



Tutorial: How to light a Beacon v0.4?

Sabela de la Torre

www.elixir-europe.org

Contents

- Reference implementation
- Components
- Database
- Compilation
 - Project structure
 - Configuration files
- Deployment & testing
- Usage



Compilation – Core lib

- Clone the repository

```
git clone git@github.com:ga4gh-beacon/beacon-elixir.git
```

- Switch branch

```
cd beacon-elixir
```

```
git checkout v0.4
```

- Compile core library

```
cd elixir_core
```

```
mvn clean compile jar:jar
```



Reference implementation

- Back-end: <https://github.com/ga4gh-beacon/beacon-elixir/tree/v0.4>
 - Branch v0.4 (not released yet)
 - Code
 - How-to guide
 - Docker images for both back-end and front-end (API v0.3)
- Front-end: <https://github.com/ga4gh-beacon/beacon-ui-elixir>
 - Code
 - How-to guide
 - API v0.3



Components

- Data persistence
 - PostgreSQL
- Back-end
 - Java 8
 - Spring Boot v1.5.4
 - Hiberante
 - Jetty
 - Junit
 - Maven
- Front-end (v0.3)
 - Angular



Database

- Create two databases

```
psql -h localhost -p 7654 -U postgres  
  
CREATE DATABASE elixir_beacon_dev;  
  
CREATE DATABASE elixir_beacon_testing;
```

- Create a user

```
createuser -P microaccounts_dev
```

- Grant privileges

```
psql elixir_beacon_dev -U postgres  
  
GRANT ALL PRIVILEGES ON DATABASE elixir_beacon_dev TO microaccounts_dev;  
  
GRANT ALL PRIVILEGES ON DATABASE elixir_beacon_testing TO microaccounts_dev;
```



Database

- Download and load the new [schema](#)

```
wget https://raw.githubusercontent.com/elixirhub/human-data-beacon/v0.4/elixir_beacon/src/main/resources/META-INF/elixir_beacon_db_schema.sql
```

```
psql -h localhost -p 5432 -d elixir_beacon_dev -U microaccounts_dev < elixir_beacon_db_schema.sql
```

```
psql -h localhost -p 5432 -d elixir_beacon_testing -U microaccounts_dev < elixir_beacon_db_schema.sql
```

- If a superuser used, grant privileges

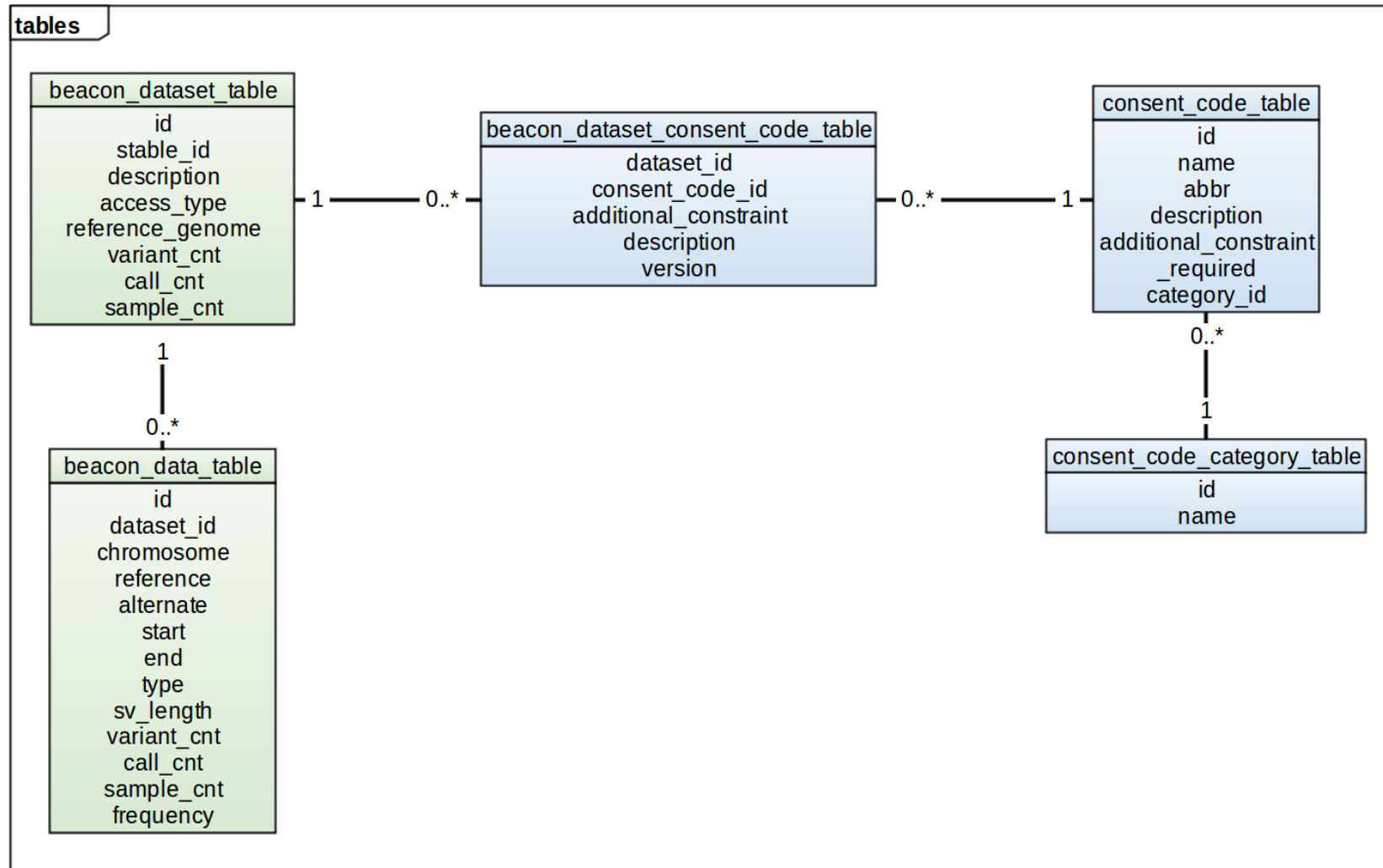
```
psql -h localhost -p 5432 -d elixir_beacon_dev -U postgres
```

```
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO microaccounts_dev;
```

