Atlas Cohort Exchange - User Guide

# Introduction

The Atlas Cohort Exchange consists of a set of steps to allow a Cohort definition in a central Atlas instance to be distributed to remote Atlas instances, where the Cohort definition can be re-created, and the inclusion report generated for the local data set. These cohort inclusion report results can then be sent back to the central Atlas instance, where these results can be imported for review.

This document describes the steps to be performed at the external Atlas site(s); creating/importing a cohort definition from the central Atlas instance, generating the Cohort inclusion report in the local Atlas instance, and exporting these results to a JSON file for sending back to the central Atlas site.

# Prerequisites

In order to be able to import a cohort definition from the central Atlas site, a local instance of Atlas has to already be set up and configured. Please see separate documentation for the steps needed to get Atlas running at your site, if needed.

Both the cohort definition import and cohort inclusion result report export are done with an R script – you’ll therefore need to have R version 3.3 or newer installed, and preferably also a recent version of R Studio.

# Installing the AtlasCohortExchange package

In order to import the cohort definition from the central Atlas site, and to send the cohort inclusion report results back, an R package (**AtlasCohortExchange**) is available for importing the definition and exporting the results.

Before installing the **AtlasCohortExchange** R package, there are a few other packages that need to be installed, as **AtlasCohortExchange** uses functionality from these packages.

In **R Studio**, execute the following statement:

> install.packages(c("devtools", "readr", "jsonlite", "DT", "DBI", "RPostgreSQL", "RSQLServer"))

Next, install the **AtlasCohortExchange** package itself:

> library(devtools)

> install\_github("Solventrix/AtlasCohortExchange")

That’s it for the installation; you are now ready to use **AtlasCohortExchange**.

# Creating/importing Cohort definition

For a Cohort for which the central Atlas site would like to have the inclusion report generated at your Atlas site, a Cohort definition will be distributed in a JSON format. The Cohort definition needs to be imported into your local Atlas instance in order to generate the cohort inclusion report against your database; the inclusion report results can then be exported for review at the central Atlas site.

When you receive the Cohort definition in JSON format, execute the following steps in **R Studio**:

### Load AtlasCohortExchange

> library(AtlasCohortExchange)

### Specify the connection parameters for the database connection:

In order to import the cohort definition to your local Atlas instance, the database connection parameters need to be specified. These are:

|  |  |
| --- | --- |
| dbms | "postgresql" or “sqlserver” |
| host | the ip address/host name of the database instance |
| dBname | (*for PostgreSQL database only*) the database name |
| dBsourceSchema | the OMOP CDM database schema |
| dBtargetSchema | the results (‘webapi’) database schema |
| user | the user name |
| password | the user password |

The database parameters are passed as a JSON string to the various functions, so you need to create a variable that contains this JSON string with parameter values. This can be done in two different ways – either specify the parameters on the command line, or load a JSON file with the parameters.

#### createConnectionParameters

To specify the parameters on the command line, use the ***createConnectionParameters*** function, with the relevant parameter values to match your local database setup:

> connParams<-createConnectionParameters(dbms="postgresql", host="localhost", dBname = "testx", dBsourceSchema="omopcdm", dBtargetSchema="webapi", user="*dbuser*", password="*dbpassword*")

#### getConnectionParameters

To load a JSON file with the parameters, you will first need to create a .json file with all the parameters in the following format:

[

{

"dbms":"postgresql",

"server":"localhost",

"dbname":"testx",

"user":"*dbUser*",

"password":"*dbPassword*",

"sourceschema":"omopcdm",

"targetschema":"webapi"

}

]

Then, use the ***getConnectionParameters*** function:

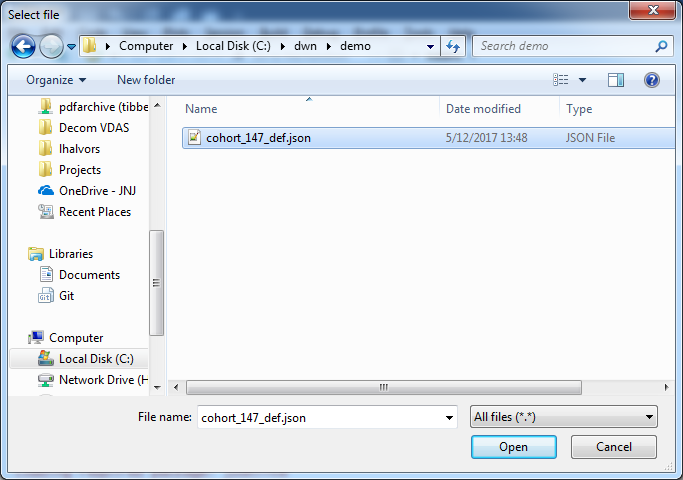
> connParams<-getConnectionParameters("C:/dwn/connect\_params.json")

### Import the cohort definition JSON file

Call the ***insertCohortDefinition*** function to import the cohort definition to your local Atlas instance:

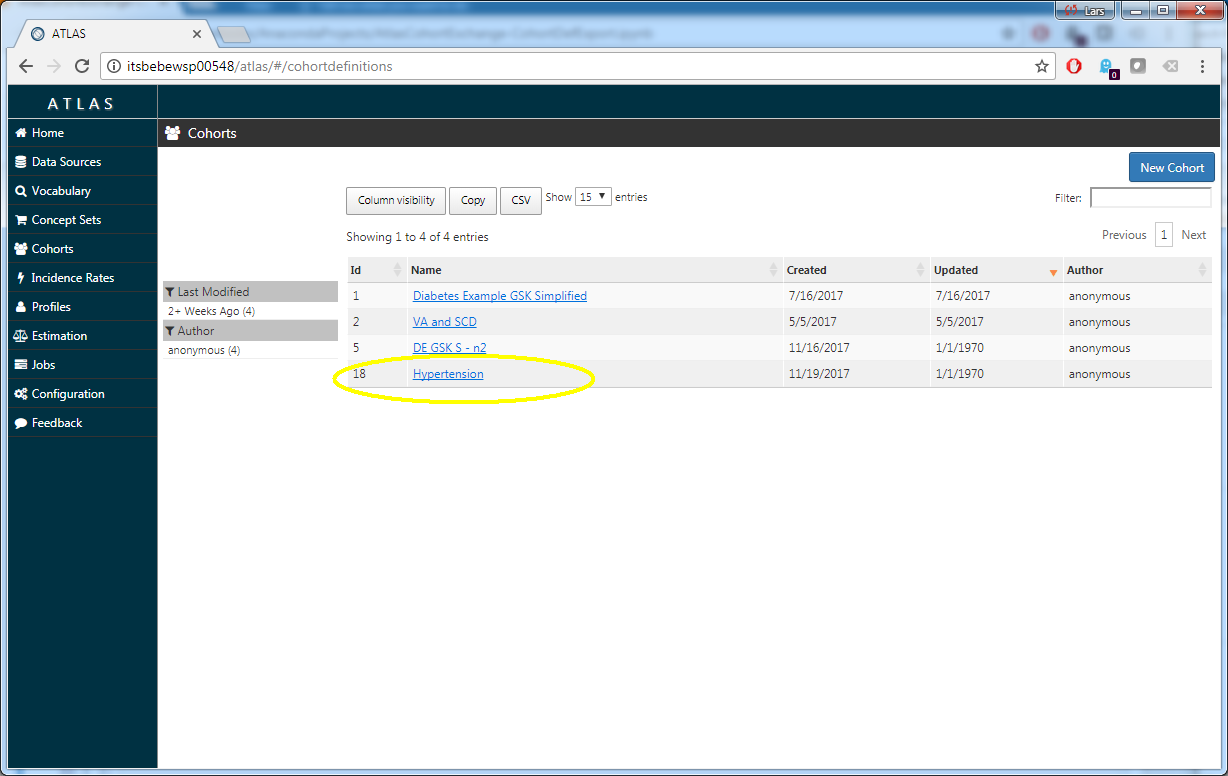
> insertCohortDefinition(connectionParameters=connParams, jsonPath="C:/dwn/cohort\_147\_def.json")

