**Hygieia documentation**

**Java Installation**

1. Download Java JDK from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. In windows environment variables. Create a JAVA\_HOME variable and specify the path to the java folder. Change PATH variable to add value;%JAVA\_HOME%\bin;

**Maven Installation**

1. Download apache\_maven from <http://maven.apache.org/download.cgi>
2. Add the MAVEN\_HOME and M2\_HOME environment variables with the path to the folder with apache\_maven. Change the path% M2\_HOME% \ bin variable

**MongoDB Installation**

1. 1) Download MongoDB https://www.mongodb.com/download-center?jmp=nav#community

You can download the archive.

1. Install to С:\mongodb\
2. Add path to environment variable С:\mongodb\bin
3. Create a data folder in a directory. It is necessary to store data from the database. Inside create db and log folders. After changing the settings mongod.cfg, which is mongodb / bin / mongod.cfg

dbpath=C:\data\db

logpath=C:\data\log\mongo.log

1. Open a command prompt on behalf of the administrator and write

«Mongod»

“I NETWORK [initandlisten] waiting for connections on port 27017”

After this message, open another command prompt on behalf of the administrator. Write «mongo». The presence of the arrow> says that you can write commands. Select a database, enter use dashboarddb. After we enter:

We receive information that the user was able to create.

**Node.Js Installation**

1. Install Git.
2. Download Node.js from https://nodejs.org/en/download/ Install. Add environment variables. Path to the Node.js folder
3. Open the command line and install bower and gulp.

npm install -g bower

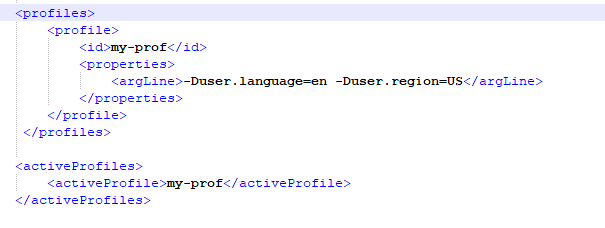
npm install -g gulp

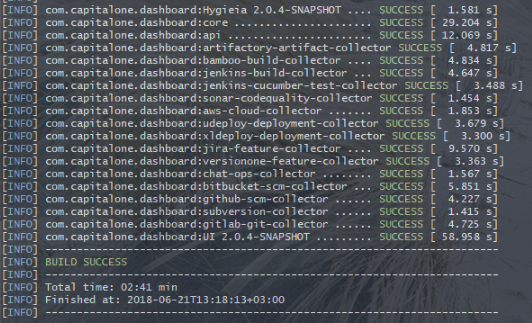
**Hygieia Installation**

1. Download Hygieia с <https://github.com/capitalone/Hygieia.git>
2. Unpack to disk.
3. Open the command line go inside the Hygieia folder. Write:

mvn clean install package

* If you have Russian windows! The version of Hygieia c master is not going to. Does not pass the tests. Tests do not pass because of the locale. In the USA, the fractional part is separated from the whole “,” we “.” “Here either change the locale system. Or change the maven settings. Go to MAVEN\_HOME {path} /settings.xml



After installation should show this:

1. Run mongod, after mongo
2. Create a dashboard.properties file in the .. \ api folder with the data:dbname=dashboarddb

#dbusername=dashboarduser

#dbpassword=dbpassword

dbhost=localhost

dbport=27017

1. Edit the file \ api \ src \ main \ resources \ application.properties

# Default server configuration values

dbname=dashboarddb

#dbusername=dashboarduser

#dbpassword=dbpassword

dbhost=localhost

dbport=27017

server.contextPath=/api

server.port=8080

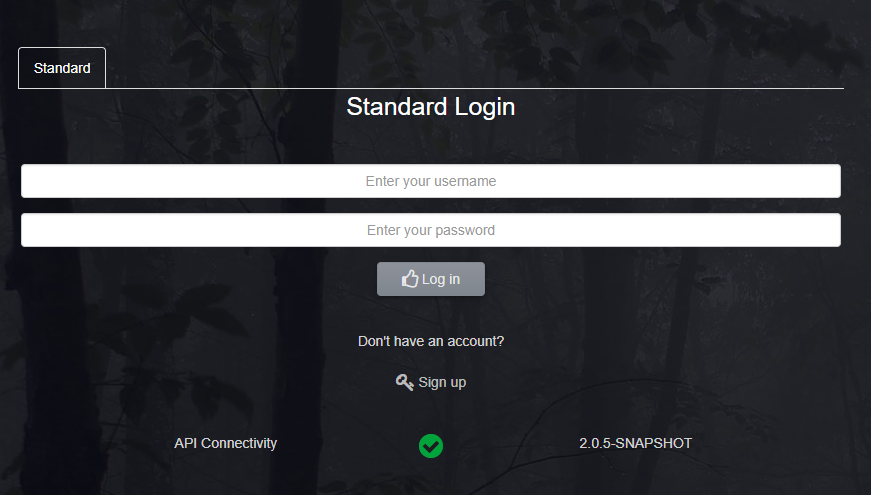
1. In the Command Line, open api \ target and launch api with the command

java -jar api.jar --spring.config.location=C:\[path to]\Hygieia\api\dashboard.properties -Djasypt.encryptor.password=hygieiasecret