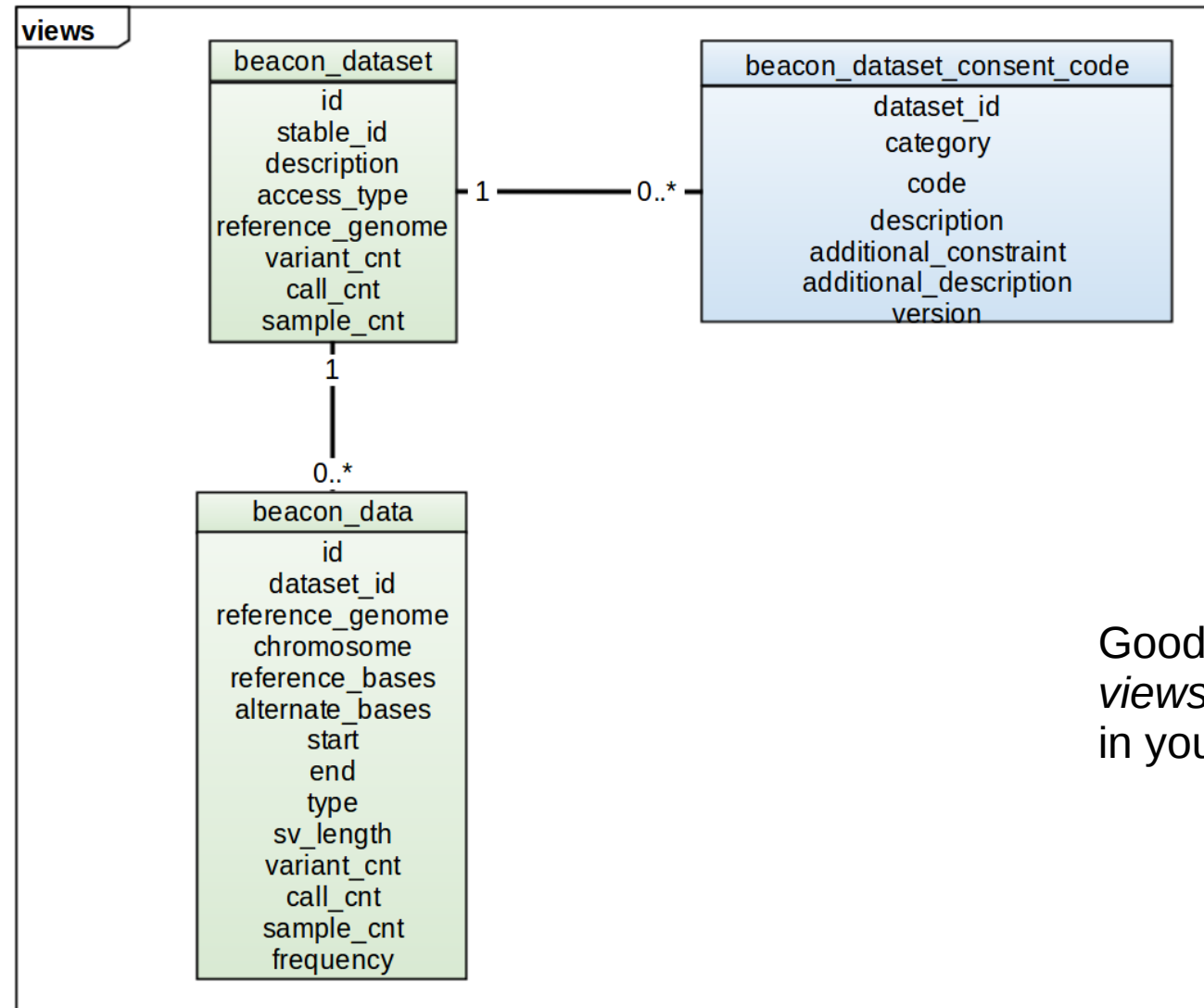
```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO microaccounts_dev;
```



Database schema



Database schema



Good practice: reference *views* instead of *tables* in your code.