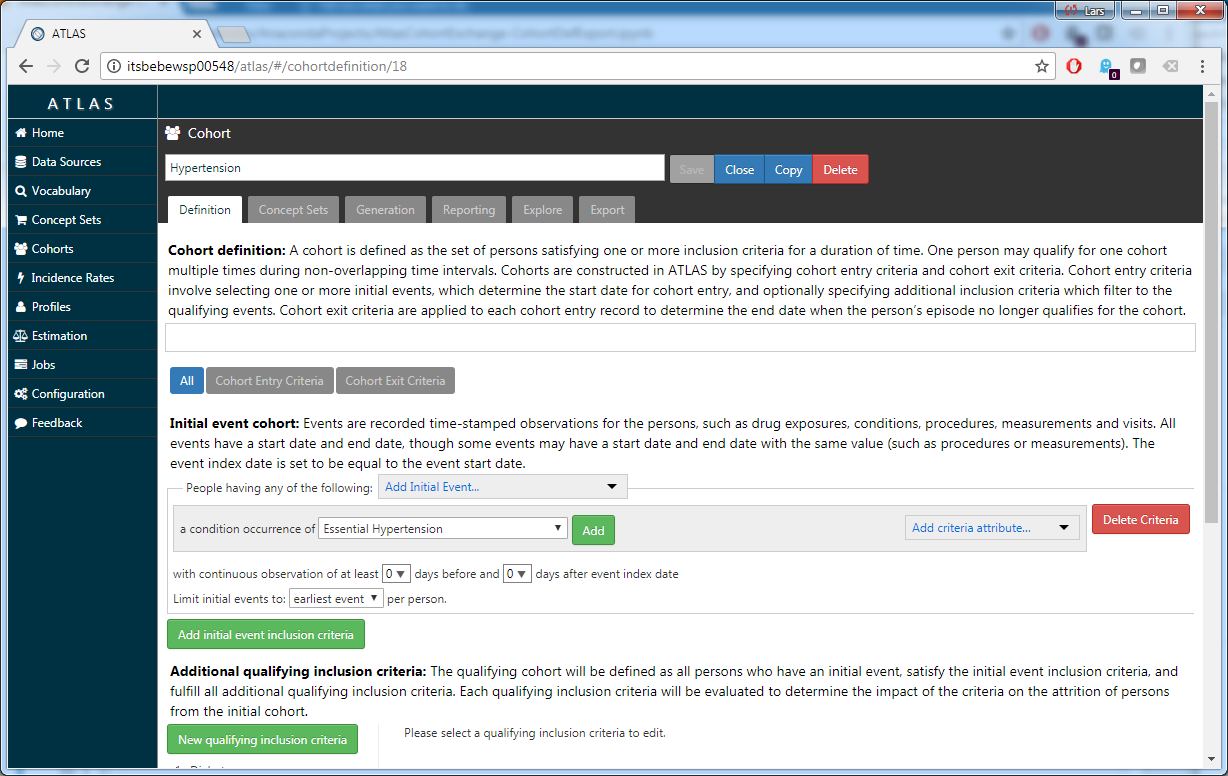
If you don’t specify a value for the jsonPath parameter, you will be prompted to select the JSON file to import:



The cohort definition will then be imported from the specified JSON file.