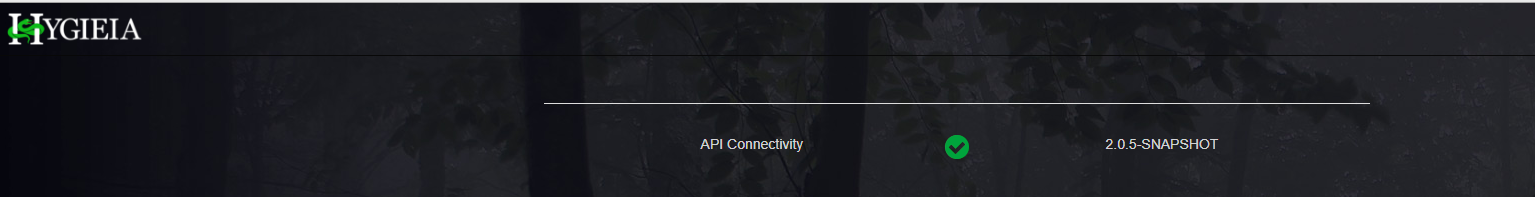
\* Change the path if it is different.

Log says that api is running.

2018-07-26 15:25:45,082 INFO com.capitalone.dashboard.Application - Started Application in 22.585 seconds (JVM running for 24.17)

1. Open another command line, go to the Hygieia \ UI folder and start the UI with the gulp serve command
2. Open the localhost: 3000 page in the browser.
3. Create a user. A tick API Connectivity says that the API is running and working properly. If there is a red cross, then most likely the API is not running or does not work correctly.

\* If at start Exception MongoTimeoutException gets out: Timed out after 30000 ms. Then there is a problem with the database, or mongod is not running, or the url is not specified correctly, you can try 127.0.0.1 instead of localhost. Perhaps the user dashboarduser was not created.

\* Sometimes when you run the Hygieia page. The login page can have this form.

To fix this you need to clear your browser history.

**Enable collectors**

1. First you need to download my development from <https://github.com/matjukov-nikolaj>
2. Download library which called codesecurity from <https://github.com/matjukov-nikolaj/codesecurity.git>
3. Move it to the Hygieia / folder. Go inside and install: mvn clean install
4. Download Hygieia-utilites from <https://github.com/matjukov-nikolaj/Hygieia-utilites.git>
5. Move files from api to directories specified in the dashboard. Also replace the pom.xml file with a file from Hygieia-utilites \ api
6. Build Api without tests Hygeia \ api -> mvn clean install –DskipTests
7. Move files from Hygieia-utilites \ core to Hygieia \ core in the same directories.
8. Build the Hygieia \ core. Write: mvn clean install kernel
9. Move files from Hygieia-utilites \ UI to directories UI.

**Sonar**

1. First you need to turn off api and UI. To run the sonar collector you need to go to the collectors / build / sonar directory
2. Create an application.properties file

Write the following settings:

# Database Name

dbname=dashboarddb

# Database HostName - default is localhost

dbhost=127.0.0.1

# Database Port - default is 27017

dbport=27017

# Database Username - default is blank

dbusername=dashboarduser

# Database Password - default is blank

dbpassword=dbpassword

# Collector schedule (required).

sonar.cron=0 0/1 \* \* \* \*

# Sonar server(s) (required) - Can provide multiple.

sonar.servers[0]=http://localhost:9000

# Sonar version, match array index to the server. If not set, will default to version prior to 6.3.

sonar.versions[0]=7.2

# Sonar Metrics - Required.

# Sonar versions lesser than 6.3

# sonar.metrics[0]=ncloc,line\_coverage,violations,critical\_violations,major\_violations,blocker\_violations,violations\_density,sqale\_index,test\_success\_density,test\_failures,test\_errors,tests

# For Sonar version 6.3 and above

sonar.metrics[0]=ncloc,violations,new\_vulnerabilities,critical\_violations,major\_violations,blocker\_violations,tests,test\_success\_density,test\_errors,test\_failures,coverage,line\_coverage,sqale\_index,alert\_status,quality\_gate\_details

# Sonar login credentials.

sonar.username=admin

sonar.password=admin

1. Start the sonar from the target folder

java -jar sonar-codequality-collector-2.0.5-SNAPSHOT.jar --spring.config.name=sonar --spring.config.location=С:\{path to}\Hygieia\collectors\build\sonar\application.properties

4) The application will start and search for sonar projects. It will be launched after the time specified in cron.

