NoSQL for Java - 1.0

Summary: This document provides basic information about the extension providing **MongoDB, Marklogic** and **CouchDB** support for the JEE analyzer.

What's new?

Please see NoSQL for Java - 1.0 - Release Notes for more information.

Description

The **NoSQL for Java** provides support for MongoDB, Marklogic and CouchDB for the JEE analyzer.

In what situation should you install this extension?

- If you need to analyze **MongoDB, MarkLogic and CouchDB** queries in Java client code.

Function Point, Quality and Sizing support

This extension provides the following support:

- **Function Points (transactions)**: a green tick indicates that OMG Function Point counting and Transaction Risk Index are supported
- **Quality and Sizing**: a green tick indicates that CAST can measure size and that a minimum set of Quality Rules exist

Function Points (transactions)	✓
Quality and Sizing	✓

CAST AIP compatibility

This extension is compatible with:

CAST AIP release	Supported
8.3.x	✓
8.2.x	✓
8.1.x	✓
8.0.x	✓
7.3.x	✗

Supported DBMS servers

This extension is compatible with the following DBMS servers:

DBMS	Supported
CSS	✓
Oracle	✓
Microsoft SQL Server	✗

Prerequisites

✓	An installation of any compatible release of CAST AIP (see table above)

Download and installation instructions

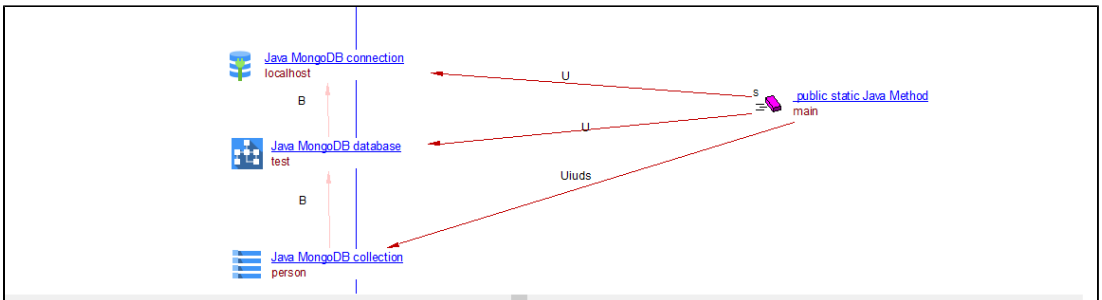
Please see:

- Download an extension
- Install an extension

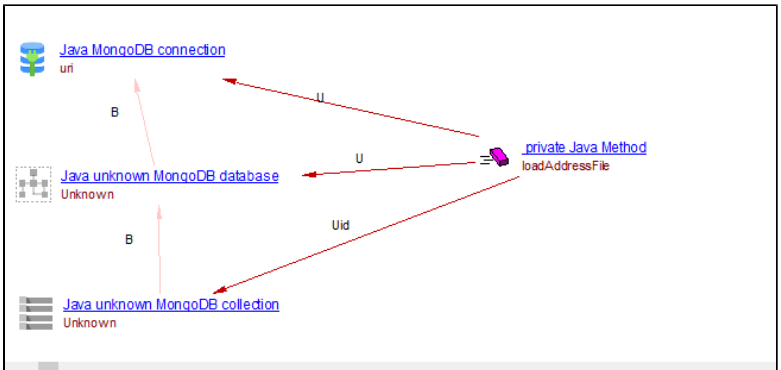
> The latest release status of this extension can be seen when downloading it from the CAST Extend server.

What results can you expect?

