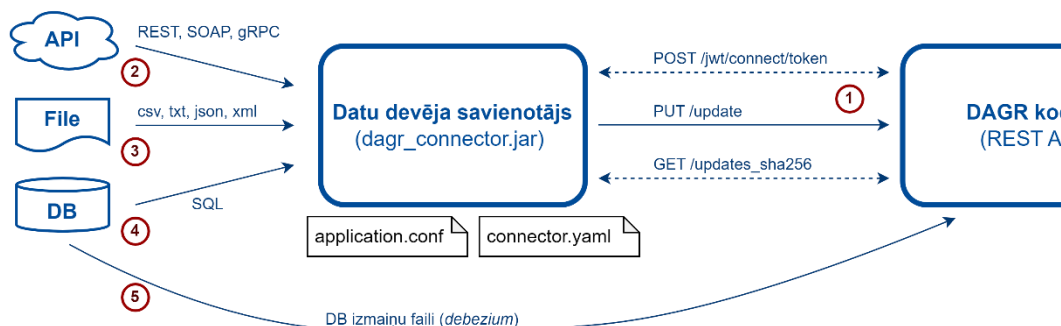
Nr.	Darbība	Apraksts
0.	Iepazīties ar pieejamajām datu kopām	Pirms uzsākt darbus ir vēlams iepazīties ar datu kopām, kuras datu devēji izplata caur DAGR, lai noskaidrotu ka nepieciešamās datu kopas ir pieejamas. Skatīt sadaļu "4.1 Iepazīšanās ar pieejamajiem datiem un servisiem".
1.	Iegūt datu piekļuves 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 patērētāju	Lai uzsāktu darbu ar DAGR iestādei jāreģistrējas DAGR kā datu patērētājam. To var izdarīt aizpildot datu patērētāja pieteikumu DAGR pašapkalpošanās portālā vai iesniedzot to rakstiskā formā DAGR pārzinim. Skatīt sadaļu "4.2.1 Datu patērētāja pieteikums".
3.	Reģistrēt datu patērētāja savienojumu	Pēc pieteikuma apstiprināšanas pieteikumā norādītajai personai ir jāpieslēdzas DAGR pašapkalpošanās portālam un jāreģistrē datu patērētāja savienojums. Skatīt sadaļu "4.2.2 Datu patērētāja savienotāja reģistrēšana".
4.	Izgūt datus	Jāizmanto DAGR API datu iegūšanai. Vispirms, norādot reģistrētā datu patērētāja 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:

Nr.	Darbība	Apraksts
1.	Pieteikšanās par datu devēju	Lai uzsāktu darbu ar DAGR vajag aizpildīt datu devēja pieteikumu DAGR pašapkalpošanās portālā vai iesniegt to rakstiskā formā DAGR pārzinim. Skatīt sadaļu "5.1 Pieteikšanās par datu devēju".
2.	Datukopu definēšana	Datu devējs definē caur DAGR izplatāmās datu kopas. Datu devējs šīs datu kopas var reģistrēt DAGR pašapkalpošanās portālā pats, aizpildot datu kopas reģistrācijas formu, kuru tālāk DAGR pārvaldnieks apstiprina. Skatīt sadaļu "5.2 Izplatāmo datu kopu definēšana". Alternatīvi datu devējs datu kopu reģistrāciju var lūgt veikt DAGR pārvaldniekam.
3.	Datu devēja savienojuma reģistrēšana	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". 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.
4.	Datu devēja savienotāja uzstādīšana	Datu devējs lejupeļādē datu devēja savienotāju un tā uzstādīšanas instrukcijas, veic datu devēja savienotāja konfigurēšanu un iedarbina to datu devēja infrastruktūrā. Skatīt sadaļu "0

3.3 Datu nodošanas mehānisma izvēle



Ilustrācija 2 - datu nodošana uz DAGR

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

1. **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 "5.5.1 DAGR REST API izmantošana".
2. **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.
3. **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.
4. **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.
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 "Piekļuve DAGR" apakš sadaļā "Savienotāji". Šajā sadaļā ir pieejams saraksts ar visiem savienojumiem:



A A A
EN


Akciju sabiedrība "Citadele bank"
Juridiska persona 40103303559

🏠 **Piekļuve DAGR**
Darbību monitorings

🚨 **Some alert text**

Sākums > Datu patērētāja savienotāji

Piekluve DAGR ^

Pieteikumi

Savienotāji

Darbību monitorings v

+ Jaun

ABC test client URPK f61d5a2d-f084-4a90-9c22-69f15c266f8a F61D5A2D-F084-4A90-9C22-69F15C266F8A	✓
ABC test client URPK try change by DPUR 5fe5c410-c037-498d-a446-1d5... 5FE5C410-C037-498D-A446-1D586750B785	✓
ABC test client URPK 0a7bef25-facd-45c9-bfee-5fe35869f991 0A7BEF25-FACD-45C9-BFEE-5FE35869F991	✓
ABC test client URPK 03aebdf4-07bd-4430-ad6a-3cc83546a721 03AEBDF4-07BD-4430-AD6A-3CC83546A721	✓
ABC test client URPK changed a9e9ead4-240e-447b-aeb2-a22458f8102a A9E9EADA-240E-447B-AEB2-A22458F8102A	✓
ABC secrets test client URPK 926803bb-0686-42fe-86a1-85e2c65fd3df 926803BB-0686-42FE-86A1-85E2C65FD3DF	✓
ABC secrets test client URPK 0253fddc-e6d1-4107-bde1-651c8f70a40b 0253FDDC-E6D1-4107-BDE1-651C8F70A40B	✓

Lai reģistrētu jaunu savienojumu jāaizpilda forma:

Datu patērētāja savienotāja reģistrācija

NOSAUKUMS

IDENTIFIKATORS

Aktīva

DROŠĪBAS TALONA DŽĪVES LAIKS (SEKUNDĒS)

APRAKSTS

Saglabāt

Atcelt

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. Drošības sertifikāta pievienošana datu devēja savienotāja reģistrācijas scenārijā neatšķiras no datu patērētāja savienotāja reģistrācijas scenārija – detalizētāku aprakstu par sertifikātiem skatīt sadaļā “4.2.2 Datu patērētāja savienotāja 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 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://test.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 (testa vidē):

<https://test.dagr.gov.lv/jwt/connect/token>

Drošības talona izsaukuma piemērs:

```
curl --location 'https://test.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.eyJzdWIiOiJlJEQUdSX0FLQ19URVNUXzIiLCJqdGkiOiJkOTQwZTliNC0yYjgwLTQ4NzUtYmEyMS1mYmVmYmVhYzUzZjIiLCJpc3MiOiJlJEQUdSX0FLQ19URVNUXzIiLCJhdWQiOiJlcm46VEVTVVC1MTkI6REFHU5TVFMvY29ubmVjdC90b2t1biIsImV4cCI6IjE3MDEyOTQ5NjQ1LCJmYmYiOiIxNzA5MjkkxMzY0In0.Xq3bZNI62hS9OFYx-gNBEbh2SwQt72to_gL3Xwa9yh_uyUqd5_tDJuXkz-
```

```
F03Jn0CdcRKcdeitrKTtSCxIIP3WqQpbotsPHZvuf0ek26koyhtGhK2lw6RNF8mPkIktb4L6F
0GuTjwc9ZmQ10lYo86u6_ZZWe2sS8aISJAUBmClW5cIt_qc3xY8no_UUJwDlyRGfxntGw3T_I
yvIv0bPOLlOvyfmswpcI7I5S2qR5EIJlKggM4SninbGfB1j_u7HGSzi014kIXmOikdhDviyP0
je9kYBcwZgwYeja19dBxo1NkDd6A3BG3jVPlvxJG2QQ1HdWNkHbXSonyJajDhGnA' \
--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 jāgatavo atbilstoši RFC 7515 JSON Web Signature(JWS) standartam. Respektīvi, *client_assertion* sastāv no 'Header' un 'Payload', kas parakstīti ar datu patērētāja 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:TEST-LNB:DAGR.STS/connect/token,
  "exp": {derīguma termiņš, max nbf + 1 stunda},
  "nbf": {parakstīšanas datums un laiks}
}
```

Piemērā minēta testa vides “aud” jeb audiences vērtība. Produkcijas videi tā ir: “urn:LNB:DAGR.STS/resources”.

Drošības talona izsaukuma atbildes piemērs:

```
{
  "access_token":
  "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlc2ludG93LW50X21kIjoiVlZEX0REU18xIiwic2NvcGUiOiJvZHVjZSIiXX0.u8320MjhZzojv2wIpfie9o19gkN0AVmi_YA
CeHxmYsk4JC-_XC3ad4tj9m0j1fnGM4u80XwX2WEhbc_i3YtUMBUCsojdJ70u9-
tuggggZivKmlXbWv1D6Ui1FrKRPTdnhp5FCbTw6KgyTh_jqtXqGImT09ELPH0ts_-
8DXlhAnnBfIbewznUiUuWeNWpsejd2zVD0hsancdPwW7w9dzNjhve6Q1Trjn3Gk5Hn6b2Yj8y
fVD3bNn6r7uOWo_o79mJo4LdWDNy3MX8he54OGd93EoMt0UoiW9FETDfttiu3iioAFHo_sizk
jHNGUY48o9xglV7-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

3.5.1.2 Strukturētu datu iesūtīšana

Strukturēti dati uz DAGR nododami izmantojot /update servisu:

- Apstiprinātām datu kopām servisa adrese veidojas šādi: **PUT {dagr_adrese}/update/{datu_kopas_nosaukums}**
- Izmēģinājuma (smilškastēs) datu kopām servisa adrese veidojas šādi: **PUT {dagr_adrese}/sandbox/{datu_devēja_reģ_nr}/update/{datu_kopas_nosaukums}**

Nododamie dati iekļaujami servisa saturā (body) JSON formātā. Servisam var nodot uzreiz vairākus ierakstus, kur katrs ieraksts norādāms šādā formā:

```
{
  "key": "string",
  "timestamp": "2024-03-06T10:04:55.297Z",
  "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” aizvieto 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://test.dagr.gov.lv/sandbox/40003562863/update/tst_company_service_report' \
--header 'Content-Type: application/json' \
```


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 aprēķinātājām 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:

- Jāsigatavo 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' \
```

