NoSQL for Java - 1.0

- Description
    - In what situation should you install this extension?
- Function Point, Quality and Sizing support
- CAST AIP compatibility
- Supported DBMS servers used for CAST AIP schemas
- Prerequisites
- Download and installation instructions
- What results can you expect?
    - Rules
        - Violations in the CAST Engineering Dashboard
        - Violations in CAST Enlighten
    - Objects
    - Links
        - MongoDB
        - MarkLogic
        - Couchbase

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

What's new:

Alpha 1

New rules added:

- When using compound indexes, avoid having different indexes
- Avoid having MongoDB databases access without authentication activated

Alpha

Initial version supporting **MongoDB**, **Marklogic, Couchbase**

Description

The **NoSQL for Java** provides support for MongoDB, Marrklogic and Couchbase 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

Function Points (transactions)	Quality Rules	Security
✓	✗	✗

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 used for CAST AIP schemas

This extension is compatible with the following DBMS servers used to host CAST AIP schemas:

CAST AIP release	CSS2	Oracle	Microsoft
All supported releases	✓	✓	✓

Prerequisites

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

Download and installation instructions

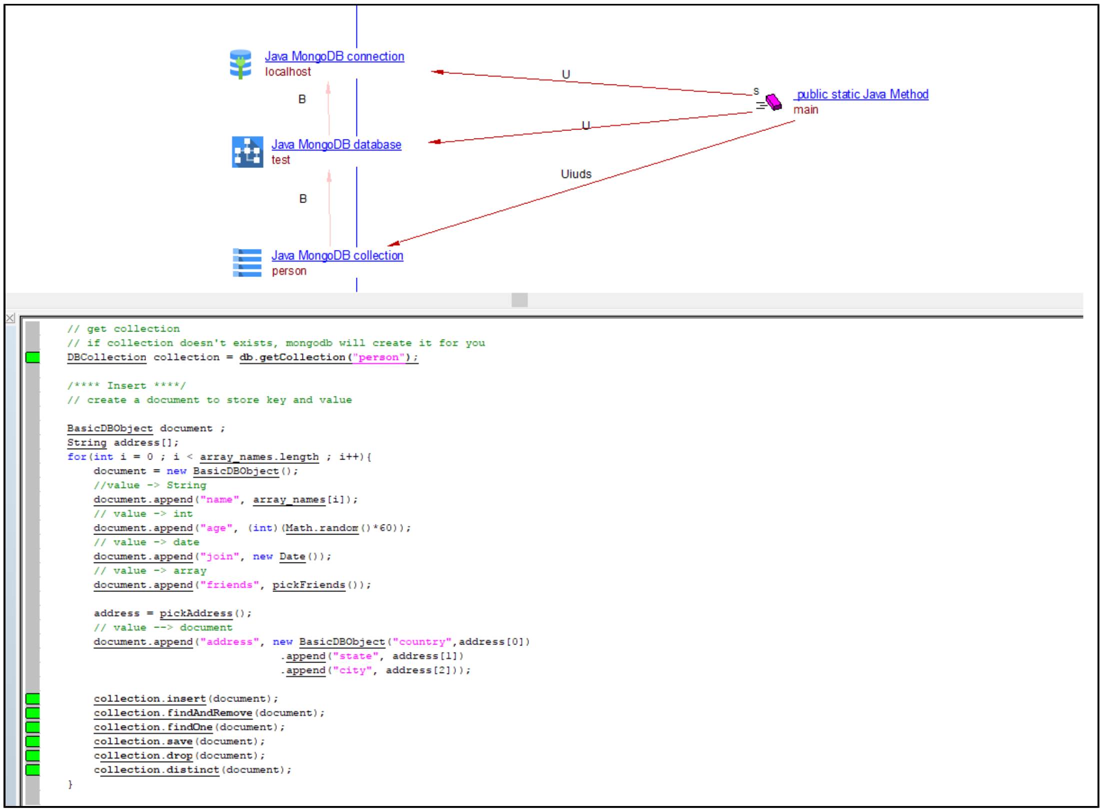
Please see:

- Download an extension
- Install an extension

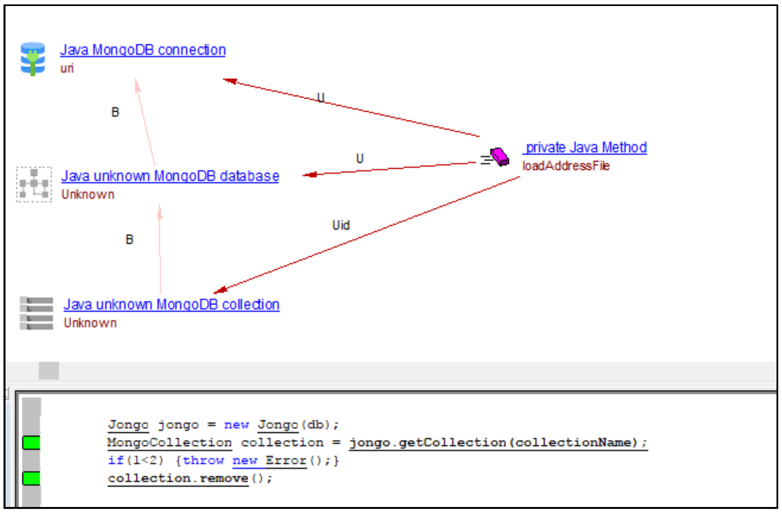
The latest release status of this extension can be seen when downloading it from the CAST Extend server.

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*:

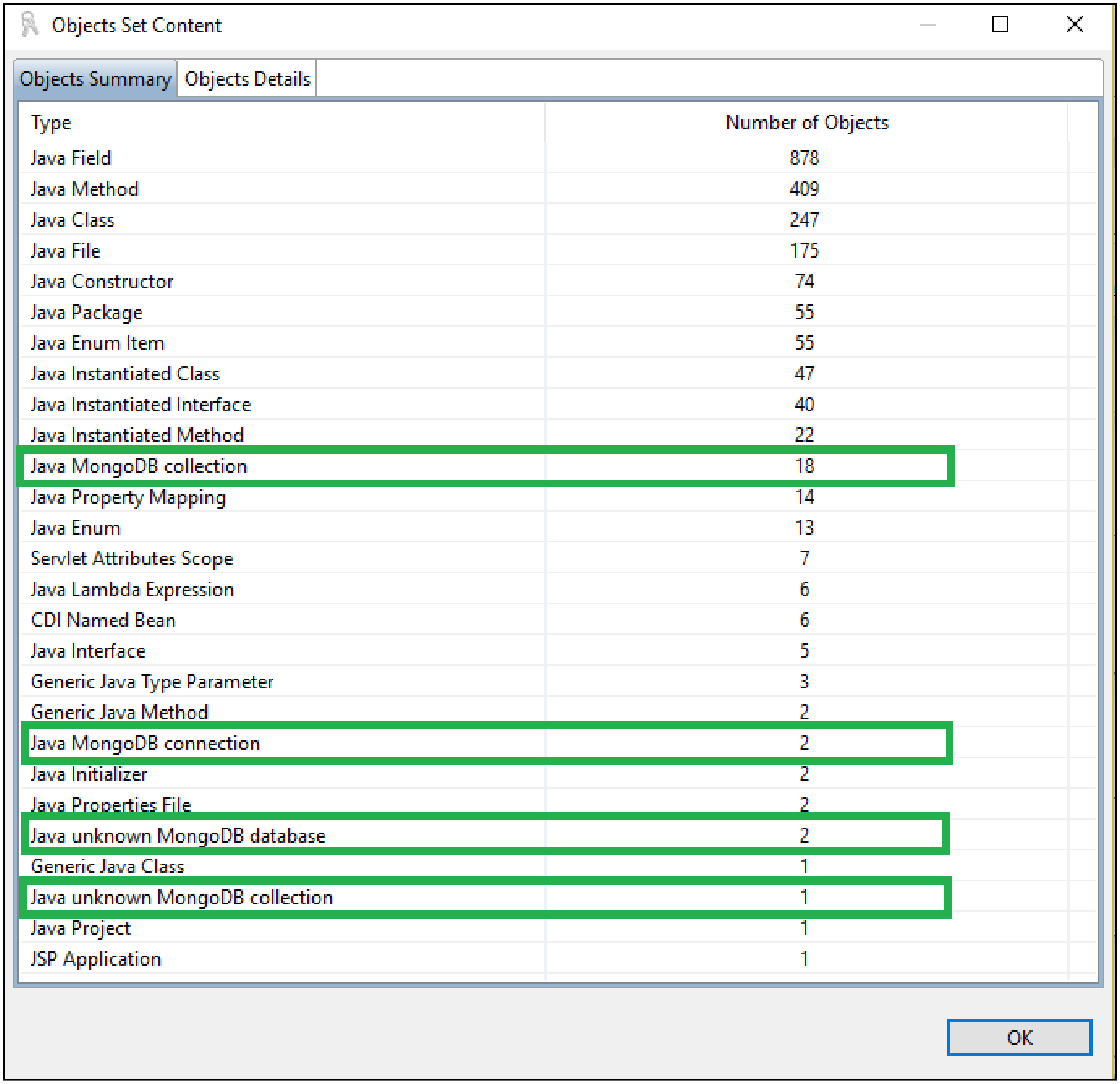


```
// get collection
// if collection doesn't exists, mongodb will create it for you
DBCollection collection = db.getCollection("person");

/**** 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:



Objects Set Content

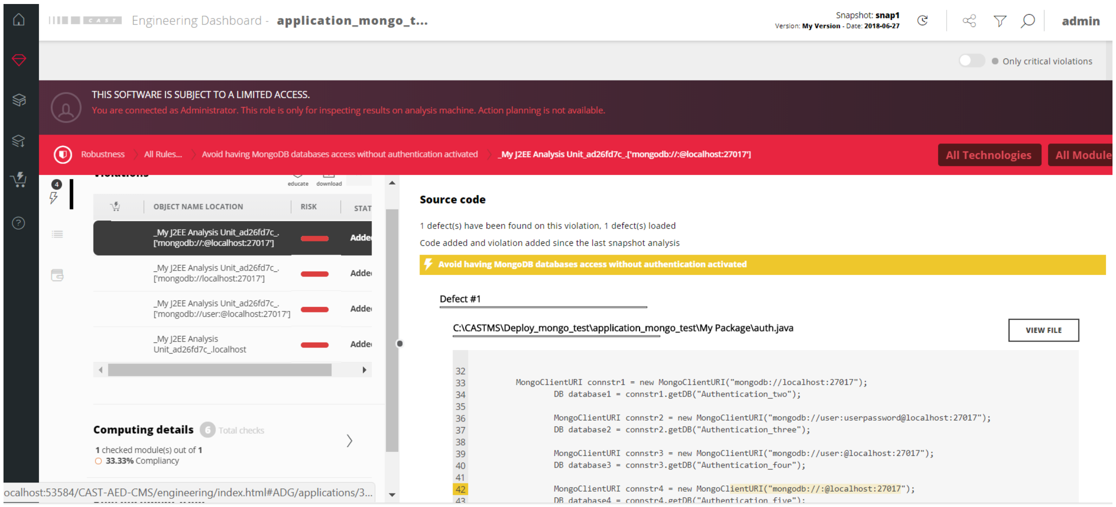
Type	Number of Objects
Java Field	878
Java Method	409
Java Class	247
Java File	175
Java Constructor	74
Java Package	55
Java Enum Item	55
Java Instantiated Class	47
Java Instantiated Interface	40
Java Instantiated Method	22
Java MongoDB collection	18
Java Property Mapping	14
Java Enum	13
Servlet Attributes Scope	7
Java Lambda Expression	6
CDI Named Bean	6
Java Interface	5
Generic Java Type Parameter	3
Generic Java Method	2
Java MongoDB connection	2
Java Initializer	2
Java Properties File	2
Java unknown MongoDB database	2
Generic Java Class	1
Java unknown MongoDB collection	1
Java Project	1
JSP Application	1

Rules

The following rules are added:

	Quality Rules
1.	When using compound indexes, avoid having different indexes
2.	Avoid having MongoDB databases access without authentication activated

Violations in the CAST Engineering Dashboard


CAST
Engineering Dashboard - application_mongo_t...
Snapshot: snap1
Version: My Version - Date: 2018-06-27
admin
Only critical violations

THIS SOFTWARE IS SUBJECT TO A LIMITED ACCESS.
You are connected as Administrator. This role is only for inspecting results on analysis machine. Action planning is not available.

Robustness
All Rules...
Avoid having MongoDB databases access without authentication activated
_My J2EE Analysis Unit_ad26fd7c_.['mongodb://:@localhost:27017']
All Technologies
All Module
4
educate
download

	OBJECT NAME LOCATION	RISK	STAT
	_My J2EE Analysis Unit_ad26fd7c_.['mongodb://:@localhost:27017']		Adde
	_My J2EE Analysis Unit_ad26fd7c_.['mongodb://localhost:27017']		Adde
	_My J2EE Analysis Unit_ad26fd7c_.['mongodb://user:@localhost:27017']		Adde
	_My J2EE Analysis Unit_ad26fd7c_.localhost		Adde

Computing details 6 Total checks

1 checked module(s) out of 1
○ 33.33% Compliancy

ocalhost:53584/CAST-AED-CMS/engineering/index.html#ADG/applications/3...

Source code

1 defect(s) have been found on this violation, 1 defect(s) loaded

Code added and violation added since the last snapshot analysis

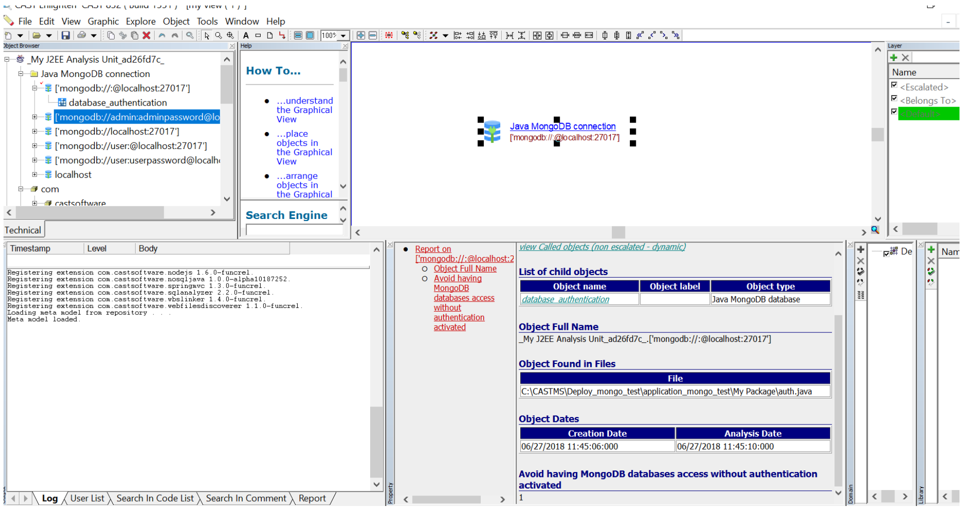
Avoid having MongoDB databases access without authentication activated

Defect #1

C:\CASTMS\Deploy_mongo_test\application_mongo_test\My Package\auth.java