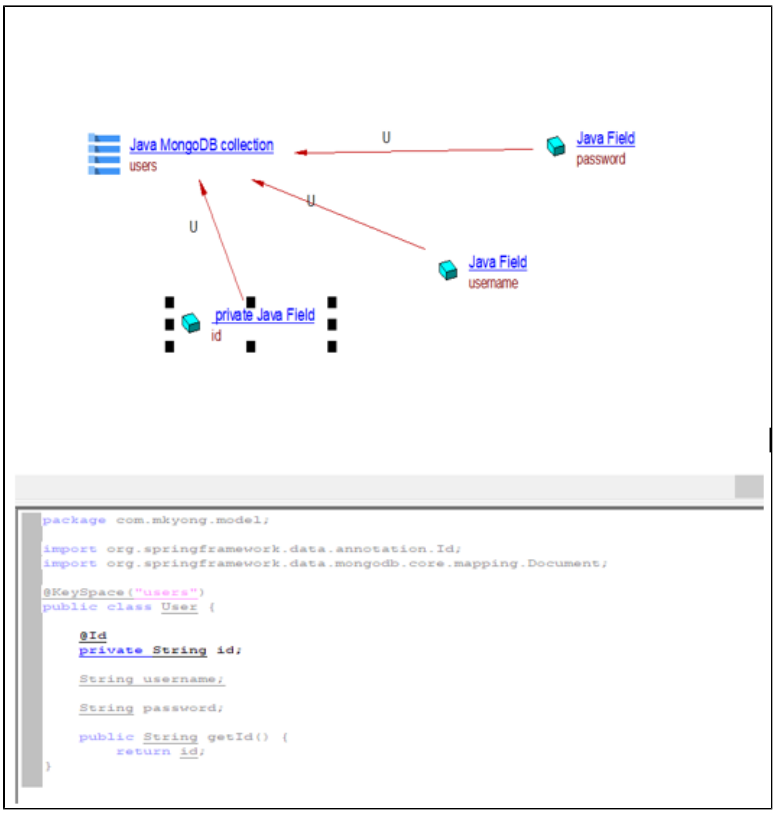
Once the analysis/snapshot generation has completed, you can view the results in the normal manner (for example via CAST Enlighten) - *click to enlarge*:



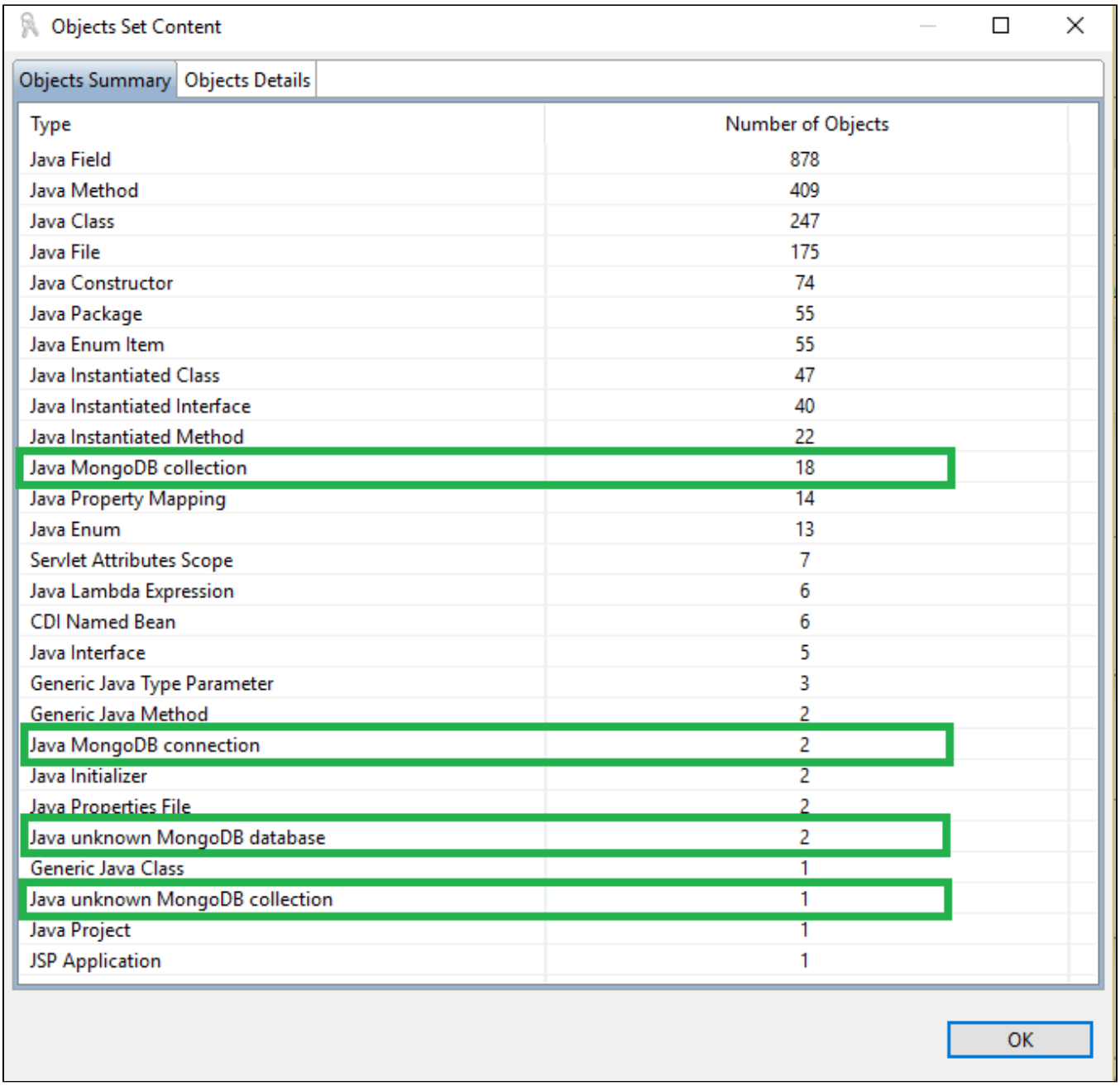

```
// get collection
// if collection doesn't exists, mongodb will create it for you
DBCollection collection = db.getCollection("person");

/**** Insert ****/
// create a document to store key and value

BasicDBObject document ;
String address[];
for(int i = 0 ; i < array_names.length ; i++){
    document = new BasicDBObject();
    //value -> String
    document.append("name", array_names[i]);
    // value -> int
    document.append("age", (int)(Math.random()*60));
    // value -> date
    document.append("join", new Date());
    // value -> array
    document.append("friends", pickFriends());

    address = pickAddress();
    // value --> document
    document.append("address", new BasicDBObject("country",address[0])
                               .append("state", address[1])
                               .append("city", address[2]));

    collection.insert(document);
    collection.findAndRemove(document);
    collection.findOne(document);
    collection.save(document);
    collection.drop(document);
    collection.distinct(document);
}
```




```
Jongo jongo = new Jongo(db);
MongoCollection collection = jongo.getCollection(collectionName);
if(1<2) {throw new Error();}
collection.remove();
```