	<pre>'https://test.dagr.gov.lv/file/vsaa/test_file_name.txt' \ -H 'accept: text/plain' \ -H 'Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcm46VEVTVG1MTkI6REFHUWU5TVFMeiLCJzdW IiOiI5MDAwMTY2OTQ5NiIsImF1ZCI6WyJwcm9kdWNlciIsInVyb2RlLnVudS1wcm9udS1pbnNlIiwiaWF0Ijoi XzJ5ZmVudS1wcm9udS1pbnNlIiwiaWF0IjoiXzJ5ZmVudS1wcm9udS1pbnNlIiwiaWF0IjoiXzJ5ZmVudS1wcm9udS1pbnNl YWxlb2R1Y2VyIi19.gZgNI6451nrLAMYp4UHRWLBtGlvPjIBxPpP1j2xiU5RyUf2lrad3CWdmyFyBIrlhz SjmQ5ITvrtJrvlMm0ID4j3BPoDiUsrOtr-60FEbR3rwYlruQyu_CiHDBLbyfyWr8KKSbyov834- 37xlwrZjablC-t_tncs7Uyd9-pvQp_RjkVmEYgwfPLiq- edOyczQQ4WwcYwVoyKgh05yA5TX3LQ7n4oOGvJvMa9KozenU2a49y4bLaxwQGKtxV_kox75ABMXroaGT511 1FpTRz1AnzCBh6n4D3s3D0novL2IppqYgBxA5jhybkwJpdlynD_n4SVQDJNlLp_UiRYq-gnbCGw' \ -H 'Content-Type: application/octet-stream' \ --data-binary '@test_file.txt'</pre> <p>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.</p> <p>Datu devēja savienotāja uzstādīšana”.</p>
<p>4 Dat b u . de vēj a sav ien otā ja uzs tād īša na un ko nfi gur ēša na DA GR infr ast ruk tūr ā</p>	<p>Alternatīvi datu devēja savienotāja uzstādīšanai datu devēja infrastruktūrā datu devējs DAGR pārvaldniekam var lūgt datu devēja savienotāju uzstādīt DAGR centrālajā infrastruktūrā.</p>

4 Datu patērētāja 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://test.dagr.gov.lv/api>

Saraksts ar datu kopām un servisiem caur kuriem tiek izplatīti datu devēju dati pieejams servisā: https://test.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://test.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 patērētājiem 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 patērētāju

Kā pirmais solis, datu devējs vai datu saņēmējs, iesniedz DAGR pārzinim attiecīgu pieteikumu, izmantojot oficiālo elektronisko adresi.

Tālāk pieteikties par datu patērētāju un reģistrēt datu patērētāja savienojumu var DAGR pašapkalpošanās portālā, kurš pieejams adresē: <https://test.dagr.gov.lv/>

Lai izmantotu pašapkalpošanās portālā pieejamo funkcionalitāti, lietotājam ir jāautenticējas ar VPM, izmantojot kvalificēti paaugstinātas drošības elektroniskās identifikācijas līdzekļi: eID, eParaksts un eParaksts mobile.

4.2.1 Datu patērētāja pieteikums

Par datu patērētāju ar VPM autentificēti lietotāji var pieteikties portāla sadaļā "Pieteikumi":

The screenshot shows the DAGR portal interface. At the top, there is a header with the Latvian coat of arms, language settings (A A A and EN), and a user profile for 'Kaspars Eglītis, Fiziska persona'. Below the header, there are navigation tabs for 'Pieklūve DAGR' and 'Darbību monitorings'. A notification bar at the top left says 'Some alert text'. The main content area is titled 'Pieteikumu pārvaldība' and includes a breadcrumb 'Sākums > Pieteikumu pārvaldība'. On the left, there is a sidebar menu with options: 'Pieklūve DAGR', 'Pieteikumi' (selected), 'Savienotāji', and 'Darbību monitorings'. The main area displays 'Jums ir šādi pieteikumi:' followed by a table of requests:

Request Description	Status
Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	✓
ABC test company 40003627089 40003627089	⌚
Pilsonības un migrācijas lietu pārvalde 40001006969	✓
Pilsonības un migrācijas lietu pārvalde 40000006969	⌚

Below the table, there is a section for 'DAGR datu savienotāja licences pieprasījums' with two buttons: 'Datu devēja pieteikums' and 'Datu patērētāja pieteikums'. At the bottom left, there is contact information for the Ministry of Environmental Protection and Regional Development.

Pieteikšanās procesam ir divas plūsmas:

- Uzņēmumi, kas ir reģistrēti Latvijas Uzņēmumu reģistrā, var pieteikties par datu patērētāju, ja lietotājs portālam izmantojot VPM autentifikāciju pieslēdzies kā juridiskā persona. Šajā gadījumā pieteikumā iekļaujамie fiziskās personas un uzņēmuma dati tiks nolasīti no autentifikācijas drošības talona;
- Valsts pārvaldības iestādes var pieteikties par datu patērētāju, ja lietotājs portālam pieslēdzies kā fiziskā persona. Šajā gadījumā pieteikumā iekļaujамie fiziskās personas dati tiek nolasīti no autentifikācijas drošības talona, savukārt iestādes dati ir manuāli norādāmi pieteikumā.

Nospiežot uz “Datu patērētāja pieteikums”, atkarībā no autentificētas personas veida, lietotājs tiek pārvirzīts uz attiecīga pieteikuma aizpildīšanu.

Ja autentificēta persona ir uzņēmums kā juridiskā persona, lietotājam tiek piedāvāts aizpildīt šādu pieteikuma formu:

Licences pieprasījums datu patērētājam

REĢISTRĀCIJAS NUMURS	40000000010
NOSAUKUMS	Sabiedrība ar ierobežotu atbildību "Testing Solutions"
TEHNISKĀ KONTAKTPERSONA	Selenium Testing
PERSONAS KODS	040193-11692
TĀLRUNIS	22853809
E-PASTA ADRESE	antra.liepa@pmlp.gov.lv
<input checked="" type="checkbox"/> Deleģēt datu savienotāja konfigurēšanu	
DELEĢĒTA KONTAKTPERSONA	Jānis Bērziņš
PERSONAS KODS	123456-12345
TĀLRUNIS	22853810
E-PASTA ADRESE	janis.berzins@pmlp.gov.lv

Ja autentificēta persona ir Valsts pārvaldības iestādes pārstāvis kā fiziskā persona, lietotājam tiek piedāvāts aizpildīt šādu pieteikuma formu:

Licences pieprasījums datu patērētājam

TEHNISKĀ KONTAKTPERSONA	Kaspars Eglītis
PERSONAS KODS	070884-11754
TĀLRUNIS	22853809
E-PASTA ADRESE	antra.liepa@pmlp.gov.lv
REĢISTRĀCIJAS NUMURS	90000000501 <input type="button" value="Q"/>
NOSAUKUMS	

Deleģēt datu savienotāja konfigurēšanu

DELEĢĒTA KONTAKTPERSONA	Jānis Bērziņš
PERSONAS KODS	123456-12345
TĀLRUNIS	22853810
E-PASTA ADRESE	janis.berzins@pmlp.gov.lv

Šajā formā iestāde norādāma ievadot tās reģistrācijas numuru un nospiežot uz vizuālā elementa ar palielināmā stikla ikonu. Ja iestāde tiks veiksmīgi atrasta, tad parādīsies iestādes nosaukums.

Abu veidu pieteikumos ir iespējams pievienot papildus tehniskās kontaktpersonas, kas varēs darboties datu patērētāja vārdā.


Pēc pieteikuma iesniegšanas tas tiek nosūtīts DAGR pārvaldniekam un līdz apstiprināšanas vai noraidīšanas brīdim nonāk statusā "Gaidīšanā".

Pieteikuma noraidīšanas vai iepriekš apstiprināta pieteikuma anulēšanas gadījumā, iespējams iesniegt atkārtotu pieteikumu. Piesakoties atkārtoti lietotājs tiek pārvirzīts uz licences pieprasījuma lapu, kur ir iespējams rediģēt iepriekš pieteiktā pieprasījuma lauku vērtības.

4.2.2 Datu patērētāja savienotāja reģistrēšana

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

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

 **Piekluve DAGR**

Sākums > Datu patērētāja savienotāji


Piekluve DAGR ^





Pieteikumi

Savienotāji

Pielāgotie servisi

IZVĒLIETIES IESTĀDI

 **Jauns savienotājs**

 Zaka testa datu patērētāja savienotājs ZAKA_TEST_1		Dzēst
 ZAKA savienotājs ZAKA_TEST_2		Dzēst

Darbību monitorings v

Datu izplatīšana v

Lai reģistrētu jaunu savienotāju jāaizpilda forma:

Datu savienotāja reģistrācija

NOSAUKUMS

IDENTIFIKATORS

DATU SAVIENOTĀJA TIPS

Datu paterētājs v

Aktīva

DROŠĪBAS TALONA DZĪVES LAIKS (SECONDS)

0

APRAKSTS

Saglabāt

Atcelt

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:

Datu patērētāja savienotāja informācija

Visparīgi Sertifikāti

+ Pievienot jaunu sertifikātu

Nav neviena sertifikāta

Saglabāt Atcelt

Sertifikāta pievienošana

PUBLISKĀ ATSLĒGA (*.CER)

Choose File No file chosen

Saglabāt Atcelt

Informācija par sertifikātu

DAGR no pievienotā sertifikāta saglabā publisko atslēgu, kura pēc tam tiek izmantota lai pārbaudītu vai drošības talona pieprasījums parakstīts ar saistīto privāto atslēgu. DAGR pievienojamais x509 sertifikāts var būt arī pašparakstīts. Pastāv vairāki varianti kā iegūt pašparakstītu X.509 sertifikātu. MS Windows vidē to var izdarīt caur PowerShell izmantojot openssl komandrīku:

```
openssl genrsa -out c:\temp\nosaukums.key 2048
```

Šī komanda ģenerē privāto atslēgu, kas pēc tam būs nepieciešama, lai parakstītu drošības talona pieprasījumu (skatīt sadaļu "4.3.1 Autentifikācija (drošības talona pieprasīšana)"). Privātā atslēga paliek pie datu patērētāja un tās aizsargāšana ir datu patērētāja pienākums.

```
openssl req -key c:\temp\nosaukums.key -new -x509 -subj  
"/C=LV/ST=Riga/L=Riga/O=Iestades_nosaukums/OU=IT/CN=nosaukums" -days 365 -out  
c:\temp\nosaukums.crt
```

Šī komanda ģenerē pašparakstītu X.509 sertifikātu, kuru jāpievieno DAGR reģistrētajam datu patērētāja savienotājam.

