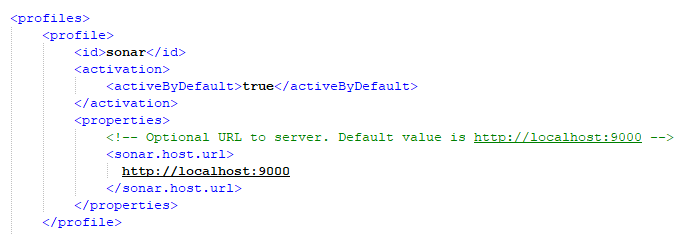
**Hygieia documentation**

**Установка Java**

1. Скачать Java JDK из <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. В переменных среды windows. Создать переменную JAVA\_HOME и указать путь до папки с java. Изменить переменную PATH добавить занчение; %JAVA\_HOME%\bin;

**Установка Maven**

1. Скачать apache\_maven c <http://maven.apache.org/download.cgi>
2. Добавить переменные среды MAVEN\_HOME и M2\_HOME c указанием пути до папки с apache\_maven. Изменить переменную Path %M2\_HOME%\bin
3. Зайти в папку ../apache\_maven/conf/ открыть settings.xml. Убрать комментарий и сделать настройки как на картинке. Если в системе есть sonar



**Установка MongoDB**

1. Скачать MongoDB <https://www.mongodb.com/download-center?jmp=nav#community>

Можно скачать архив.

1. Установить в С:\mongodb\
2. Добавить в переменную среды path С:\mongodb\bin
3. Создать в каком-нибудь каталоге папку data. Она нужная для хранения данных из базы. Внутри создать папки db и log. После изменить настройки mongod.cfg, который находится mongodb/bin/mongod.cfg dbpath=C:\data\db

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

1. Открыть командную строку от имени администратора прописать

Mongod

I NETWORK [initandlisten] waiting for connections on port 27017

После данного сообщения открыть еще одну командную строку от имени администратора. Прописать mongo. Наличие стрелки > говорит о том, что можно писать команды. Выбираем базу данных, вводим use dashboarddb. После вводим:



Получаем информацию о том, что пользователь успешно создан.

**Установка node.Js**

1. Установить Git.
2. Скачать Node.js c <https://nodejs.org/en/download/> Установить. Добавить переменные среды. Путь до папки Node.js
3. Открыть командную строку и установить bower и gulp. Командами: npm install -g bower

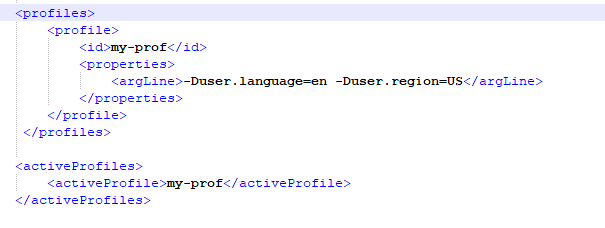
npm install -g gulp

**Установка Hygieia**

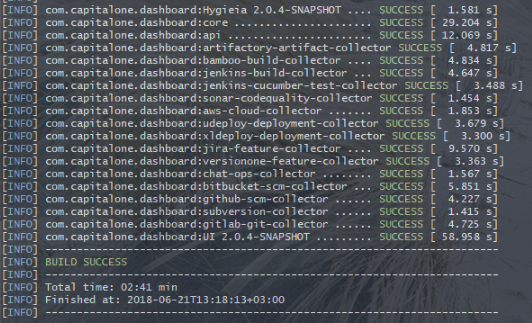
1. Скачать Hygieia с <https://github.com/capitalone/Hygieia.git>
2. Распаковать на диск.
3. Открыть командную строку зайти внутрь папки Hygieia. Прописать

mvn clean install package

* Версия Hygieia c мастера не собирается. Не проходит тесты. Тесты не проходит из-за locale. В США дробная часть от целой отделяется “,” у нас “.” Тут либо менять locale системы. Либо изменить настройки maven. Зайти в MAVEN\_HOME{path}/settings.xml