To verify that the definition has been loaded, go to the **Cohort** section of your local **Atlas** instance, and verify that the imported cohort definition is available:

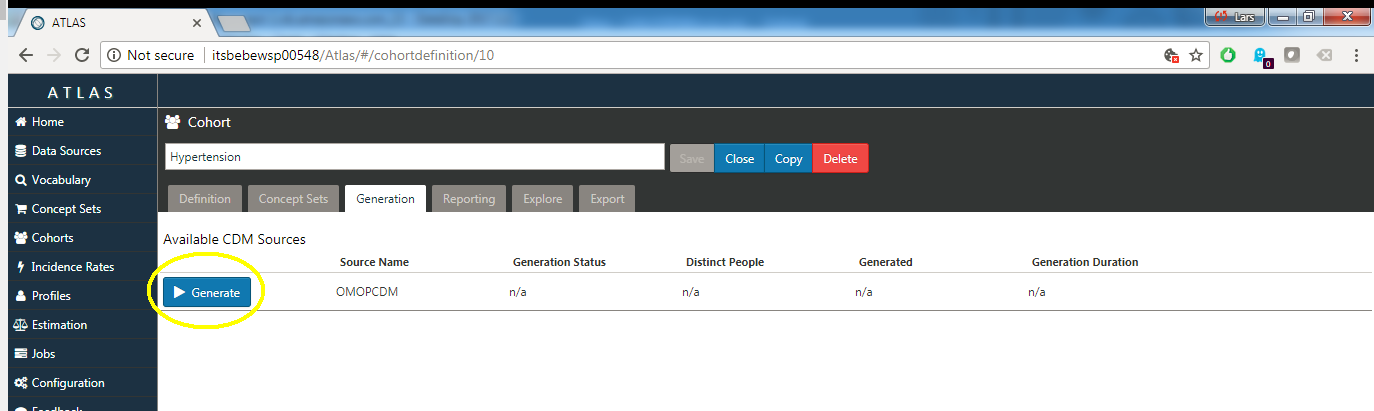




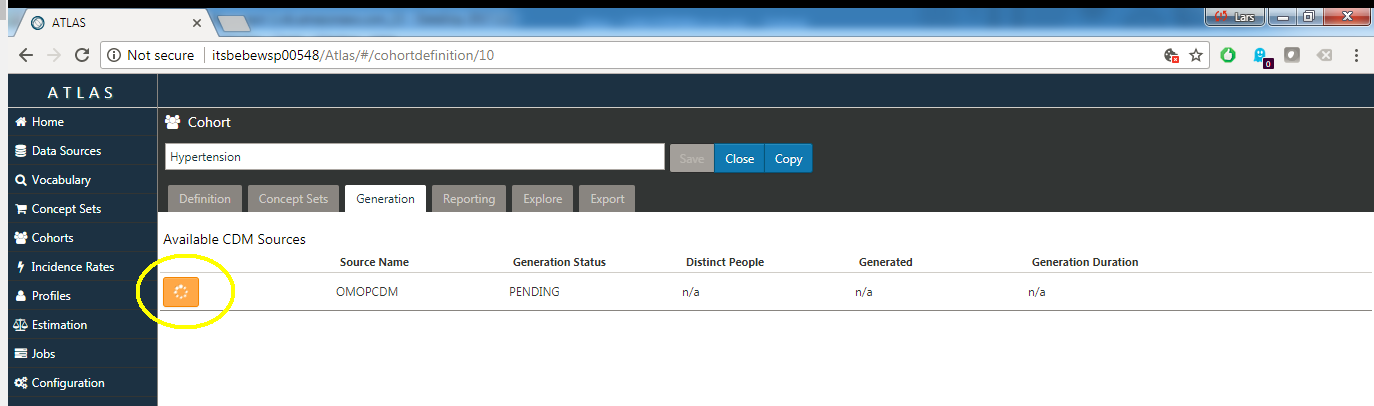
# Generating the Cohort inclusion report

Once the new Cohort definition has been created/imported, the inclusion report can be generated.