4.3 Datu izgūšana

4.3.1 Autentifikācija (drošības talona pieprasīšana)

Datu patērētāja savienotāji tiek autentificēti un autorizēti saskaņā ar *OAuth 2.0 OpenId Connect* standarta *Client Credentials Flow* plūsmu.

Lai izgūtu datus no DAGR vispirms ir jāiegūst JWT drošības talons. Drošības talona pieprasīšanas adrese (testa vidē): <https://test.dagr.gov.lv/jwt/connect/token>

Drošības talona izsaukuma piemērs:

```
curl --location 'https://parvaldiba-
test.dagr.gov.lv/Identities.STS/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.eyJzdWIiOiJWQVJBTV
90ZXN0X2NvbW51Y3Rvcl8xIiwianRpIjoimMFiYWlYOTctNDViNC00ZjdkLTg5NWYtOTMxM2JlN2Q0ZTElI
iwiaXNzIjojVksQU1fdGVzdF9jb25uZWN0b3JfMSIsImF1ZCI6InVybjpURVNUUxOQjpeQUdSLlNUUy9j
b25uZWN0L3Rva2VuIiwiaXNzIjoimTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4MTY4
UQ7WHrF9A_pV26WSeljBZCKPx3QSDs41A_c2oxSRVdn_laqSpNydlmGAEPD9o0dgvP7efwq-
w7jldf_4lY8TJyrju_niPaE3krIViLA_Auwv7q66k_TW6jjWud2jGuqgDLYGuRMgIGTMYVv9qVN70ctEBRA
JzLyVFZAK3Z72EOzv_YkKGxb4NnVf4CAbXl6qQSU5dAYz0DPbMvdHGH2AsMNsXeGTuSf7SVQvIl5CShMm1K
wr2D0B6tsruuAGcdj92Z8tupy2InDB-
rti6cpTorTdNPQcClcegGbQDStlSK9k7XsVJf4_Youe3EBCXICGindtxc2AQ' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'client_id=VARAM_test_connector_1' \
--data-urlencode 'scope=consumer'
```

Šajā izsaukumā *client_id* ir jānorāda datu patērētāja savienotāja identifikators 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 patērētāja savienotāja privāto atslēgu.

'Header' veidojams kā JSON objekts, kas satur paraksta parametrus, piemēram:

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

'Payload' ir iekļaujama šāda informācija (testa videi):

```
{
  "sub": {datu patērētāja savienotāja id},
  "jti": {uuid},
  "iss": {datu patērētāja savienotāja id},
  "aud": urn:TEST-LNB:DAGR.STS/connect/token,
  "exp": {derīguma termiņš, max nbf + 1 stunda},
  "nbf": {parakstīšanas datums un laiks}
```


4.3.2 Atļauju noskaidrošana

Lai izgūtu datus pieprasījumā ir jānorāda VIRSIS izsniegtās atļaujas numurs. Datu patērētājam 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 izgūt ar metodi: https://test.dagr.gov.lv/current_user_permissions

Metodes izsaukuma piemērs:

```
curl --location --request GET 'https://test.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": "fpr_person",
    "permission_number": "32423536",
    "purpose_of_use": "testēšanai, pilnas tiesības uz personas datiem",
    "use_count": 0,
    "use_count_quota": 0,
    "valid_from": null,
    "valid_to": null,
    "access_control_tree": {
      "name": "fpr_person",
      "read_access": "full"
    }
  },
  {
    "service_name": "fpr_person",
    "permission_number": "23532235",
    "purpose_of_use": "testēšanai, daļējas tiesības uz personas datiem",
    "use_count": 0,
    "use_count_quota": 0,
    "valid_from": null,
    "valid_to": null,
    "access_control_tree": {
      "name": "fpr_person",
      "read_access": "partial",
      "properties": [
        {
          "name": "PersonCode",
          "read_access": "full"
        },
        {
          "name": "ActualPersonCode",
          "read_access": "full"
        }
      ]
    }
  }
]
```

```

        "name": "PersonsCode",
        "read_access": "none"
    },
    {
        "name": "PersonsName",
        "read_access": "partial",
        "properties": [
            {
                "name": "ReasonCode",
                "read_access": "none"
            },
            {
                "name": "ReasonName",
                "read_access": "none"
            },
            {
                "name": "ValidFrom",
                "read_access": "none"
            },
            {
                "name": "FirstName",
                "read_access": "full"
            },
            {
                "name": "LastName",
                "read_access": "full"
            },
            {
                "name": "OtherNames",
                "read_access": "full"
            }
        ]
    },
    {
        "name": "PersonsBirth",
        "read_access": "none"
    },
    {
        "name": "PersonsDocument",
        "read_access": "none"
    },
    {
        "name": "PersonsResidence",
        "read_access": "none"
    },
    {
        "name": "PersonsDeath",
        "read_access": "none"
    }
}
]
}

```

```
}  
]
```

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 RESTful tipa pakalpes ar adresi šādā formā: https://test.dagr.gov.lv/{datu_kopa}/{objekta_identifikators}

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://test.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://test.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.