После установки должно показать вот это:



1. Запускаем mongod, после mongo
2. Создать в папке ..\api файл dashboard.properties с данными:

dbname=dashboarddb

#dbusername=dashboarduser

#dbpassword=dbpassword

dbhost=localhost

dbport=27017

1. Изменить файл \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. В Командной строке открыть api\target и запустить api командой

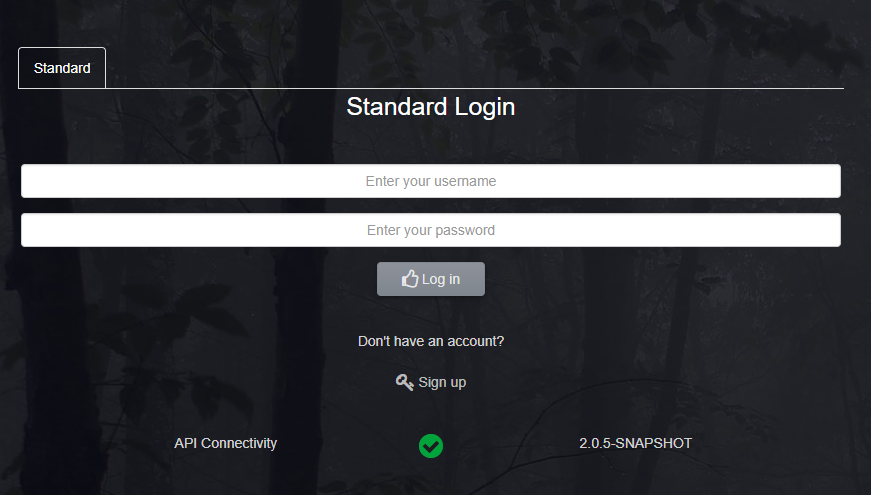
java -jar api.jar --spring.config.location=C:\[path to]\Hygieia\api\dashboard.properties -Djasypt.encryptor.password=hygieiasecret

\*Изменить путь, если он другой.

Log говорит о том что api запущен.

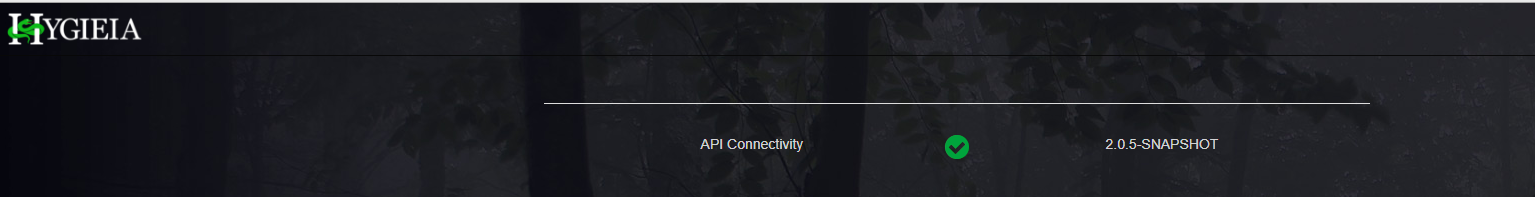
2018-07-26 15:25:45,082 INFO com.capitalone.dashboard.Application - Started Application in 22.585 seconds (JVM running for 24.17)

1. Открыть другую командную строку, перейти в папку Hygieia\UI и запустить UI командой gulp serve
2. Открыть в браузере страницу localhost:3000.



1. Создать пользователя. Галочка API Connectivity говорит о том, что API запущен и исправно работает. Если будет красный крестик, то скорее всего API не запущен или работает не корректно.

\*Если при запуске вылезает Exception MongoTimeoutException: Timed out after 30000 ms. То тут проблема с базой данных, либо не запущен mongod либо не верно указан url, можно попробовать 127.0.0.1 вместо localhost. Возможно не был создал пользователь dashboarduser.