Database

- Load [function](#)

```
wget https://raw.githubusercontent.com/ga4gh-beacon/beacon-elixir/v0.4/elixir_beacon/src/main/resources/META-INF/elixir_beacon_function.sql
```

```
psql -h localhost -p 5432 -d elixir_beacon_dev -U microaccounts_dev <  
elixir_beacon_function.sql
```

```
psql -h localhost -p 5432 -d elixir_beacon_testing -U microaccounts_dev <  
elixir_beacon_function.sql
```



Database

```
CREATE OR REPLACE FUNCTION public.query_data(  
  _variant_type text,  
  _start integer,  
  _start_min integer,  
  _start_max integer,  
  _end integer,  
  _end_min integer,  
  _end_max integer,  
  _chromosome character varying,  
  _reference_bases text,  
  _alternate_bases text,  
  _reference_genome text,  
  _dataset_ids text)  
  RETURNS TABLE(  
    id integer, dataset_id integer, start integer, chromosome character varying,  
    reference_bases text, alternate_bases text, "end" integer, type character varying,  
    sv_length integer, variant_cnt integer, call_cnt integer, sample_cnt integer,  
    frequency numeric, reference_genome text  
  )  
  LANGUAGE 'plpgsql'  
  ...
```



Database

- Get data
 - Use your data, or
 - Use the provided [script](#) to parse VCF files

```
wget https://raw.githubusercontent.com/elixirhub/human-data-beacon/v0.4/elixir_beacon/src/main/resources/META-INF/vcf_parser.sh
```

```
chmod +x vcf_parser.sh
```

```
./vcf_parser.sh dataset_id file.vcf
```

- **Output:** file.SNPs and file.samples



Database

file.vcf →

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SAMPLE001
1 14930 . A G 1009.29 PASS AC=1;AF=0.50;AN=2;...
...
1 1474895 . A <DEL> 1 . ...;END=1475324;...;SVLEN=428;SVTYPE=DEL;...
```

file.SNPs →

```
1;14929;1;A;G;;SNP;;1;2;;0.50
1;717484;1;C;A;;SNP;;1;2;;0.50
1;871041;1;C;CA;;SNP;;1;2;;0.50
1;880237;1;A;G;;SNP;;2;2;;1.00
...
```

file.samples → 1;SAMPLE001



```
#Output format:
dataset_id;start;chromosome;referenceBases;alternateBases;end;
structuralVariantType;svLength;variantCount;callCount;sampleCount;frequency
```



Database

- Load data
 - Init dataset information

```
INSERT INTO beacon_dataset_table(id, stable_id, description, access_type,  
reference_genome, variant_cnt, call_cnt, sample_cnt) VALUES (1, 'EGAD00000000028',  
'Sample variants', 'PUBLIC', 'grch37', 1, 1, 1);
```

- Load parsed file

```
cat file.SNPs | psql -h localhost -p 5432 -U microaccounts_dev -c "copy  
beacon_data_table  
(dataset_id,start,chromosome,reference,alternate,\"end\", \"type\",sv_length,variant_cnt  
,call_cnt,sample_cnt,frequency) FROM STDIN USING DELIMITERS ';' CSV"  
elixir_beacon_dev
```



Database

- Update counts

```
SELECT dataset_id, COUNT(*) AS variant_count  
FROM beacon_data_table  
GROUP BY dataset_id;
```

```
SELECT dataset_id, SUM(call_cnt) AS call_count  
FROM beacon_data_table  
GROUP BY dataset_id;
```

- Count samples

```
wc -l file.samples  
UPDATE beacon_dataset_table  
SET variant_cnt=47, call_cnt=80, sample_cnt=1  
WHERE id=1;
```



Compilation – Core lib

- Clone the repository

```
git clone git@github.com:ga4gh-beacon/beacon-elixir.git
```

- Switch branch

```
cd beacon-elixir
```

```
git checkout v0.4
```

- Compile core library

```
cd elixir_core
```

```
mvn clean compile jar:jar
```



Compilation – Core lib

- Install library

```
mvn install:install-file -Dfile=path_to_folder/beacon-elixir/elixir_core/target/elixir-core-beacon_api_v0.4-SNAPSHOT.jar -DgroupId=org.ega_archive -DartifactId=elixir-core -Dversion=beacon_api_v0.4-SNAPSHOT -Dpackaging=jar -DgeneratePom=true
```

- Implements useful tools:

- Filter to convert query params to lowercase (e.g. assemblyId/ASSEMBLYID > assemblyid)
- Filter to log requests params