You can also use the CAST Management Studio option **View Analysis Unit Content** to see the objects that have been created following the analysis:

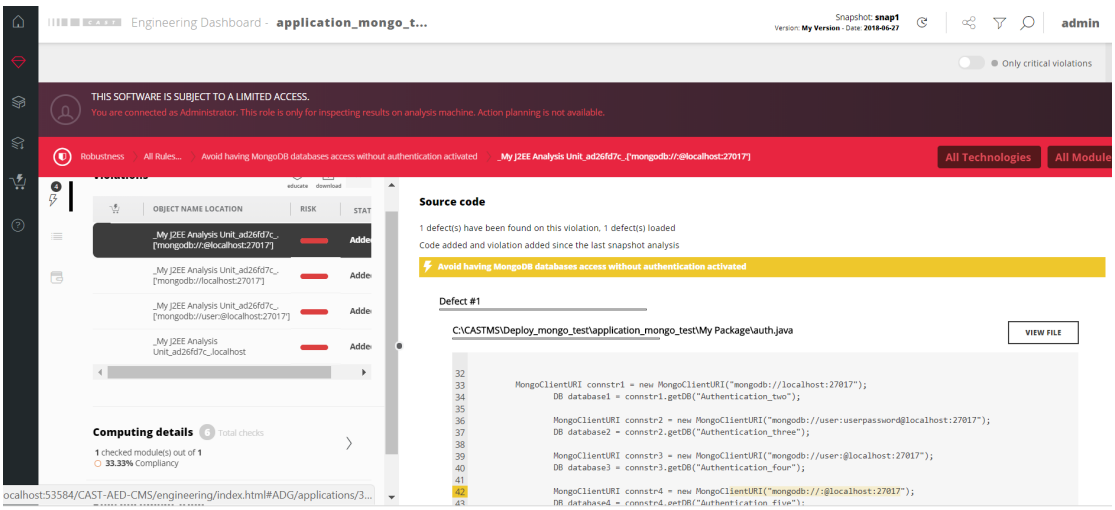


Rules

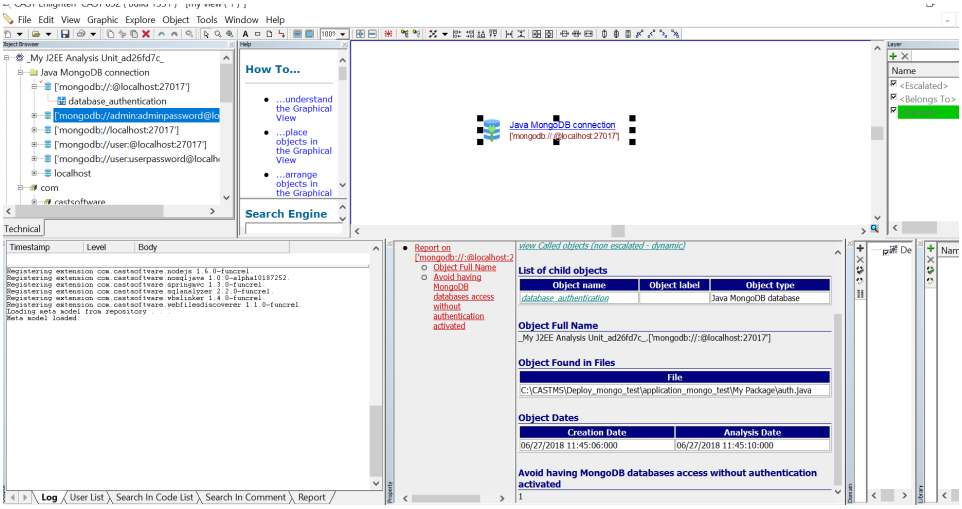
1.0.2-funcrel	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.2-funcrel
1.0.1-funcrel	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.1-funcrel
1.0.0-funcrel	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-funcrel
1.0.0-beta4	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-beta4
1.0.0-beta3	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-beta3
1.0.0-beta2	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-beta2
1.0.0-beta1	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-beta1

1.0.0-alpha3	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-alpha3
1.0.0-alpha2	https://technologies.castsoftware.com/rules?sec=srs_nosqljava&ref=||1.0.0-alpha2

Violations in the CAST Engineering Dashboard



Violations in CAST Enlighten



Objects

The following objects are displayed in CAST Enlighten:

Icon	Description
	Java MongoDB connection
	Java MongoDB database
	Java MongoDB collection
	Java unknown MongoDB database

	Java unknown MongoDB collection
	Java MarkLogic database
	Java MarkLogic collection
	Java unknown MarkLogic database
	Java unknown MarkLogic collection
	Java Couchbase connection
	Java Couchbase database
	Java Couchbase collection
	Java unknown Couchbase connection
	Java unknown Couchbase database
	Java unknown Couchbase collection

Note that:

- Objects and links are detected via parametrization and when parametization is not enough we parse the Java caller object or even the entire Java file:
  - **com.mongodb.MongoClient.MongoClient** is mapped as a MongoDB connection
  - **com.mongodb.Mongo.getDB** to MongoDB databases
  - **com.mongodb.DB.getCollection** to MongoDB collections, .update, .updateMulti, .findAndModify and .save methods are mapped as **useUpdateLinks**.
- For MongoDB we resolve Jongo queries, via parametrization and also:
  - **org.jongo.Jongo.Jongo** is mapped as a MongoDB connection
  - **org.jongo.Jongo.getCollection** to a MongoDB collection

Links

Links are created for transaction and function point needs.

MongoDB

Link type	When is this created?
parentLink	The connection is the parent of database which is the parent of a collection. Connection's parent is the caller's project.
useLink	Between the caller Java objects and connections, databases or collections.

useSelectLink useUpdateLink useDeleteLink useInsertLink	Between the caller Java object and a database or a collection.

MarkLogic

Link type	When is this created?
parentLink	The database is the parent of a collection. Database's parent is the caller's project.
useLink	Between the caller Java objects and a database or a collection.
useSelectLink useUpdateLink useDeleteLink useInsertLink	Between the caller Java object and a database or a collection.

CouchDB

Link type	When is this created?
parentLink	The connection is the parent of database which is the parent of a collection. Connection's parent is the caller's project.
useLink	Between the caller Java objects and connections, databases or collections.
useSelectLink useUpdateLink useDeleteLink useInsertLink	Between the caller Java object and a database or a collection.