5) There is a problem when the assembly of the kernel does not create a collection of code\_quality. This can be viewed in mongo. Use dashboarddb-> show tables. If there is no code\_quality among them, then you need to create this collection: db.createCollection (“code\_quality”); After copying the file from Hygieia-utilites / settings / sonar / SonarCollectorTask to the sonar / src / ... / collector / folder and reinstalling mvn clean install;

6) To check the performance of the collector, it must put the data in the code\_quality collection: db.code\_quality.find ()

\* Perhaps the problem with the lack of data has already been solved by the developers of Hygieia

**Jenkins codequality**

1. The essence of the launch is the same as in Sonar.
2. Configure the application.properies file
3. *# Database Name*
4. dbname=dashboarddb
5. *# Database HostName - default is localhost*
6. dbhost=10.0.1.1
7. *# Database Port - default is 27017*
8. dbport=27017
9. *# MongoDB replicaset*
10. dbreplicaset=[false, if you are not using MongoDB replicaset]
11. dbhostport=[host1:port1,host2:port2,host3:port3]
12. *# Database Username - default is blank*
13. dbusername=dashboarduser
14. *# Database Password - default is blank*
15. dbpassword=dbpassword
16. *# Collector schedule (required)*
17. jenkins-codequality.cron=0 0/1 \* \* \* \*
18. *# Collector servers (required) - can be multiple servers*
19. jenkins-codequality.servers[0]=https://jenkins.company.com
20. *# Collector types (not required, but regex should match only the type specified)*
21. jenkins-codequality.artifactRegex.junit=TEST-.\*\\.xml
22. jenkins-codequality.artifactRegex.findbugs=findbugsXml.xml
23. jenkins-codequality.artifactRegex.pmd=pmd.xml
24. jenkins-codequality.artifactRegex.checkstyle=checkstyle-result.xml
25. jenkins-codequality.artifactRegex.jacoco=jacoco.xml
26. *# Collector job depth (required) should be set to at least 1, and more if you use folder jobs, and so on.*
27. jenkins-codequality.jobDepth=4
28. Install mvn clean install with build / jenkins-codequality /
29. Run with target: java -jar jenkins-codequality.jar --spring.config.name = jenkins-codequality --spring.config.location = [path to application.properties file]

**Checkmarx**

1) Create a codesecurity folder in collectors. To clone there: https://github.com/matjukov-nikolaj/checkmarx.git

2) Build mvn -Dpmd.failOnViolation = false clean install

3) In the file applicatiom.properties, specify the server with the checkmarx report. The report is in the checkmarx-current file and is updated every day.

4) Run the application: java -jar checkmarx-collector-2.0.5-SNAPSHOT.jar --spring.config.name = checkmarx --spring.config.location = D: \ {path to} \ Hygieia \ collectors \ codesecurity \ checkmarx \ application.properties

Example of properties file:

# Database Name

dbname=dashboarddb

# Database HostName - default is localhost

dbhost=127.0.0.1

# Database Port - default is 27017

dbport=27017

# Database Username - default is blank

dbusername=dashboarduser

# Database Password - default is blank

dbpassword=dbpassword

server.port=8184

# Collector schedule (required)

checkmarx.cron=0 0/1 \* \* \* \*

# HttpClient parameters

# The collector will visit the url every day to view the file, and if the project is # # # updated, it will save it to the database

checkmarx.server=http://127.0.0.1:8081/checkmarx-reports/checkmarx-current.xml

# CheckMarx login credentials

checkmarx.username=admin

checkmarx.password=admin

**BlackDuck**

1) Clone into the codesecurity folder of the blackduck collector from: https://github.com/matjukov-nikolaj/blackduck.git

2) Build Build mvn -Dpmd.failOnViolation = false clean install

3) In the file applicatiom.properties indicate the server with the blackduck report. The report is in the blackduck -current file and is updated every day.

4) Run the application: java -jar blackduck-collector-2.0.5-SNAPSHOT.jar --spring.config.name = blackduck --spring.config.location = D: \ {path to} \ Hygieia \ collectors \ codesecurity \ blackduck \ application.properties

**AppScan**

1) Clone the appscan collector in the codesecurity folder from: https://github.com/matjukov-nikolaj/appscan.git

2) Build Build mvn -Dpmd.failOnViolation = false clean install

3) In the file applicatiom.properties specify the server with the appscan report. The report is in the appscan -current file and is updated every day.

4) Run the application: java -jar appscan-collector-2.0.5-SNAPSHOT.jar --spring.config.name = appscan --spring.config.location = D: \ {path to} \ Hygieia \ collectors \ codesecurity \ appscan \ application.properties

\* -Dpmd.failOnViolation = false is used because there is a pmd error in the Hygieia code. Methods are taken from them, and they somehow hide it. In the CollectorTask class, the Component.class Spring and the Component.class Hygieia clash.

\* In application.propeties, you can configure cron - the application launch interval