```
2018-06-01 16:34:53,725 10626 ELXBEAC-ccc234fe-044f-454b-8773-5cbe93435b5a INFO
LogRequestFilter.java 54 User[null] URL[/elixirbeacon/v04/beacon/query]
Query[referenceName=Y&start=0&assemblyId=GRCh37] Headers[{Connection=[keep-alive], User-Agent=[Apache-HttpClient/4.2.6 (java 1.5)], Host=[localhost:9075], X-CLIENT-IP-ADDRESS=[127.0.0.1]}] Parameters[null]
```



Compilation - Beacon

- Compile and package

```
cd ../elixir_beacon
```

```
mvn clean compile package -Dspring.profiles.active="dev"
```

- Optional flag: `-Dmaven.test.skip=true`

- JAR file generated in

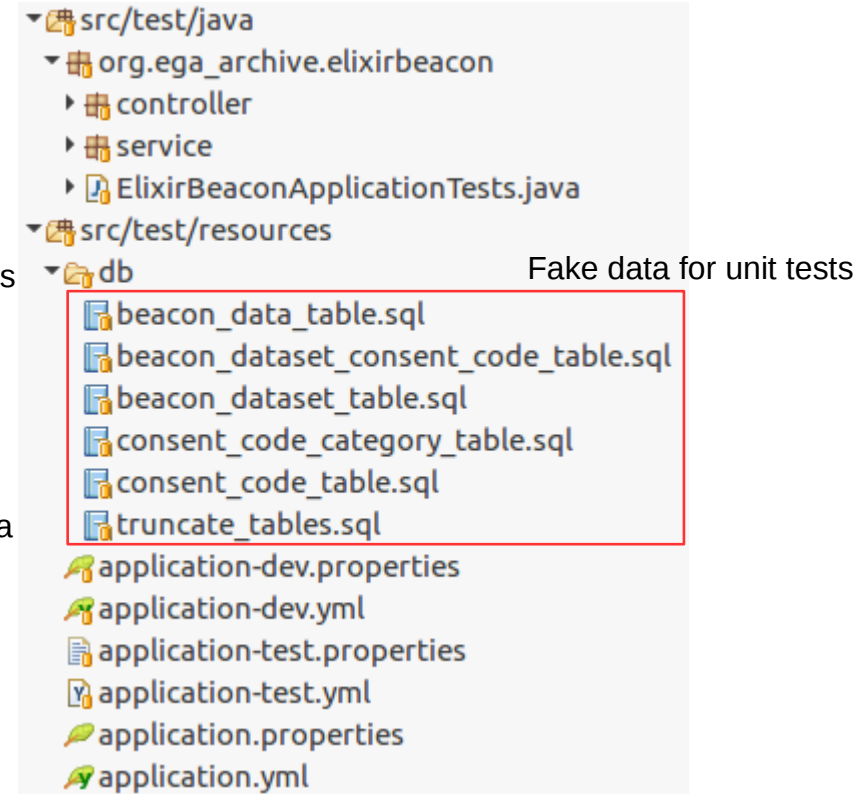
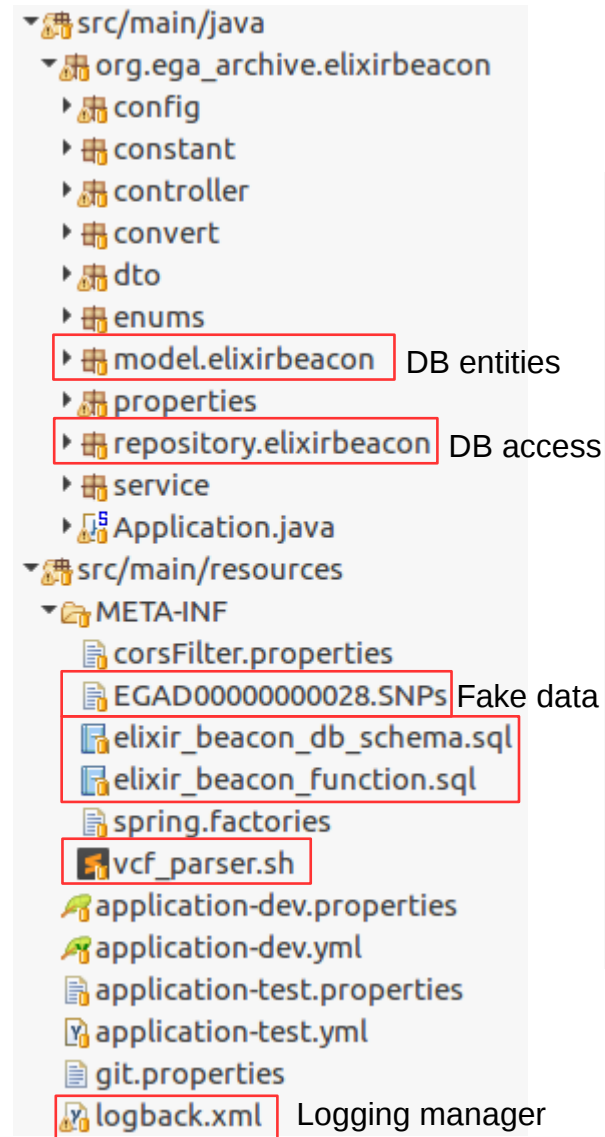
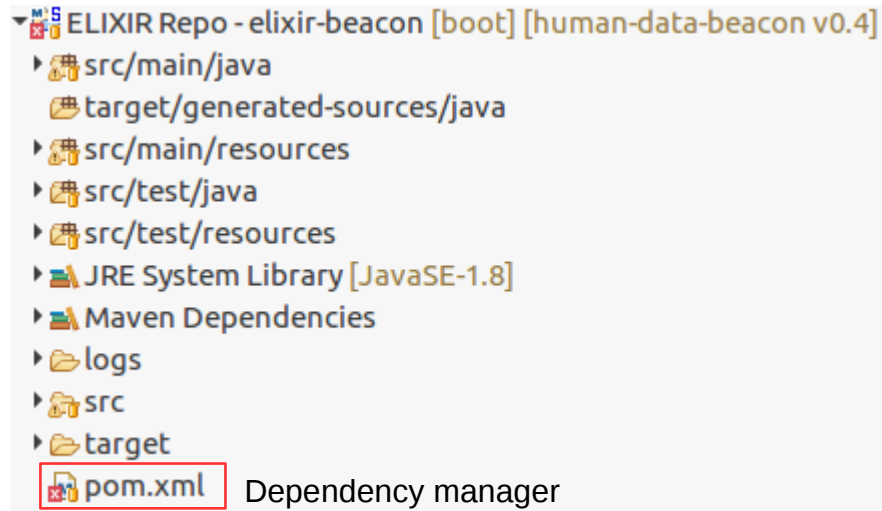
```
/target/elixir-beacon-0.4.jar
```

- Run unit tests