```

MongoClientURI connstr1 = new MongoClientURI("mongodb://localhost:27017");
        DB database1 = connstr1.getDB("Authentication_two");

        MongoClientURI connstr2 = new MongoClientURI("mongodb://user:userpassword@localhost:27017");
        DB database2 = connstr2.getDB("Authentication_three");

        MongoClientURI connstr3 = new MongoClientURI("mongodb://user:@localhost:27017");
        DB database3 = connstr3.getDB("Authentication_four");

        MongoClientURI connstr4 = new MongoClientURI("mongodb://:@localhost:27017");
        DB database4 = connstr4.getDB("Authentication_five");
```

Violations in CAST Enlighten



- _My J2EE Analysis Unit_ad26fd7c_
  - Java MongoDB connection
    - ['mongodb://:@localhost:27017']
      - database_authentication
    - ['mongodb://admin:adminpassword@lo
    - ['mongodb://localhost:27017']
    - ['mongodb://user:@localhost:27017']
    - ['mongodb://user:userpassword@localh
    - localhost
  - com
    - castsoftware

How To...

- ...understand the Graphical View
- ...place objects in the Graphical View
- ...arrange objects in the Graphical

Search Engine

```
Registering extension com.castsoftware.nodejs 1.6.0-funcrel.
Registering extension com.castsoftware.nosqljava 1.0.0-alpha10187252.
Registering extension com.castsoftware.springmvc 1.3.0-funcrel.
Registering extension com.castsoftware.sqlanalyzer 2.2.0-funcrel.
Registering extension com.castsoftware.wbslinker 1.4.0-funcrel.
Registering extension com.castsoftware.webfilesdiscoverer 1.1.0-funcrel.
Loading meta model from repository . . .
Meta model loaded.
```

- Report on ['mongodb://:@localhost:2
  - Object Full Name
  - Avoid having MongoDB databases access without authentication activated

view Called objects (non escalated - dynamic)

List of child objects

Object name	Object label	Object type
database_authentication		Java MongoDB database

Object Full Name

_My J2EE Analysis Unit_ad26fd7c_.['mongodb://:@localhost:27017']

Object Found in Files

File
C:\CASTMS\Deploy_mongo_test\application_mongo_test\My Package\auth.java

Object Dates

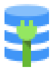
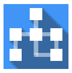
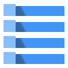
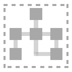
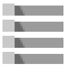
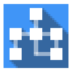
Creation Date	Analysis Date
06/27/2018 11:45:06:000	06/27/2018 11:45:10:000

Avoid having MongoDB databases access without authentication activated

1

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.

Couchbase

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.