Table of Contents

A database application always requires metadata where single columns are defined and described. This includes information about which column is the primary key, the column´s data type, and potentially the length and if zero values can be accepted. All of this information is called metadata.

JVx automatically assumes metadata from the database. This way domain models do not have to be created, although nothing would stop us from doing so.

The user interface also needs metadata to work properly, since, for example, no letters can be entered in a numerical column. The metadata, therefore, has to be transmitted from the server to the client (of course, for each server object). The necessary request – and also the data transmission, of course – take some time. During productive operation, we would rather avoid this loss of performance.

For this reason, JVx has an automatic metadata cache.

This cache supports various modes:

- **Global**
  The cache works statically, per VM instance, depending on how the cache is built up (dynamically or preset). Ideally, only one request for the metadata of all server objects of the entire application is required.
- **Datasource**
  The cache is tied to a datasource; the metadata for all server objects is transmitted via a single request. Therefore, one request for metadata is required per datasource.
- **Off**
  Deactivates the metadata cache; one metadata request is required per server object.

The cache is configured using the following method:

```
RemoteDataSource.setMetaDataCacheRole(MetaDataCacheRole);
```

The **Global Cache** can be addressed using the following methods:

```
//put metadata to the global cache
RemoteDataSource.putGlobalMetaData(String, MetaData);

//get metadata from the global cache
RemoteDataSource.getGlobalMetaData(String);
```

The **DataSource Cache** can be accessed, per RemoteDataSource instance, using the following methods:

```
//put datasource metadata
dataSource.putMetaData(String, MetaData);

//get datasource metadata
dataSource.getMetaData(String);
```

Note

The metadata cache is dynamically built up at the server and updated continuously. To enable the

cache buildup, the user interface has to access the server objects after the start of the application server.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/client/gui/controlling_metadata_cache

Last update: **2020/06/08 13:44**