```
mvn test -Dspring.profiles.active="dev"
```



Project structure



Configuration files

- Location

```
/src/main/resources/application-{profile}.properties
```

```
/src/test/resources/application-{profile}.properties
```

- Deployed at: localhost:9075/elixirbeacon/v04/

```
server.port=9075
```

```
server.servlet-path=/v04
```

```
server.context-path=/elixirbeacon
```

- DB configuration

```
datasource.elixirbeacon.url=jdbc:postgresql://127.0.0.1:5432/elixir_beacon_dev
```

```
datasource.elixirbeacon.username=microaccounts_dev
```

```
datasource.elixirbeacon.password=PUT HERE YOUR PASSWORD
```

```
datasource.elixirbeacon.driverClassName=org.postgresql.Driver
```



Deployment & testing

- Deploy

```
cd elixir_beacon
```

```
java -jar target/elixir-beacon-0.4.jar --spring.profiles.active=dev
```

- Run integration tests

```
cd elixir_beacon_tests
```

```
mvn -P local clean verify
```

- Configuration of profile `local`:

```
host: localhost
```

```
port: 9075
```

- For other configurations add/modify profiles in `pom.xml`



Usage

- localhost:9075/elixirbeacon/v04/beacon/

```
{
  "id" : "elixir-demo-beacon",
  "name" : "Elixir Demo Beacon",
  "apiVersion" : "0.4",
  "organization" : {
    "id" : "EGA",
    "name" : "European Genome-Phenome Archive (EGA)",
    "description" : "The European Genome-phenome Archive (EGA)...",
    "address" : "",
    "welcomeUrl" : "https://ega-archive.org/",
    "contactUrl" : "mailto:beacon.ega@crg.eu",
    "logoUrl" : "https://ega-archive.org/images/logo.png",
    "info" : null
  },
  "description" : "This <a href=\"http://ga4gh.org/#/beacon\">Beacon</a> ...",
  "version" : "v04",
  "welcomeUrl" : "https://ega-archive.org/elixir_demo_beacon/",
  "alternativeUrl" : "https://ega-archive.org/elixir_demo_beacon_web/",
  "createDateTime" : "2015-06-01T00:00.000Z",
  "updateDateTime" : null,

```

```
"datasets" : [ {
  "id" : "EGAD000000000028",
  "name" : null,
  "description" : "Sample variants",
  "assemblyId" : "grch37",
  "createDateTime" : null,
  "updateDateTime" : null,
  "dataUseConditions" : null,
  "version" : null,
  "variantCount" : 74,
  "callCount" : 74,
  "sampleCount" : 1,
  "externalUrl" : null,
  "info" : {
    "accessType" : "PUBLIC",
    "authorized" : "true"
  }
} ],
"sampleAlleleRequests" : [ {
  "alternateBases" : null,
  "referenceBases" : "N",
  "referenceName" : "X",
  "start" : null,
  "startMin" : 153592310,
  "startMax" : 153592317,
  "end" : null,
  "endMin" : 153517030,
  "endMax" : 153517050,
  "variantType" : "DEL",
  "assemblyId" : "GRCh37",
  "datasetIds" : [ "EGAD000000000028" ],
  "includeDatasetResponses" : null
} ],
"info" : {
  "size" : "74"
}
}
```



Usage

- localhost:9075/elixirbeacon/v04/beacon/query?referenceName=1&start=16358147&referenceBases=T&alternateBases=C&assemblyId=GRCh37&includeDatasetResponses=NONE