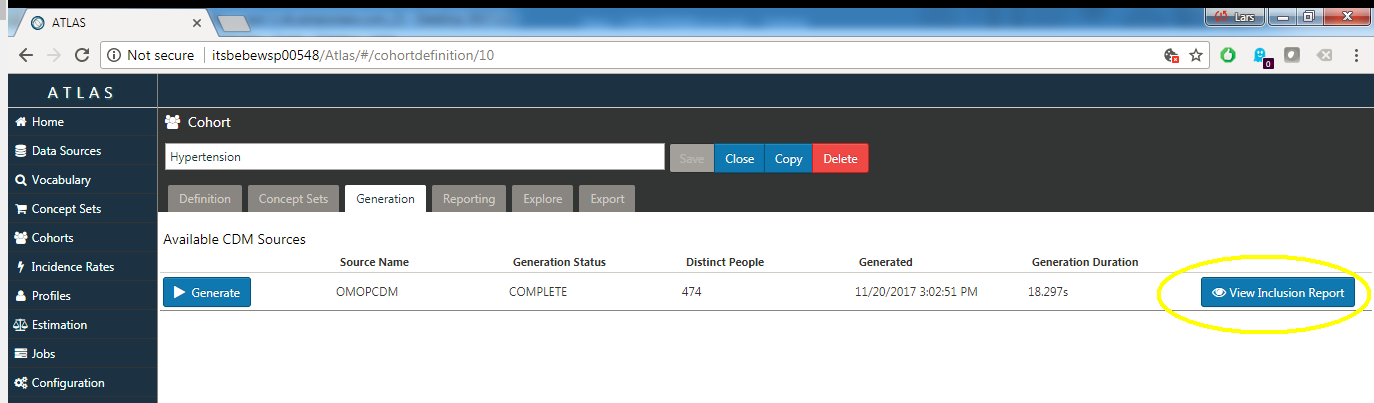
1. Open the Cohort, and go to the **Generation** tab. Click on the **Generate** button to generate the cohort inclusion report:



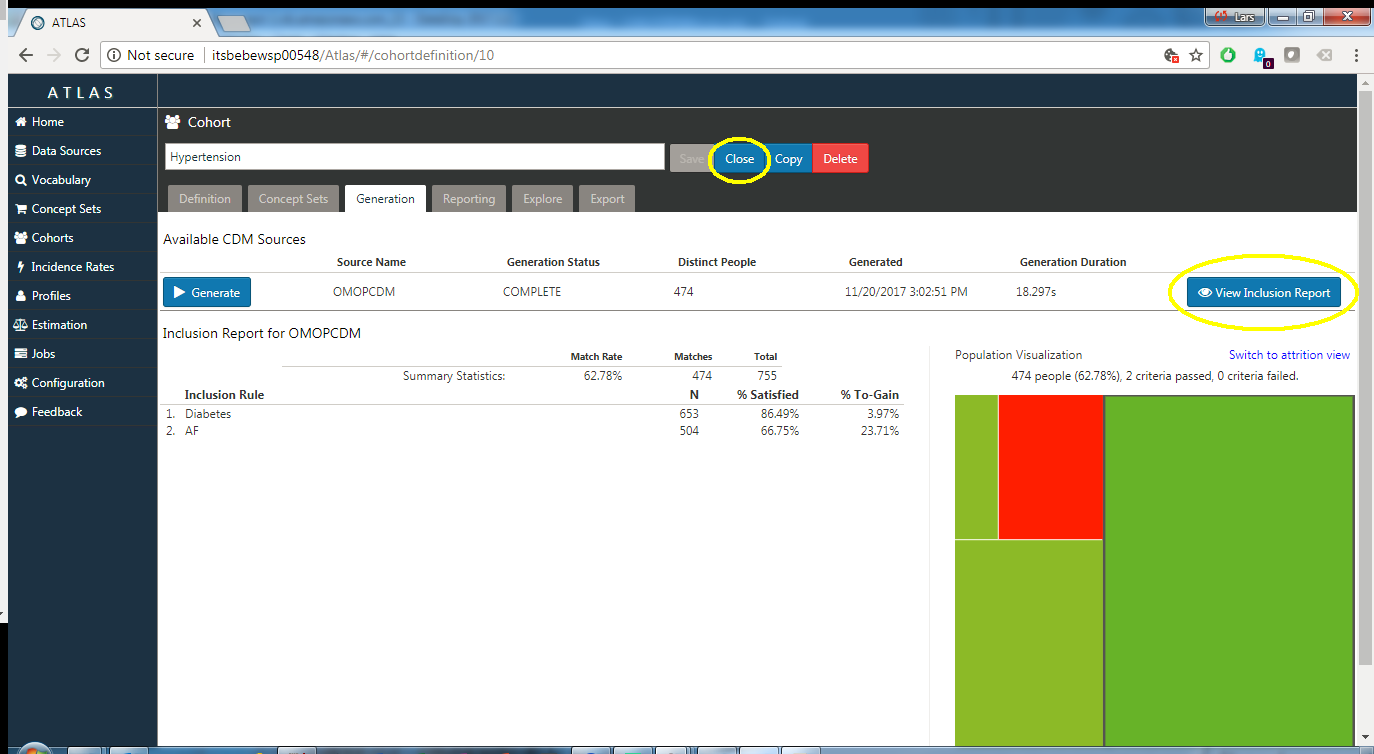
1. Depending on the complexity of the cohort definition and the size of the data set, the report generation may take a little while – the orange spinning icon in place of the **Generate** button indicates that the report creation is in progress:



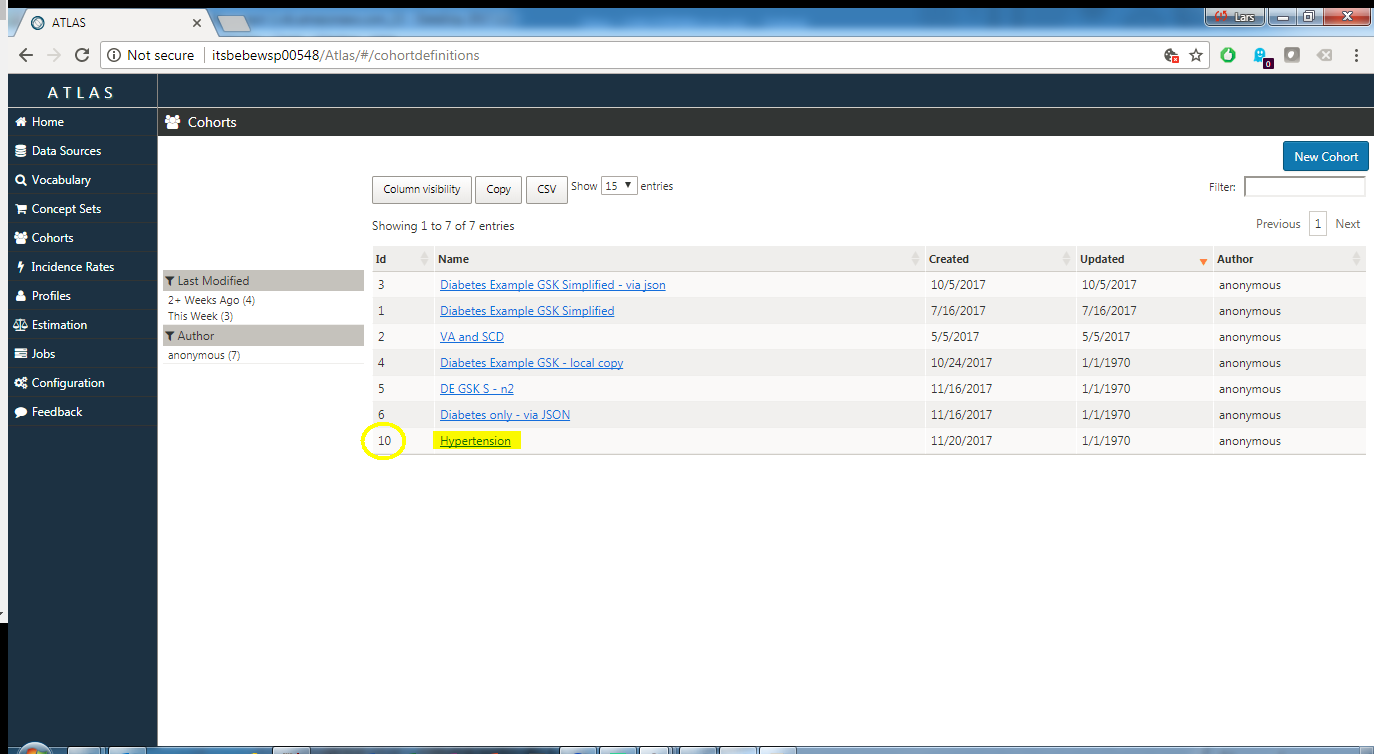
1. Once the report generation has finished, the row is updated with status information, and a new button (**View Inclusion Report**) is displayed:



1. Clicking the **View Inclusion Report** button will display the cohort inclusion report results:



1. When you close the cohort definition, please make a note of the **ID** number of this cohort (*10 in this example*), as you will need this ID when you next go to export the results in R:



# Exporting cohort inclusion report results to JSON file

The cohort inclusion report results are exported using the **AtlasCohortExchange** R package. *See section 3 above for installation instructions.*

In **R Studio**, execute the following steps:

### Load AtlasCohortExchange

> library(AtlasCohortExchange)

### Specify the connection parameters for the database connection:

In order to export the cohort inclusion report results from your local Atlas instance, the database connection parameters need to be specified. These are:

|  |  |
| --- | --- |
| dbms | "postgresql" or “sqlserver” |
| host | the ip address/host name of the database instance |
| dBname | (*for PostgreSQL database only*) the database name |
| dBsourceSchema | the OMOP CDM database schema |
| dBtargetSchema | the results (‘webapi’) database schema |
| user | the user name |
| password | the user password |

The database parameters are passed as a JSON string to the various functions, so you need to create a variable that contains this JSON string with parameter values. This can be done in two different ways – either specify the parameters on the command line, or load a JSON file with the parameters.

#### createConnectionParameters

To specify the parameters on the command line, use the ***createConnectionParameters*** function, with the relevant parameter values to match your local database setup:

> connParams<-createConnectionParameters(dbms="postgresql", host="localhost", dBname = "testx", dBsourceSchema="omopcdm", dBtargetSchema="webapi", user="*dbuser*", password="*dbpassword*")

#### getConnectionParameters

To load a JSON file with the parameters, you will first need to create a .json file with all the parameters in the following format:

[

{

"dbms":"postgresql",

"server":"localhost",

"dbname":"testx",

"user":"*dbUser*",

"password":"*dbPassword*",

"sourceschema":"omopcdm",

"targetschema":"webapi"

}

]

Then, use the ***getConnectionParameters*** function:

> connParams<-getConnectionParameters("C:/dwn/connect\_params.json")

### Export the cohort inclusion report results to JSON file

Call the ***exportCohortResults*** function to export the cohort inclusion report results from your local Atlas instance:

> exportCohortResults(connectionParameters=connParams, outFilePath="C:/dwn/hypertension\_results.json")

You will be asked to provide the cohort ID for which you wish to export the cohort inclusion report results – this is the ID that was assigned to the cohort when you created/imported it above:

Please enter the cohort ID [integer]: 10

The cohort inclusion report results will then be extracted and saved to the JSON file specified.

This exported JSON file can then be sent back to the central Atlas site for review.