\*Иногда при запуске страницы Hygieia. Страница авторизации может иметь такой вид.

Чтобы исправить это нужно почистить историю браузера.

**Включение коллекторов**

1. Для начала нужно скачать мои разработки с <https://github.com/matjukov-nikolaj>
2. Скачать библиотечку codesecurity c <https://github.com/matjukov-nikolaj/codesecurity.git>
3. Переместить ее в Hygieia. Зайти внутрь и установить: mvn clean install
4. Скачать Hygieia-utilites c <https://github.com/matjukov-nikolaj/Hygieia-utilites.git>
5. Переместить файлы с api по директориям указанным в dashboard. Так же заменить файл pom.xml файлом из Hygieia-utilites\api
6. Собрать Api без тестов Hygeia\api -> mvn clean install -DskipTests
7. Переместить файлы с Hygieia-utilites\core в Hygieia\core по тем же директориям.
8. Собрать ядро Hygieia\core mvn clean install
9. Переместить файлы с Hygieia-utilites\UI по директорриям UI.

**Sonar**

1. Сначала нужно выключить api и UI. Чтобы запустить sonar collector нужно перейти в директорию collectors/build/sonar
2. Создать файл application.properties

Прописать такие настройки:

# ---Настройки базы

# 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

# Настройки sonar collector

# Collector schedule (required). Время через которое перезапускается приложение

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

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

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

# Взависимости от установленной версии sonar прописать нужную версию и выбрать метрики.

# 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. Пароль и логин для доступа к проектам sonar.

sonar.username=admin

sonar.password=admin

1. Запустить sonar с папки target

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

1. Приложение запустится и будет искать sonar проекты. Запускаться будет через время указанное в cron.
2. Есть проблема когда при сборке ядра не создается коллекция code\_quality. Это можно посмотреть в mongo. Use dashboarddb-> show tables. Если среди них нет code\_quality то нужно создать эту коллекцию: db.createCollection(“code\_quality”); После скопировать файл с Hygieia-utilites/settings/sonar/SonarCollectorTask в папку sonar/src/…/collector/ и заново переустановить mvn clean install;
3. Чтобы проверить работоспособность коллектора он должен в коллекцию code\_quality класть данные: db.code\_quality.find()

**Jenkins codequality**

1. Суть запуска такая же как в Sonar.
2. Настроить файл application.properies
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. Установить mvn clean install c build/jenkins-codequality/
29. Запустить с target: java -jar jenkins-codequality.jar --spring.config.name=jenkins-codequality --spring.config.location=[path to application.properties file]

**CheckMarx**

1. Создать папку codesecurity в collectors. Склонировать туда: <https://github.com/matjukov-nikolaj/checkmarx.git>
2. Собрать mvn -Dpmd.failOnViolation=false clean install
3. В файле applicatiom.properties указать сервер с checkmarx отчётом. Отчёт находится в файле checkmarx-current и обновляется каждый день.
4. Запускаем приложение: 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

**BlackDuck**

1. Склонировать в папку codesecurity blackduck collector из: <https://github.com/matjukov-nikolaj/blackduck.git>
2. Собрать Собрать mvn -Dpmd.failOnViolation=false clean install
3. В файле applicatiom.properties указать сервер с blackduck отчётом. Отчёт находится в файле blackduck -current и обновляется каждый день.
4. Запускаем приложение: 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. Склонировать в папку codesecurity appscan collector из: https://github.com/matjukov-nikolaj/appscan.git
2. Собрать Собрать mvn -Dpmd.failOnViolation=false clean install
3. В файле applicatiom.properties указать сервер с appscan отчётом. Отчёт находится в файле appscan -current и обновляется каждый день.
4. Запускаем приложение: 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 используется потому что есть pmd ошибки в коде Hygieia. Методы взяты у них, и они это как-то скрывают. В классе CollectorTask происходит сталкивание Component.class Spring и Component.class Hygieia.

\*В application.propeties можно настроить cron – промежуток запуска приложения