4.3.4 Datu izmaiņu izgūšana un pilnas datu kopas pārļasišana

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://test.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 atlasi. 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 atlasi;
- **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://test.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"
  }
}
]

```

4.3.5 Datu izmaiņu monitorings

Datu izmaiņu monitorings ļauj sekot līdzi izmaiņām datos datu patērētāja noteiktiem datu kopas ierakstiem. Lai uzsāktu datu izmaiņu monitoringu ir jādefinē **izmaiņu monitoringa kopa** jeb jānosaka kura datu kopa un kādas VIRSIS atļaujas ietvaros tiks monitorēta. Kad izmaiņu monitoringa kopa ir definēta, tai jāpievieno saraksts ar identifikatoriem ierakstiem kuri tiks monitorēti. Tālāk datu patērētājam ir iespēja atlasīt izmaiņu datus, kuros parādīsies informācija par tiem datu kopas ierakstiem, kam dati mainījušies kopš brīža kad ieraksta identifikators tika pievienots monitoringa kopai. Gadījumā, ja ieraksta identifikatoru pēc tam no izmaiņu monitoringa kopas noņem, tad dati par ieraksta izmaiņām pieejami tikai līdz brīdim kamēr šis identifikators tika no datu izmaiņu kopas noņemts.

4.3.5.1 Izmaiņu monitoringa kopas definēšana

Izmaiņu monitoringa kopu iespējams definēt DAGR pašapkalpošanās portālā.

Vispirms automātiskās uzmeklēšanas logā ir jāizvēlas datu patērētājs kura vārdā lietotājs darbojas no datu patērētājiem kuru pieteikumos lietotājs norādīts kā kontaktpersona.

Tālāk parādās saraksts ar visām definētajām izmaiņu monitoringa kopām, kur par katru izmaiņu monitoringa kopu parādās izmaiņu monitoringa kopas nosaukums, datu kopas nosaukums un VIRSIS atļaujas numurs.

DAGR > Izmaiņu monitorings > Sabiedrība ar ierobežotu atbildību "UNISO"

Izmaiņu monitorings

DATU PATĒRĒTĀJS

Sabiedrība ar ierobežotu atbildību "UNISO"

UR_TST_KOPA_1	ur_legal_entity	nr 2
UR_TST_KOPA_2	ur_legal_entity	nr 2
teacher_changes	viis_teacher	40003562863/viis_teacher
test_001	vsaa_person_pension	var skatīties vsaa

No saraksta iespējams atvērt apskatei eksistējošu izmaiņu monitoringa kopu vai definēt jaunu izmaiņu monitoringa kopu. Definējot jaunu izmaiņu monitoringa kopu ir jānorāda izmaiņu monitoringa kopas nosaukums, tad jāizvēlas monitorējamā datu kopa no datu kopu saraksts un visbeidzot jāizvēlas VIRSIS atļaujas no datu patērētājam esošajām atļaujām uz izvēlēto datu kopu.

DAGR > Izmaiņu monitorings > Sabiedrība ar ierobežotu atbildību "UNISO" > Jauna kopa

Jauna monitoringa kopa

NOSAUKUMS: Jauna_izmainu_kopa
Tehniskais kopas nosaukums — pieļaujami latīņu burti, cipari un zemsvitra.

DATU KOPA: fpr_person

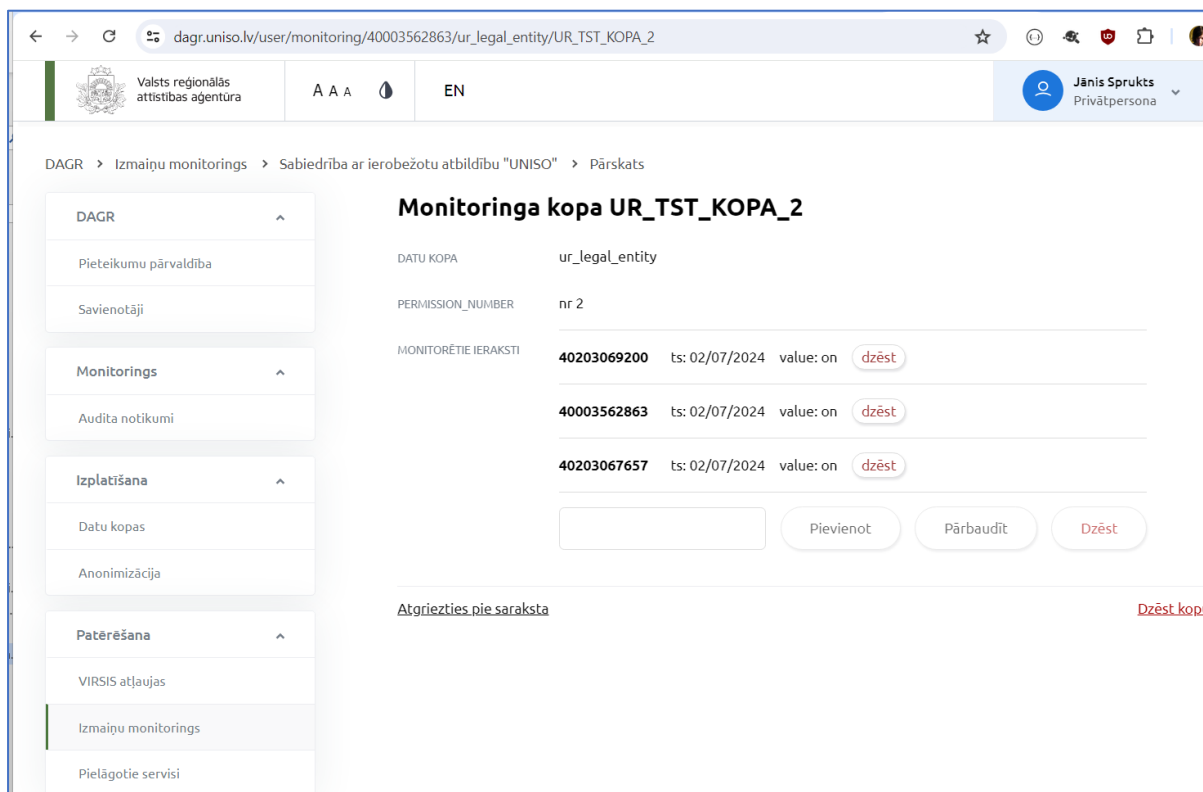
PERMISSION_NUMBER: |

atļauja 40003562863/fpr_person
atļauja 40003562863/fpr_person
multi_permission_test_01
multi_permission_test_02

Izveidot kopu

Kad izmaiņu monitoringa kopa ir definēta, tad tai var pievienot, pārbaudīt, dzēst monitorējamo ierakstu identifikatorus. Identifikatoriem ir jābūt tieši tādiem kādi tie tiek izmantoti monitorējamā datu kopā.

Darbības ar monitorējamo ierakstu identifikatoriem var izdarīt gan DAGR pašapkalpošanās portālā, gan izmantojot DAGR REST API. DAGR pašapkalpošanās portālā tas ir izdarāms izmaiņu monitoringa kopas kartiņā. Ņemot vērā, ka monitorējamo identifikatoru var būt ļoti daudz, kartiņā tiek attēloti tikai pēdējie 20. Kartiņā var ievadīt monitorējamā ieraksta identifikatoru un izmantojot spiedpogas vai nu to pievienot, dzēst vai pārbaudīt tā esamību izmaiņu monitoringa kopā.



The screenshot shows a web browser window with the URL `dagr.uniso.lv/user/monitoring/40003562863/ur_legal_entity/UR_TST_KOPA_2`. The page title is "Monitoringa kopa UR_TST_KOPA_2". The main content area displays a table of monitoring records:

MONITORĒTIE IERAKSTI	ts:	value:	action
40203069200	02/07/2024	on	<button>dzēst</button>
40003562863	02/07/2024	on	<button>dzēst</button>
40203067657	02/07/2024	on	<button>dzēst</button>

Below the table, there is an input field and three buttons: Pievienot, Pārbaudīt, and Dzēst. At the bottom right, there is a link [Dzēst kopu](#).

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 - skatīt "5.2.2 Datu kopas pārceļšana uz pamata sadaļu (dataset)". Tas varētu būt noderīgi, piemēram, monitorējot iestāžu reģistrācijas numurus nodokļu parādnieku datu kopā. Tādā gadījumā varētu monitorēt iestādes parādnieka statusu neatkarīgi no tā vai iestāde ir vai nav uz monitoringa brīdi parādnieks.

Izmaiņu monitoringa automatizācijas vajadzībām visas ar izmaiņu monitoringa kopām veicamās darbības ir pieejamas arī DAGR REST API servisu formā. Servisu pilns apraksts pieejams:

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

Servisus var izsaukt izmantojot datu patērētāja drošības talonu (skatīt sadaļu 4.3.1 Autentifikācija (drošības talona pieprasīšana)).

Izmaiņu monitoringa vajadzībām pieejami šādi DAGR REST API servisi:

- **PUT /monitoring_set/{registration_number}/{dataset}/{name}**
Pievieno jaunu monitoringa kopu
- **DELETE /monitoring_set/{registration_number}/{dataset}/{name}**
Dzēš eksistējošu monitoringa kopu
- **GET /monitoring_set**
Izgūst sarakstu ar datu patērētāja izmaiņu monitoringa kopām

- **PUT /monitor/{monitoring_set_name}/{registration_number}/{dataset}/{key}**
Pievieno jaunu identifikatoru izmaiņu monitoringa kopai. Viens un tas pats identifikators var tikt pievienots vairākas reizes. DAGR darbību tas negatīvi neietekmē, bet pašam datu patērētājam tad zūd iespēja noteikt pirmreizējo identifikatora pievienošanas datumu un laiku.
- **DELETE /monitor/{monitoring_set_name}/{registration_number}/{dataset}/{key}**
Dzēs izmaiņu monitoringa kopai pievienotu identifikatoru
- **GET /monitoring_is_on/{monitoring_set_name}/{registration_number}/{dataset}/{key}**
Pārbauda vai identifikators tiek monitorēts
- **GET /monitored_keys_changes/{monitoring_set_name}/{registration_number}/{dataset}**
Ļauj izgūt sarakstu ar visiem monitorējamajiem identifikatoriem. Piebilde, identifikatoru saraksts tiek glabāts rindas formā un vienādu identifikatoru gadījumā aktuāla ir saraksta pēdējā vērtība. Neaktuālās identifikatoru vērtības pēc laika no rindas tiek dzēstas.

4.3.5.2 Izmaiņu monitoringa kopas nolasīšana

Izmaiņu monitoringa kopas dati ir nolasāmi tāpat kā izmaiņu dati (4.3.4 Datu izmaiņu izgūšana un pilnas datu kopas pārslasīšana) ar divām atšķirībām:

- Izsaukumam jāpievieno parametrs “monitoring_set” kurā tiek norādīts monitoringa kopas nosaukums;
- Izsaukuma galvenē “X-DAGR-Permission” jānorāda tā pati atļauja, kas izmantota monitoringa kopas definēšanai.

Vispārīgā formā izmaiņu monitoringa dati nolasāmi izsaucot šādu servisu:

https://{dagr_vides_adrese}/{datu_kopas_nosaukums} changes/?monitoring_set={izmaiņu_monitoringa_kopas_nosaukums}

Izsaukuma piemērs:

```
curl --location
'https://dev.dagr.gov.lv/vbtai_orphan_certificate_changes/?offset=0&limit=100&monitoring_set=VBTAI_demo_1' \
--header 'X-DAGR-Permission: 40003562863/vbtai' \
--header 'Authorization: .....
```

Šim izsaukumam 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.

Piebilde par sekošanu līdz izmaiņu datiem: DAGR neatzīmē kuri izmaiņu monitoringa dati jau ir nolasīti. Sekot līdz izmaiņu nolasīšanas stāvoklim ir datu patērētāja uzdevums. Tas ir viegli izdarāms izmantojot parametru *offset* vai *timestamp*. Ja datu patērētājam savā informācijas sistēmā ir process, kas ar noteiktu regularitāti pieprasa izmaiņu monitoringa datus, tad pēc datu nolasīšanas ir

jā saglabā pēdējā nolasītā ieraksta *timestamp* vai *offset* vērtība. Šo saglabāto vērtību nākošajā procesa izpildes ciklā jāpadod izmaiņu monitoringa datu izgūšanas servisam kā ieejas parametru un tas garantēs, ka nolasītas tiek tikai jaunpienākušās izmaiņas.

Piebilde par datu dzēšanu: DAGR izmaiņu monitoringa kopā katram ierakstam ilglaicīgi uzglabā tikai pēdējās izmaiņas.

4.3.6 Lielapjoma datu (failu) izgūšana

Daļai no DAGR izplatāmajām datu kopām ir nodrošināta iespēja izgūt lielapjoma datus (failus).

Uz katru datu kopu, kurai ir pieejami faili, datu izgūšanai ir izveidots GET RESTful tipa pakalpojums pāris ar adresēm šādā formā:

- https://test.dagr.gov.lv/{datu_kopa}/{objekta_identifikators}, kas satur izgūstamā faila metadatus;
- https://test.dagr.gov.lv/{datu_kopa}_file/{objekta_identifikators}, kas satur bināros datus jeb pašu failu. Atbildes dati tiek straumēti.

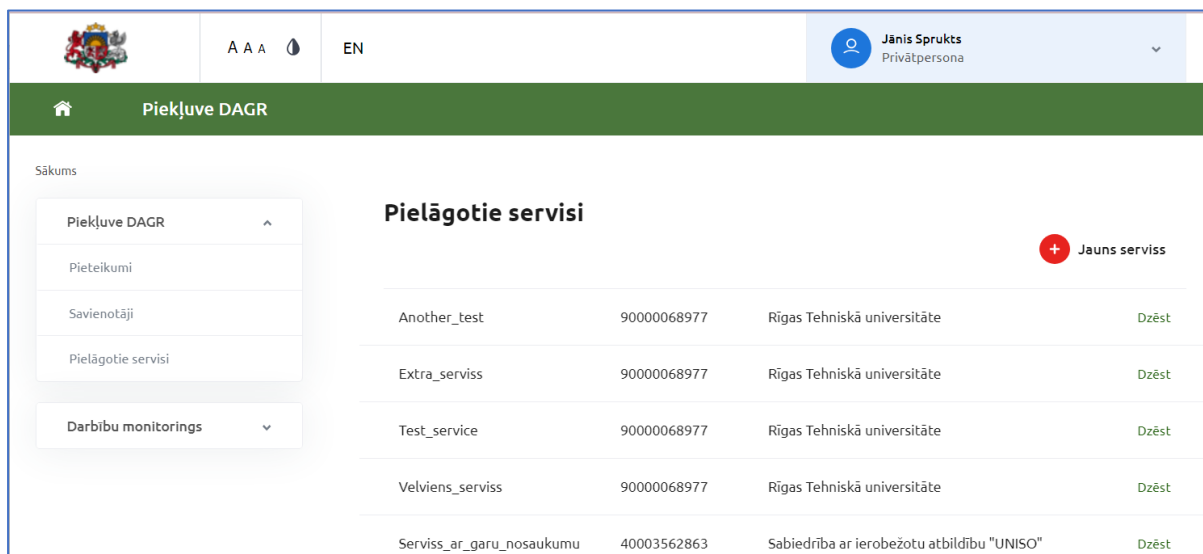
Faila izsaukuma piemērs (uzņēmuma gada pārskats pēc reģistrācijas numura un gada):