```
{
  "beaconId" : "elixir-demo-beacon",
  "exists" : true,
  "error" : null,
  "alleleRequest" : {
    "alternateBases" : "C",
    "referenceBases" : "T",
    "referenceName" : "1",
    "start" : 16358147,
    "startMin" : null,
    "startMax" : null,
    "end" : null,
    "endMin" : null,
    "endMax" : null,
    "variantType" : null,
    "assemblyId" : "GRCh37",
    "datasetIds" : null,
    "includeDatasetResponses" : "NONE"
  },
  "apiVersion" : "0.4",
  "datasetAlleleResponses" : null
}
```



Usage

- localhost:9075/elixirbeacon/v04/beatron/query?referenceName=1&start=13035406&end=13379464&referenceBases=N&variantType=DEL&assemblyId=GRC_h37&datasetIds=EGAD00000000028&includeDatasetResponses=HIT

```
{
  "beaconId" : "elixir-demo-beacon",
  "exists" : true,
  "error" : null,
  "alleleRequest" : {
    "alternateBases" : null,
    "referenceBases" : "N",
    "referenceName" : "1",
    "start" : 13035406,
    "startMin" : null,
    "startMax" : null,
    "end" : 13379464,
    "endMin" : null,
    "endMax" : null,
    "variantType" : "DEL",
    "assemblyId" : "GRCh37",
    "datasetIds" : [ "EGAD00000000028" ],
    "includeDatasetResponses" : "HIT"
  },
  "apiVersion" : "0.4",
  "datasetAlleleResponses" : [ {
    "datasetId" : "EGAD00000000028",
    "exists" : true,
    "error" : null,
    "frequency" : 0.6,
    "variantCount" : 1,
    "callCount" : 1,
    "sampleCount" : 1,
    "note" : "OK",
    "externalUrl" : null,
    "info" : null
  } ]
}
```



Usage

- localhost:9075/elixirbeacon/v04/beatron/query?referenceName=1&startMin=13035404&startMax=13035410&endMin=13379464&endMax=13379468&referenceBases=N&variantType=DEL&assemblyId=GRCh37&datasetIds=EGAD00000000028&includeDatasetResponses=HIT