```
curl --location 'https://test.dagr.gov.lv/ur_annual_report_file/40003228733/2019' \
---header 'Authorization: Bearer {autentifikācijas ietvaros saņemtais drošības talons}' \
--header 'X-DAGR-Permission: {VIRSIS atļaujas numurs}'
```

4.3.7 Pielāgoti servisi

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

Pielāgota servisa definēšana. Pielāgotus servisius datu patērētājs var apskatīt un definēt pašapkalpošanās portāla sadaļā “Piekljuve DAGR” → “Pielāgotie servisi”.



The screenshot shows the 'Piekljuve DAGR' web application interface. At the top, there is a navigation bar with the Latvian coat of arms, language settings (A A A, EN), and a user profile for 'Jānis Sprukts Privātpersona'. Below the navigation bar is a green header with a home icon and the text 'Piekljuve DAGR'. The main content area is titled 'Sākums' and features a sidebar menu on the left with options: 'Piekljuve DAGR', 'Pieteikumi', 'Savienotāji', 'Pielāgotie servisi', and 'Darbību monitorings'. The main area is titled 'Pielāgotie servisi' and contains a table of custom services. A red '+' button labeled 'Jauns serviss' is located in the top right corner of the table area. The table lists five services with columns for name, ID, location, and action.

Service Name	ID	Location	Action
Another_test	90000068977	Rīgas Tehniskā universitāte	Dzēst
Extra_serviss	90000068977	Rīgas Tehniskā universitāte	Dzēst
Test_service	90000068977	Rīgas Tehniskā universitāte	Dzēst
Velviens_serviss	90000068977	Rīgas Tehniskā universitāte	Dzēst
Serviss_ar_garu_nosaukumu	40003562863	Sabiedrība ar ierobežotu atbildību "UNISO"	Dzēst

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 patērētāja reģistrācijas numurs, kuram serviss tiek veidots. Patērētāja nosaukums pēc reģistrācijas numura tiek pielasīts automātiski.
- **Atlases parametru.** 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://test.dagr.gov.lv/api>. Ja pamata servisa ceļš satur atlases parametru, 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 patērētājam pieejami izmantotās atļaujas ietvaros.

Pielāgotais serviss Extra_serviss

SERVISA NOSAUKUMS:

REĢISTRĀCIJAS NUMURS: 🔍

PATĒRĒTĀJS:

+ Pievienot parametru

PARAMETRI

PATH_PARAMETER: ✖

PATH_PARAMETER: ✖

+ Pievienot datu kopu

DATU KOPAS

Datu kopa 1

URL_TEMPLATE: ✖

PERMISSION_NUMBER:

+ Pievienot īpašību

Datu kopa 2

URL_TEMPLATE: ✖

PERMISSION_NUMBER:

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

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

4.3.8 Pēdējo izmaiņu datums

Ja datu patērētājam ir atļauja veikt izmaiņu izgūšanu (skatīt **Error! Reference source not found.** REF_Ref167189131 \h **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 patērētājam ir atļauja izgūt datus tikai pēc ieraksta identifikatora, tad pēdējo izmaiņu datumu un laiku var uzzināt izmantojot palīg-servisu:

https://test.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://test.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 pieprasījumu audits

Visi datu pieprasījumi tiek auditēti. Gan datu patērētājiem, gan datu devējiem ir piekļuve audita datiem. Datu patērētājiem ir piekļuve audita pierakstiem par sevis veiktajiem datu pieprasījumiem. Datu devējam ir piekļuve pieprasījumiem, kuri veikti par datu devēja datu kopām.

4.4.1 Audita apskate portālā

Audita datus iespējams apskatīties DAGR portāla sadaļā “Darbību monitorings” -> “Manas veiktās darbības”.

4.4.2 Audita izgūšana caur REST API

Audita datus var izgūt arī caur DAGR REST API. Audita datu izguves serviss vispārīgā formā:

https://test.dagr.gov.lv/audit_list?action_time_from={datums_un_laiks_no}&action_time_to={datums_un_laiks_līdz}

Servisa izsaukuma piemērs:

```
curl --location 'https://test.dagr.gov.lv/audit_list?action_time_from=2024-01-16T00%3A00%3A00&action_time_to=2024-01-17T01%3A00%3A00' \
```

```
--header 'Authorization: Bearer {autentifikācijas ietvaros saņemtais drošības talons}'
```

4.5 Datu pieprasījumu audits

Visi datu pieprasījumi tiek auditēti. Gan datu patērētājiem, gan datu devējiem ir piekļuve audita datiem. Datu patērētājiem ir piekļuve audita pierakstiem par sevis veiktajiem datu pieprasījumiem. Datu devējam ir piekļuve pieprasījumiem, kuri veikti par datu devēja datu kopām.

4.5.1 Audita apskate portālā

Audita datus iespējams apskatīties DAGR portāla sadaļā “Darbību monitorings” -> “Manas veiktās darbības”.

Gan datu patērētājiem, gan datu devējiem ir piekļuve audita datiem sadaļā “Darbību monitorings” -> “Audita notikumu pārskats”. Datu patērētāju pārstāvjiem ir pieejams audits par datu patērētāja veiktajiem datu pieprasījumiem. Datu devējiem ir pieejams audits par datu ielādes notikumiem un datu patērētāju pieprasījumiem datu devēja datiem. Administratoriem ir pieejama visi audita notikumi.

Datu patērētājiem un devējiem ir nodrošinātas šādas audita filtrēšanas iespējas:

- **Datu patērētājs** – filtrs ļauj meklēt iestādi sarakstā no iestādēm, kurām datu patērētāja pieteikumā lietotājs norādīts kā kontaktpersona, vai norādīt ierakstot iestādes reģistrācijas numuru;
- **Datums, no** – filtrs ļauj norādīt datumu un laiku sākot no kura audita ieraksti atlasāmi;
- **Datums, līdz** – filtrs ļauj norādīt datumu un laiku līdz kuram audita ieraksti atlasāmi;
- **Darbības veids** – filtrs ļauj norādīt notikuma veidu par kuru atlasāmi audita ieraksti;
- **Lietotāja veids** – filtrs ļauj norādīt kāda veida lietotāju veiktās darbības atlasāmas;
- **Subjekta tips** – filtrs ļauj norādīt par kāda veida subjektiem veikto datu pieprasījumu audita ierakstus atlasīt. Šis filtrs strādā tikai uz datu pieprasījuma veida audita notikumiem;
- **Pieprasījuma IP adrese** – filtrs ļauj norādīt IP adresi no kuras veikti audita notikumi atlasāmi.
- **Klienta ID** – filtrs ļauj norādīt identifikatoru datu patērētāja savienotājam, kurš tika izmantots darbības veikšanai;
- **Subjekta identifikatori** – filtrs ļauj norādīt vienu vai vairākus subjekta identifikatorus, piemēram, personas kodus, par kuriem veiktos datu pieprasījumus atlasīt. 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 patērētāja pārstāvim jānorāda sava iestāde kā datu patērētājs.

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 patērētājs, ja audita notikumus atlasa datu patērētāja pārstāvis. Personām, kuras pārstāv gan datu patērētāju/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 parametru, piemēram, samazinot atlasītā laika posma garumu.

4.5.2 Audita izgūšana caur REST API

Audita datus var izgūt arī caur DAGR REST API. Audita datu izguves serviss vispārīgā formā:

https://test.dagr.gov.lv/audit_list?action_time_from={datums_un_laiks_no}&action_time_to={datums_un_laiks_līdz}

Servisa izsaukuma piemērs:

```
curl --location 'https://test.dagr.gov.lv/audit_list?action_time_from=2024-01-16T00%3A00%3A00&action_time_to=2024-01-17T01%3A00%3A00' \
--header 'Authorization: Bearer {autentifikācijas_ietvaros_saņemtais_drošības_talons}'
```

Atbildes piemērs:

```
[
  {
    "action_time": "2024-07-01T07:10:30.439Z",
    "related_identity": "40003562863",
    "consumer_identity": "40003562863",
    "consumer_name": "UNISO_test_connector_1",
    "consumer_id": "40003562863",
    "request_ip": "192.168.173.241",
    "request_session_id":
"$s0$e0801$2U4DTdm1TcCbh6EOWYOBjg==$PUN4u+lHZEKZRz1A7PRkamyVyTVnVMsLL7sSA615pxE=",
    "request_info":
{"action": "get", "service": "fpr_person", "key": ["32595465639"], "params": null},
    "producer_reg_nr": "90000054163",
    "producer_name": "Pilsonības un migrācijas lietu pārvalde (PMLP)",
    "permission_number": "atlauja 40003562863/fpr_person",
    "purpose_of_use": "educational",
    "successful": true,
    "subject_id": "32595465639",
    "subject_type": "persona",
    "endpoint_id": "dc1.dagr-http-5b7466894-4mk51"
  }
]
```

Atbildes datu skaidrojums:

- action_time – notikuma laiks;
- related_identity – saistītais reģistrācijas numurs (tipiski sakrīt ar datu patērētāja reģ.nr.);
- consumer_identity – datu patērētāja reģistrācijas numurs;
- consumer_name – datu patērētāja savienotāja nosaukums;
- consumer_id – datu patērētāja 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).

5 Datu devēja integrācija

5.1 Pieteikšanās par datu devēju

Pieteikties par datu devēju un reģistrēt datu devēja savienojumu var DAGR pašapkalpošanās portālā, kurš pieejams adresē: <https://test.dagr.gov.lv/>

Lai izmantotu pašapkalpošanās portālā pieejamo funkcionalitāti, lietotājam ir jāautenticējas ar VPM, izmantojot vienu no piedāvātajām opcijām: elektroniskā identifikācija, identifikācija ar bankas starpniecību, vai citas. Alternatīvi pieteikties par datu patērētāju var arī ārpus sistēmās iesniedzot parakstītu pieteikumu DAGR pārzinim.

Par datu devēju ar VPM autenticēti lietotāji var pieteikties portāla sadaļā “Pieteikumi”:

EN

Kaspars Eglītis
Fiziska persona

Pieļauve DAGR Darbību monitorings

Some alert text

Sākums > Pieteikumu pārvaldība

Pieļauve DAGR

Pieteikumi

Savienotāji

Darbību monitorings

Pieteikumu pārvaldība

Jums ir šādi pieteikumi:

Sabiedrība ar ierobežotu atbildību "UNISO" 40003562863	✓
ABC test company 40003627089 40003627089	⌛
Pilsonības un migrācijas lietu pārvalde 40001006969	✓
Pilsonības un migrācijas lietu pārvalde 40000006969	⌛

DAGR datu savienotāja licences pieprasījums

Datu devēja pieteikums

Datu patērētāja pieteikums

KONTAKTI

Vides aizsardzības un reģionālās attīstības ministrija
Peldu iela 25, Rīga, LV-1494, Latvija

Pieteikties par datu devēju var valsts pārvaldes iestādes, ja lietotājs portālam pieslēdzies kā fiziskā persona. Nospiežot uz “Datu devēja pieteikums” lietotājs tiek pārvirzīts uz attiecīga pieteikuma aizpildīšanu.

Pieteikumā iesniedzēja vārds, uzvārds un personas kods tiek aizpildīts no lietotāja autentifikācijas informācijas, savukārt tālrunis un e-pasts jānorāda ar roku. Iestāde, kura tiek pieteikta kā datu devējs ir jāizvēlas norādot iestādes reģistrācijas numurs un nospiežot uz vizuālā elementa ar palielināmā stikla ikonu. Ja reģistrācijas numurs būs korekti norādīts, tad automātiski ielasīsies iestādes nosaukums.

Vārdtelpas laukā jāievada iestādes saīsinājums vai nosaukums saīsinātā formā. Pašvaldību gadījumā var ievadīt teritoriālās vienības nosaukumu.

Papildus var norādīt vienu vai vairākas tehniskās kontaktpersonas, kuras datu devēja vārdā varēs pārvaldīt datu devēja savienojumus.

Pēc pieteikuma iesniegšanas tas tiek nosūtīts DAGR pārvaldniekam un līdz apstiprināšanas vai noraidīšanas brīdim nonāk statusā "Gaidīšanā".

Pieteikuma noraidīšanas vai iepriekš apstiprināta pieteikuma anulēšanas gadījumā, iespējams iesniegt atkārtotu pieteikumu. Piesakoties atkārtoti lietotājs tiek pārvirzīts uz pieprasījuma lapu, kur ir iespējams rediģēt iepriekš pieteiktā pieprasījuma lauku vērtības.

5.2 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 vidē ir izmantojama datu nodošanas testēšanai un atklūdošanai. 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 patērētājiem izmantojot VIRSIS piešķirtās datu piekļuves atļaujas.

5.2.1 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);
 - **vbta_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* paradigmā, 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 “Datu izplatīšana” -> “Datu kopas”.

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 kartīnā.

```
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, par atslēgu jā izvēlas lauki ar biznesa nozīmi, piemēram, personas kods, reģistrācijas numurs, adreses kods, dokumenta numurs, u.c. Gadījumā, ja ieraksta unikālītātes nodrošināšanai nepieciešams izmantot kādu lauku, kuram nav biznesa nozīmes, tad tas liekams atslēgas beigās un iekļaujams iekavās. Iekavās iekļautie lauki netiks izmantoti meklēšanā. 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.

```
fields:
```

```
- {lauka_nosaukums} {lauka_formāts} : {lauka apraksts}
```

```
- {salikta elementa nosaukums} * :
  comments      : {salikta elementa komentārs}
  fields:
```

| - {lauka_nosaukums} {lauka_formāts} : {lauka apraksts}

Datu kopas definīcijai jānoslēdzas 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);
- 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 “6.1 Datu kopu definīcijas”.

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

5.2.2 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”. Šajā sadaļā ir saraksts ar visu datu devēju reģistrētajām datu kopām priekšlikumu vidē (sandbox).

Priekšlikumu vide			
New_dataset_3	40003562863	Sabiedrība ar ierobežotu atbildību "UNISO"	Dzēst
rightsOfRepresentation	40003562863	Sabiedrība ar ierobežotu atbildību "UNISO"	Dzēst
tst_company_service_report	40003562863	Sabiedrība ar ierobežotu atbildību "UNISO"	Dzēst
tst_company_service_report_anon	40003562863	Sabiedrība ar ierobežotu atbildību "UNISO"	Dzēst

No šī saraksta ir jāizvēlas uz pamata sadaļu pārceļamā datu kopa, tā jāatver un jāspiež poga “Saglabāt” – tas pārceļ datu kopu no priekšlikumu vides (sanbox) uz pamata sadaļu (dataset). Papildus tiks parādīts jautājums vai pārceļšanas ietvaros dzēst datu kopas definīciju priekšlikumu vidē.

The screenshot shows a web browser window with the URL `parvaldiba-dagr.uniso.lv/manager/proposals/40003562863/tst_company_service_report`. The page title is "Datu kopas priekšlikums tst_company_service_report". On the left, there is a navigation menu with categories like "DAGR", "Monitorings", and "Izplatīšana". The main form contains the following fields:

- DATASET_NAME:** `tst_company_service_report`
- REGISTRATION_NUMBER:** `40003562863`
- DATU AVOTS:** `UNISO`
- COMMENT:** (empty)
- STRUCTURE:**

```

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

```

At the bottom of the form, there are two buttons: "Saglabāt" (Save) and "Uz sarakstu" (Back to list).

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

The screenshot shows the "Datu kopas" (Datasets) page. The left navigation menu is the same as in the previous screenshot. The main content area is titled "datasets" and contains a list of datasets with a "Dzēst" (Delete) button next to each:

Dataset Name	Owner	Action
<code>vsaa_person_pension</code>	-	Dzēst
<code>test_custom_events</code>	-	Dzēst
<code>vvd_drn_maksataji</code>	-	Dzēst
<code>November_dataset_2</code>	-	Dzēst
<code>vzd_address</code>	-	Dzēst
<code>vvd_drn_atbrivojumi</code>	-	Dzēst
<code>Wednesday_dataset</code>	-	Dzēst
<code>tst_company_service_report_anon</code>	-	Dzēst
<code>vdeavk_person_disability</code>	-	Dzēst
<code>viis_teacher</code>	-	Dzēst
<code>viis_person_study</code>	-	Dzēst
<code>tst_company_service_report</code>	-	Dzēst

Lai pabeigtu datu kopas pārceļšanu nepieciešams atvērt pārceltās 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://test.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.

The screenshot shows the DAGR web interface for managing datasets. The browser address bar shows 'parvaldiba-dagr.uniso.lv/manager/datasets/sif_honourable_family_card_anon'. The user is logged in as 'Jānis Sprukts' (Privātpersona). The main content area displays the configuration for the dataset 'sif_honourable_family_card_anon'. The configuration includes:

- DATASET_NAME:** sif_honourable_family_card_anon
- COMMENT:** (empty field)
- STRUCTURE:**

```

name: sif_honourable_family_card_anon
comments: "Latvijas Goda ģimenes apliecinājumu dati"
data_owner_code: "90001237779"
data_owner_name: Sabiedrības integrācijas fonds
data_namespace: sif
data_description: Latvijas Goda ģimenes apliecinājumu dati
subject_type: persona
subject_id_fieldname: personas_kods
is_public_dataset: true
key: personas_kods
fields:
- numurs: Apliecinājuma numurs
- vārds: Apliecinājuma turētāja vārds
- uzvārds: Apliecinājuma turētāja uzvārds
- personas_kods: Apliecinājuma turētāja personas kods
- tips: Apliecinājuma tips
- izveidošanas_datums date: Apliecinājuma izveidošanas datums
- derīguma_termiņš date: Apliecinājuma derīguma termiņš
- ir_anulēts boolean: Pazīme, ka apliecinājums ir anulēts
- hash: Apliecinājuma kontrolsumma. 30 ciparu skaitlis, kas sākas ar 1234, turpinās ar apliecinājuma numuru un noslēdzas ar 14 ciparu gadījuma skaitli.

```

At the bottom, there are buttons for 'Saglabāt' (Save) and 'Atgriezties pie saraksta' (Return to list).

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.

5.2.3 Specializētās datu kopas

5.2.3.1 Publiskie klasifikatori

Īpašs uz DAGR nododamās datu kopas paveids ir publiskie klasifikatori. Publisko klasifikatoru dati ir pieejami visiem datu patērētājiem 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_vecāka_koda_lauks}: parent_code
  {vecāka_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

```

5.2.3.2 *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 apliecinošo dokumentu maiņa, dzīvesvietas maiņa, utt.

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

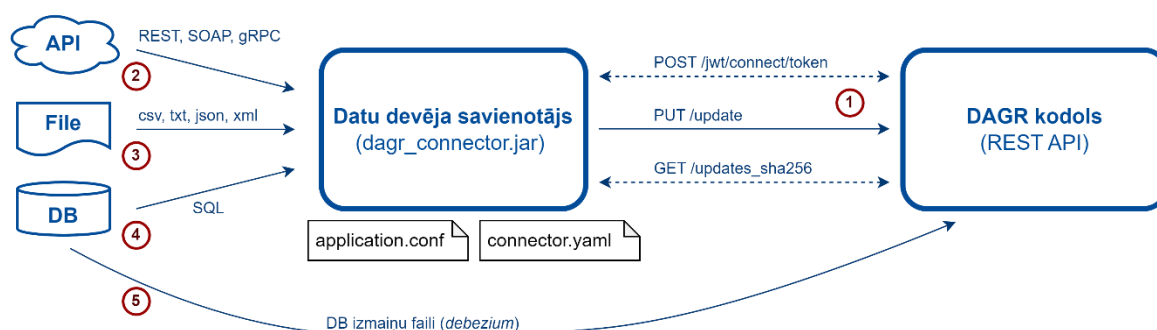
```

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

```

Datu struktūras kartējums (5.5.2.3 "Datu struktūru kartējums") biznesa notikumiem realizējams tāpat kā citām datu kopām.

5.3 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 “5.5.1 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 pielāgumus ne datu devēja ne DAGR pusē, bet tas var prasīt būtiskas izmaiņas avota DBVS konfigurācijā.

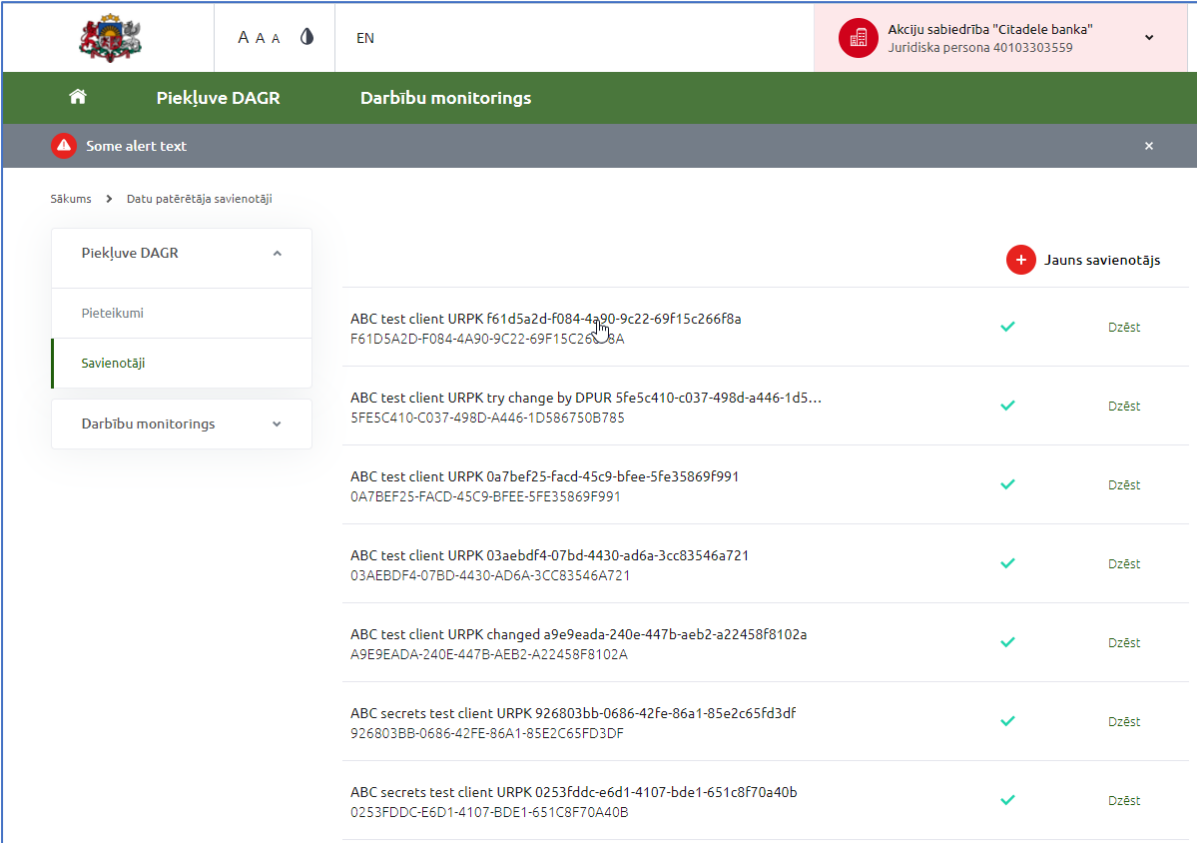
Ja datu nodošana tiek organizēta izmantojot datu devēja savienotāju (otrais līdz ceturtais datu nodošanas mehānisms), tad pastāv vēl jautājums kur šo savienotāju izvietot. Pēc noklusējuma datu devēja savienotājs uzstādāms datu devēja infrastruktūrā. Taču vienojoties ar DAGR pārvaldnieku un atverot nepieciešamās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

5.4 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 "Piekļuve DAGR" apakš sadaļā "Savienotāji". Šajā sadaļā ir pieejams saraksts ar visiem savienojumiem:



The screenshot shows the 'Piekļuve DAGR' web application interface. At the top, there is a header with the Latvian coat of arms, language settings (A A A), and 'EN'. A red notification bubble indicates 'Akciju sabiedrība "Citadele banka" Juridiska persona 40103303559'. Below the header, there is a navigation bar with 'Piekļuve DAGR' and 'Darbību monitorings'. A grey alert bar displays 'Some alert text'. The main content area is titled 'Sākums > Datu patērētāja savienotāji'. On the left, there is a sidebar menu with options: 'Piekļuve DAGR', 'Pieteikumi', 'Savienotāji', and 'Darbību monitorings'. The 'Savienotāji' option is selected. The main area shows a table of connections with a '+ Jauns savienotājs' button. The table contains the following data:

Connection Name	Status	Action
ABC test client URPK f61d5a2d-f084-4a90-9c22-69f15c266f8a F61D5A2D-F084-4A90-9C22-69F15C266F8A	✓	Dzēst
ABC test client URPK try change by DPUR 5fe5c410-c037-498d-a446-1d5... 5F5C410-C037-498D-A446-1D586750B785	✓	Dzēst
ABC test client URPK 0a7bef25-facd-45c9-bfee-5fe35869f991 0A7BEF25-FACD-45C9-BFEE-5FE35869F991	✓	Dzēst
ABC test client URPK 03aebdf4-07bd-4430-ad6a-3cc83546a721 03AEBDF4-07BD-4430-AD6A-3CC83546A721	✓	Dzēst
ABC test client URPK changed a9e9eada-240e-447b-aeb2-a22458f8102a A9E9EADA-240E-447B-AEB2-A22458F8102A	✓	Dzēst
ABC secrets test client URPK 926803bb-0686-42fe-86a1-85e2c65fd3df 926803BB-0686-42FE-86A1-85E2C65FD3DF	✓	Dzēst
ABC secrets test client URPK 0253fddc-e6d1-4107-bde1-651c8f70a40b 0253FDDC-E6D1-4107-BDE1-651C8F70A40B	✓	Dzēst

Lai reģistrētu jaunu savienojumu jāaizpilda forma:

Datu patērētāja savienotāja reģistrācija

NOSAUKUMS

IDENTIFIKATORS

Aktīva

DROŠĪBAS TALONA DZĪVES LAIKS (SEKUNDĒS)

APRAKSTS

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. Drošības sertifikāta pievienošana datu devēja savienotāja reģistrācijas scenārijā neatšķiras no datu patērētāja savienotāja reģistrācijas scenārija – detalizētāku aprakstu par sertifikātiem skatīt sadaļā “4.2.2 Datu patērētāja savienotāja reģistrēšana”.

5.5 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.

5.5.1 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 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://test.dagr.gov.lv/api-docs/update>

5.5.1.1 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 (testa vidē): <https://test.dagr.gov.lv/jwt/connect/token>

Drošības talona izsaukuma piemērs:

```
curl --location 'https://test.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.eyJzdWIiOiJlJEQUdSX0FLQ19URVNU
```



```
XzIiLCJqdGkiOiJkOTQwZTliNC0yYjgwLTQ4NzUtYmEyMS1mYmVmYmVkJkYzUzZjIiLCJpc3MiOiJEQUdSX0F  
LQ19URVNUXzIiLCJhdWQiOiJlcm46VEVTVClMTkI6REFHU5TVFMvY29ubmVjdC90b2t1biIsImV4cCI6Ij  
E3MDkyOTQ5NjQiLCJmYmYiOiIxNzA5MjkmZy0In0.Xq3bZNI62hS9OFYx-  
gNBEbh2SwQt72to_gL3Xwa9yh_uyUqd5__tDJuXkz-  
F03Jn0CdcRKcdeitrKTtSCxIlP3WqQpbotsPHZvuF0ek26koyhtGhK2lw6RNF8mPkIktb4L6F0GuTjwc9Zm  
Q10lYo86u6_ZZWe2sS8aISJAUBmClW5cIt_qc3xY8no_UUJwDlyRGfxntGw3T_IyvIv0bPOLlOvyfmswpcI  
7I5S2qR5EIJlKggM4SninbGfB1j_u7HGSziO14kIXmOikdhDviyP0je9kYBcwZgwYeja19dBxolNkDd6A3B  
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

Postman piemērs, kas iekļauj arī drošības talona izsaukumu:

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

5.5.1.2 Strukturētu datu iesūtīšana

Strukturēti dati uz DAGR nododami izmantojot /update servisu:

- Apstiprinātām datu kopām servisa adrese veidojas šādi: **PUT**
{dagr_adrese}/update/{datu_kopas_nosaukums}
- Izmēģinājuma (smilškastes) datu kopām servisa adrese veidojas šādi: **PUT**
{dagr_adrese}/sandbox/{datu_devēja_reģ_nr}/update/{datu_kopas_nosaukums}

Nododamie dati iekļaujami servisa saturā (body) JSON formātā. Servisam var nodot uzreiz vairākus ierakstus, kur katrs ieraksts norādāms šādā formā:

```
{  
  "key": "string",  
  "timestamp": "2024-03-06T10:04:55.297Z",  
  "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” aizvieto 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://test.dagr.gov.lv/sandbox/40003562863/update/tst_company_service_report' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcm46VEVTVClMTkI6REFHU15TVFMiLCJzdWIiOiI0MDAwMzU2Mjg2MyIsImF1dG8iOiJlZCJwYmYyOjE3MTA0MDA3MTESImIhdCI6MTcxMDQwMDcxMSwibGVhbnRpdHkiOiI0MDAwMzU2Mjg2MyIsImNsaWVudF9pZCI6ImRlbW9fZGRzX2NvbW51Y3Rvcl8xiiwic2NvcGUiOlsicHJvZHVjZSIiXX0.aOahd4FnJXzt5rjBzw_zMtFWL0gBUiIT2rwIG7bcdZLkd8NqEsYtq2qOqkfnncwm-iSn9uYV-1OM6q02NHcXVA68NU56PQkX_DvM-VL251GFtZFR8z9Tu64J1EaBWI2DR7x6BG7ULxO_brZUaBSn6VrBkGxNe0YoerJ5uMJZex7q461ed5Uxck0Hl8-BE4eWSE-OJcNaTAhcDnLaCEiIQgiwL9jbbq4hmWTe_O_0EO5zwRv0QiuLhHQ7whBallI5fX0WSH0cXINT7GRwcpP5UuMbpQ_-dwnBcEtsS-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      \"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 “5.5.1.1 Drošības talona pieprasīšana”, savukārt viens no piemērā nododamajiem ierakstiem neeskeiptā 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"
        }
      ]
    }
  ]
}

```


5.5.1.3 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. Ilglaicīgi 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:

- Jāsigatavo 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.