```
{
  "beaconId" : "elixir-demo-beacon",
  "exists" : true,
  "error" : null,
  "alleleRequest" : {
    "alternateBases" : null,
    "referenceBases" : "N",
    "referenceName" : "1",
    "start" : null,
    "startMin" : 13035404,
    "startMax" : 13035410,
    "end" : null,
    "endMin" : 13379464,
    "endMax" : 13379468,
    "variantType" : "DEL",
    "assemblyId" : "GRCh37",
    "datasetIds" : [ "EGAD00000000028" ],
    "includeDatasetResponses" : "HIT"
  },
  "apiVersion" : "0.4",
  "datasetAlleleResponses" : [ {
    "datasetId" : "EGAD00000000028",
    "exists" : true,
    "error" : null,
    "frequency" : 0.6,
    "variantCount" : 1,
    "callCount" : 1,
    "sampleCount" : 1,
    "note" : "OK",
    "externalUrl" : null,
    "info" : null
  } ]
}
```





How to upgrade from v0.3 to v0.4

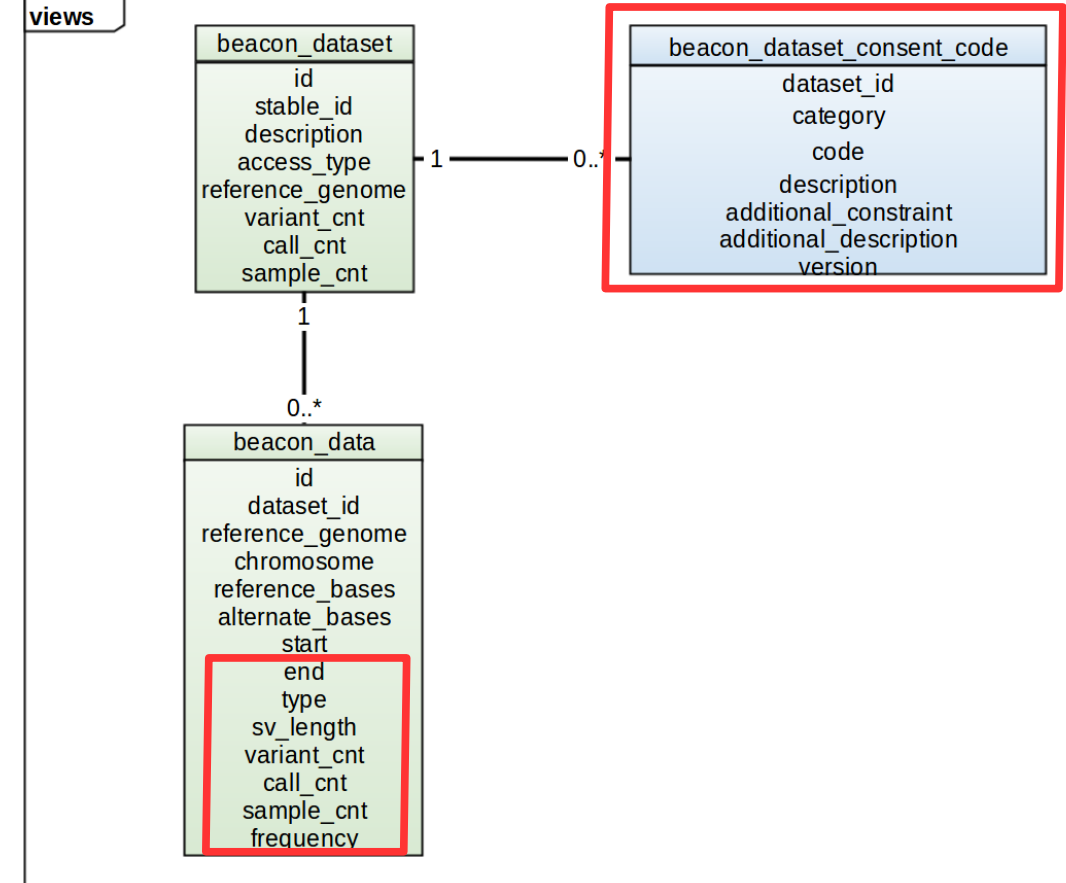
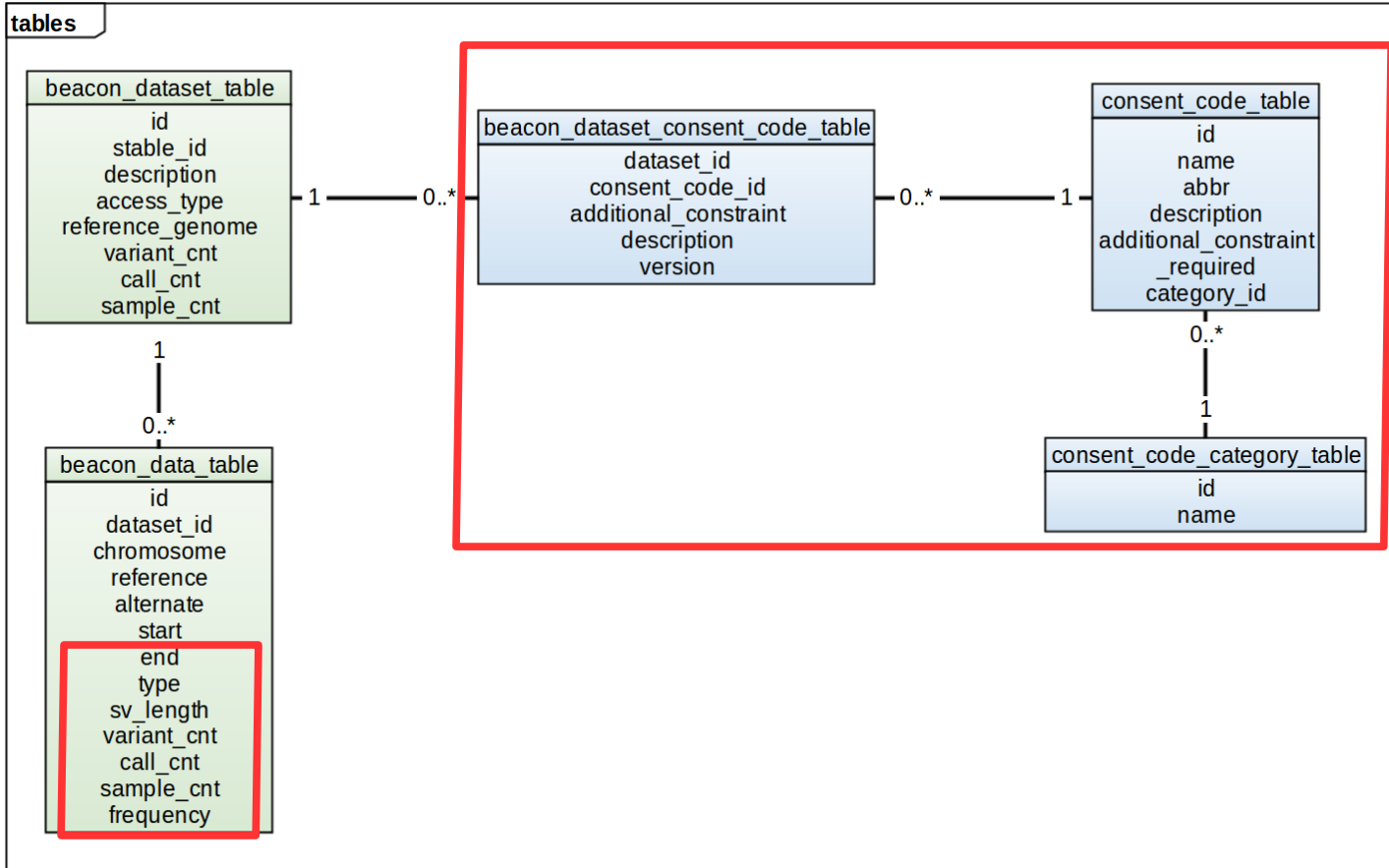
www.elixir-europe.org

Upgrading to v0.4 - Database

- Create a new DB with the new [schema](#)
 - NOTE: As it requires more information per variant, it is recommended to create a new schema instead of modifying the existing one.



Upgrading to v0.4 - Database



Upgrading to v0.4 - Database

- New [function](#)