5.5.1.4 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://test.dagr.gov.lv/file/vsaa/test_file_name.txt' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJpc3MiOiJlcm46VEVTVk1MTkI6REFFU55TVFMIjZdWIiOiI5MDAwM
TY2OTQ5NiIsImF1ZCI6WyJwcm9kdWNlciIsInVyb2pURVNUUx0QjEQUdSLLNUUy9yZXNvdXJjZXMiXSwiZXhwIjoxNzE
wMzI4MDI4LCJmYyI6IjE3OTQ5fVEVTVF9ERFNfMSIsInNjb3B1IjpbInByb2R1Y2VyI119.gZgNI6451nrLAMYp4UHRWLBtGI
vPjIBxPpP1j2xiU5RyUf2lraD3CWdmyFyBIRlhZSjmQ5ITvrtJrvlMm0ID4j3BPoDiUsrOtr-
60FEbR3rwYLruQyu_CiHDBLbyfyWr8KKSbyov834-37x1wrZjablC-t_tncs7Uyd9-pvQp_RjkVmEYgwfPLiq-
edOyczQQ4WwcYwVoYKgH05yA5TX3LQ7n4oOGvJvMa9KoZenU2a49y4bLAXwQGKtxV_kox75ABMXroaGT5111FpTRz1AnzC
Bh6n4D3s3D0novL2IpgYgBxA5jhybkwJPdlynD_n4SVQDJNlLp_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.

5.5.2 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://test.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. Atklūdošanas nolūkiem uzstādāms lielāks apjoms, produkcijas darbināšanas režīmam mazāks.

5.5.2.1 *Datu devēja savienotāja uzstādīšanas instrukcija*

Datu devēja savienotāja tehniskās uzstādīšanas instrukcijas (piemērs:

1. Instalēt java 17
2. Lejupielādet 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. Koriģe 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://test.dagr.gov.lv"
jwt.service.uri      = "https://test.dagr.gov.lv/jwt/connect/token"
jwt.payload-audience = "urn:TEST-LNB:DAGR.STS/connect/token"
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
```

5.5.2.2 *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
# Datu laika saīsināšana, ja te nav null. Iespējamās vērtības: seconds, millis, micros, nanos

data.timezone = "Europe/Riga"
# tiek izmantots, lai formatētu date/Time 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 iepriekšējās 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 neatrodas ienākošajā ("avota") datu kopā

target_server_uri = null
# DAGR serveris, kas saņems datus, piemēram "https://dagr.gov.lv/" vai "https://test.dagr.gov.lv/". Gadījumā, ja
# dati tiek nodoti uz izmēģinājuma vidi (sandbox), tad adrese jānorāda formā
# "https://test.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,
# "MIIEvQIBADANBgkqhkiw0BAQEFAASCBAcwggSjAgEAAoIBAQQDFrFlz6URnDrmITZAestc7qUk37y3wp2KJhUJG1+4
# mFt5fQTyM/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYLMFgVcNhWjMJYD
# MaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYLMFgV
# cNhWjMJYD MaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJp
# okwtYLMFgVcNhWjMJYD MaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkH
```

```
awmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/PyWVRtkOxDEJrdVlkQnj7B3VMb0vIXzHEglAN0Ilo90scVz+QkHawmxmAllJpokwtYlMfgVcNhWjMJYDMaPGnCSQKBgA1bWjtQ/EPaFqC7o8jNR+”
```

```
provider.client_id = null
```

```
# Datu devēja savienotāja identifikators, piemēram, "MY_ORG_connector_1"
```

```
jwt.service.uri = null
```

```
# Drošības talona pieprasījuma adrese, piemēram "https://test.dagr.gov.lv/jwt/connect/token "
```

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

```
jwt.payload-audience = null
```

```
# Drošības talona audience, piemēram testa vidē "urn:LNB:DAGR.STS/connect/token"
```

```
jwt.scope = "producer"
```

```
# Datu devēja savienotājam "producer", datu patērētājam "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
}
}
```

5.5.2.3 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 “5.2 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
```

5.5.2.4 *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 pārlicinās, 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 pārlicinās, 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.

5.5.2.5 Žurnalēšanas 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). Žurnalēšanas 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>

  <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>
```

5.5.3 Debeziem savienotājs

Alternatīva iepriekšējās nodaļās aprakstītajam datu devēja savienotājam ir datu nodošana uz DAGR izmantojot *Debezium* 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.

5.6 Personas datu izmantošanas žurnālpieraksti

DAGR nodrošina iespēju datu patērētāju 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://test.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
```

5.7 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.

5.7.1 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_skaits}
rules:
- {datu_lauka_nosaukums}: {anonimizācijas_veids} {veida_parametri}
- {datu_lauka_nosaukums}: {anonimizācijas_veids} {veida_parametri}
- ...
```

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 troksnis, 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 ***0000****). 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 ***0000*** 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 parametrā *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 nosacījumu (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:
  - anonimizēts saskaņots
  - anonimizēts iesniegts
  - anonimizēts 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

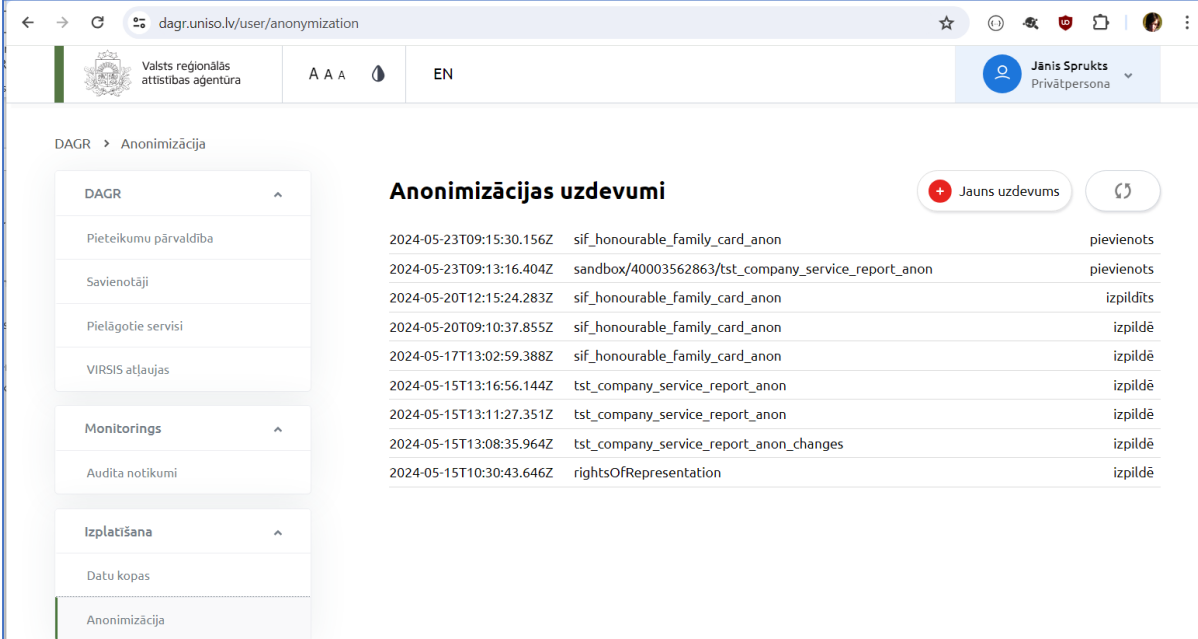
```

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.

5.7.2 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.



The screenshot shows the DAGR user interface for data anonymization. The browser address bar is 'dagr.uniso.lv/user/anonymization'. The page title is 'DAGR > Anonimizācija'. On the left is a navigation menu with categories: 'DAGR' (containing 'Pieteikumu pārvaldība', 'Savienotāji', 'Pielāgotie servi', 'VIRSIS atļaujas'), 'Monitorings' (containing 'Audita notikumi'), and 'Izplatīšana' (containing 'Datu kopas', 'Anonimizācija'). The main content area is titled 'Anonimizācijas uzdevumi' and features a '+ Jauns uzdevums' button and a refresh icon. Below is a table of tasks:

Timestamp	Task Name	Status
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 kopa 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 “5.7.1 Datu anonimizācija izmantojot datu devēja savienotāju”;
- **Lauku saraksts mērķa kopas ieraktu atslēgai.** Saraksts ar datu laukiem, kas veido mērķa datu kopas ierakstu primāro atslēgu.

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 patērētājiem bez VIRSIS atļaujas.

6 Pielikums – papildus paraugi

6.1 Datu kopu definīcijas

Šajā sadaļā iekļauti datu kopu definīciju piemēri. Datu kopu definēšana aprakstīta sadaļā “5.2 Izplatāmo datu kopu definēšana”. Piemēriem ir tikai ilustratīva nozīme un lasīšanas brīdī tie var būt neaktuāli salīdzinājumā ar DAGR produkcijas vidi.

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	: Interesešu programmas ID
- InterestEducationProgramSubTypeCode		: Interesešu programmas
apakštīpa kods		
- InterestEducationProgramSubTypeName		: Interesešu programmas
apakštīpa 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	: interesešu izglītības grupas
ID		
- UniversalGroupName		: Interesešu iglītības grupas
nosaukums		
- UniversalGroupTypeId	int	: interesešu izglītības grupas
veida ID		
- UniversalGroupName		: interesešu 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 adrešes neatsaucas uz šo adrešes 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 (reliģiskām 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 izdarīšanas 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 izdarīšanas 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 "Saites",
- 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
ipašumtiesību izmaiņas.
  fields:
  - areaInHectares      decimal 15 3 : Zemnieku saimniecībām - zemes platība hektāros;
  - landOwnershipChanges * :
    table:
    comments            : Zemes ipašumtiesību 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 izdarīšanas 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 izdarīšanas datums
- removedOn                            date : Ieraksta izslēgšanas datums
- limitedPartnerContribution :
  table:
  comments                             : Komandīta ieguldījums (Komandītsabiedrībām).
  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 (religijskām
organizācijām).
  - registrationTermUntil                date : Reģistrācijas termiņš (religijskām
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 izdarīšanas iemesls. Klasifikatora tips
- "registration_removal_reason".
  - registeredOn               date : Ieraksta izdarīšanas 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 izdarīšanas 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 izdarīšanas 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 izdarīšanas 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 izdarīšanas 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 izdarīšanas 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".
```

6.2 Datu struktūras kartējums (connector.yaml)