```
CREATE OR REPLACE FUNCTION public.query_data(  
    _variant_type text,  
    _start integer,  
    _start_min integer,  
    _start_max integer,  
    _end integer,  
    _end_min integer,  
    _end_max integer,  
    _chromosome character varying,  
    _reference_bases text,  
    _alternate_bases text,  
    _reference_genome text,  
    _dataset_ids text)  
    RETURNS TABLE(  
        id integer, dataset_id integer, start integer, chromosome character varying,  
        reference_bases text, alternate_bases text, "end" integer, type character varying,  
        sv_length integer, variant_cnt integer, call_cnt integer, sample_cnt integer,  
        frequency numeric, reference_genome text  
    )  
    LANGUAGE 'plpgsql'  
    ...
```



Upgrading to v0.4 - Data

- Reload the data
 - Parse the files with the new [script](#) and reload the data, or
 - Use your own data
 - New data required:
 - end
 - structural variant type (e.g. DUP, DEL, INDEL...)
 - call count
 - variant count
 - sample count
 - frequency

#Output format:

```
dataset_id;start;chromosome;referenceBases;alternateBases;end;  
structuralVariantType;svLength;variantCount;callCount;sampleCount;frequency
```



Upgrading to v0.4 – Back-end

- Deploy the back-end:
 - Deploy the reference implementation, or
 - Adapt your own implementation:
 - New queries
 - New fields
 - Data Use Conditions
 - Changes in existing fields



Upgrading to v0.4 - Adapt your implementation

- New queries:
 - Structural variants
 - Range queries



Upgrading to v0.4 - Adapt your implementation

- New fields:
 - In BeaconAlleleRequest
 - variantType: **DUP, DEL**
 - startMin
 - startMax
 - end
 - endMin
 - endMax
 - In BeaconAlleleResponse
 - apiVersion



Upgrading to v0.4 - Adapt your implementation

- In BeaconDataset
 - dataUseConditions

See Consent Codes [specification](#)

```
{
  "dataUseConditions": {
    "consentCodedataUse": {
      "primaryCategory": {
        "code": "",
        "description": "",
        "additionalConstraint": ""
      },
      "secondaryCategories": [
        {
          "code": "",
          "description": "",
          "additionalConstraint": ""
        }
      ],
      "requirements": [
        {
          "code": "",
          "description": "",
          "additionalConstraint": ""
        }
      ]
    },
    "version": null
  }
}
```

name	abbr	description
character varying (100)	character varying (20)	character varying (400)
No restrictions	NRES	No restrictions on data use.
General research use and cl...	GRU(CC)	For health/medical/biomed...
Health/medical/biomedical ...	HMB(CC)	Use of the data is limited to ...
Disease-specific research a...	DS-[XX](CC)	Use of the data must be rel...
Population origins/ancestry ...	POA	Use of the data is limited to ...
Oher research-specific restr...	RS-[XX]	Use of the data is limited to ...
Research use only	RUO	Use of data is limited to res...
No "general methods" resea...	NMDS	Use of the data includes me...
Genetic studies only	GSO	Use of the data is limited to ...
Not-for-profit use only	NPU	Use of the data is limited to ...
Publication required	PUB	Requestor agrees to make r...
Collaboration required	COL-[XX]	Requestor must agree to col...
Ethics approval required	IRB	Requestor must provide doc...
Geographical restrictions	GS-[XX]	Use of the data is limited to ...
Publication moratorium/em...	MOR-[XX]	Requestor agrees not to pu...
Time limits on use	TS-[XX]	Use of data is approved for ...
User-specific restrictions	US	Use of data is limited to use ...
Project-specific restrictions	PS	Use of data is limited to use ...
Institution-specific restrictio...	IS	Use of data is limited to use ...



Upgrading to v0.4 - Adapt your implementation

- Changes in existing fields:
 - `includeDatasetResponses`: ALL, HIT, MISS, NONE
 - `referenceBases`: `pattern ^([ACGT]+|N)$`
 - `alternateBases`: `pattern ^([ACGT]+|N)$`





Questions?



www.elixir-europe.org

