# SEC YIELD API AND ENGINES CODE CHALLENGE

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Revision Number** | **Date** | **Comments** |
| TopCoder | 1 | 2016-08-05 | SEC YIELD API AND ENGINES CODE CHALLENGE |
|  |  |  |  |

Table of Contents

[SEC YIELD API AND ENGINES CODE CHALLENGE 1](#_Toc458189033)

[1 Prerequisites 4](#_Toc458189034)

[2 Project Structure 4](#_Toc458189035)

[2.1 Root Structure 4](#_Toc458189036)

[2.2 Resources (src/main/resources) 4](#_Toc458189037)

[2.3 Test classes (src/test/java) 5](#_Toc458189038)

[2.4 Java classes (src/main/java) 5](#_Toc458189039)

[3 Configuration 6](#_Toc458189040)

[3.1 YtmYieldCalculationEngine 6](#_Toc458189041)

[3.2 SECYieldServiceIml 7](#_Toc458189042)

[4 Deployment 8](#_Toc458189043)

[5 Verification 9](#_Toc458189044)

[6 Sonar 11](#_Toc458189045)

[7 Contact 11](#_Toc458189046)

# Prerequisites

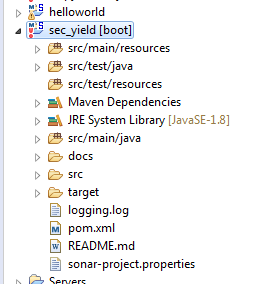
Java 8 <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Tomcat 8 <https://tomcat.apache.org/download-80.cgi>

Maven <https://maven.apache.org/download.cgi>

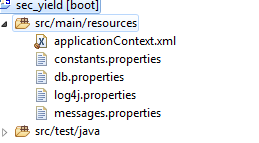
# Project Structure

## Root Structure

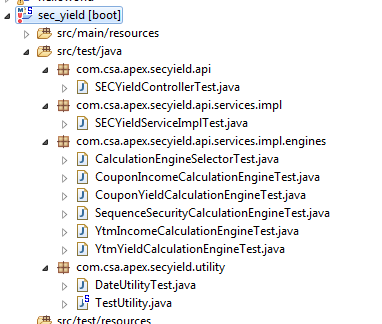


## Resources (src/main/resources)

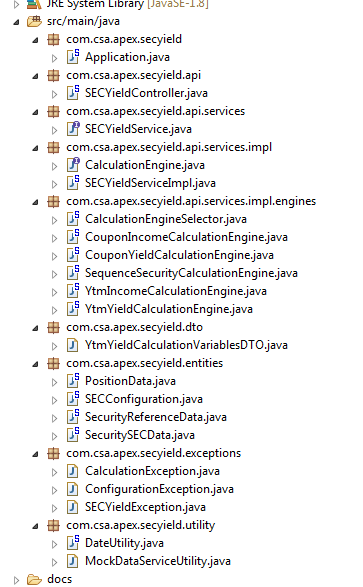
* applicationContext.xml: Application Context file
* constant.properties : All constants, methods names and any other info
* db.properties : DB properties
* messages.properties : All messages



## Test classes (src/test/java)



## Java classes (src/main/java)



# Configuration

## YtmYieldCalculationEngine

* calculationStep

It is the step to increase the beginning yield in order to find the range of the yield.

* binarySearchCount

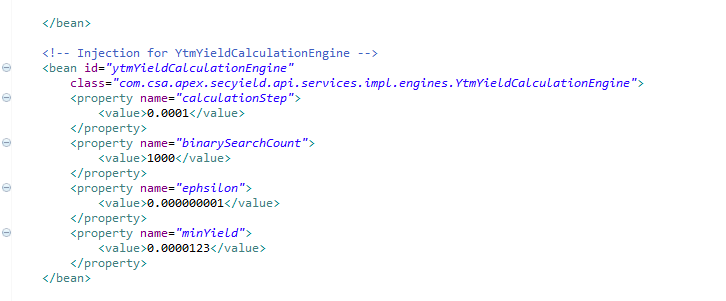
It is count to perform binary search. The larger value will result in more precise outcome yield.

* ephsilon

It is used in binary search for convergence, so we can say it is amount of the least error in calculation.

* minYield

Yield value if F(Y = 0 + ephsilon) < 0 , so basically yield value when actual yield is less than 0



## SECYieldServiceIml

* getConfigApiPath

The REST API path to the endpoint used to obtain calculation engine configurations. After injection should be non-null and non-empty.

* getCustomerDataApiPath

The REST API path to the endpoint used to get security SEC data. After injection should be non-null and non-empty.

* saveCalculatedSecuritySECDataApiPath

The REST API path to the endpoint used to save calculated security SEC data. After injection should be non-null and non-empty.

* getCalculatedSecuritySECDataApiPath

The REST API path to the endpoint used to get already calculated security SEC data. After injection should be non-null and non-empty.

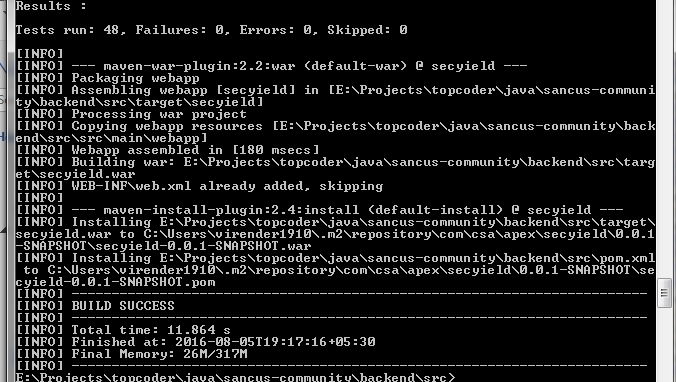
* doMock

Mocks all the calls to customer API for demo purpose

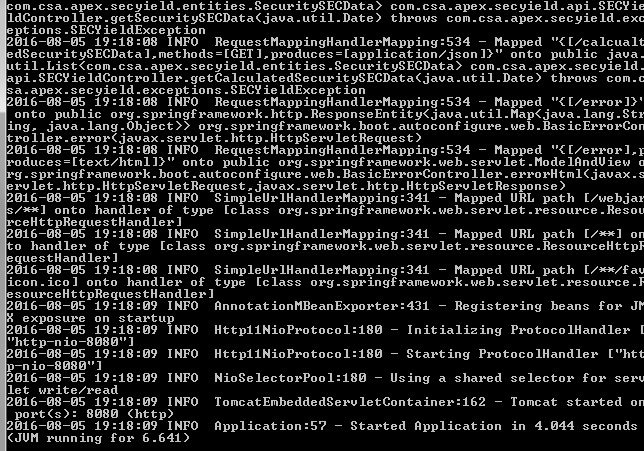


# Deployment

$ mvn clean install



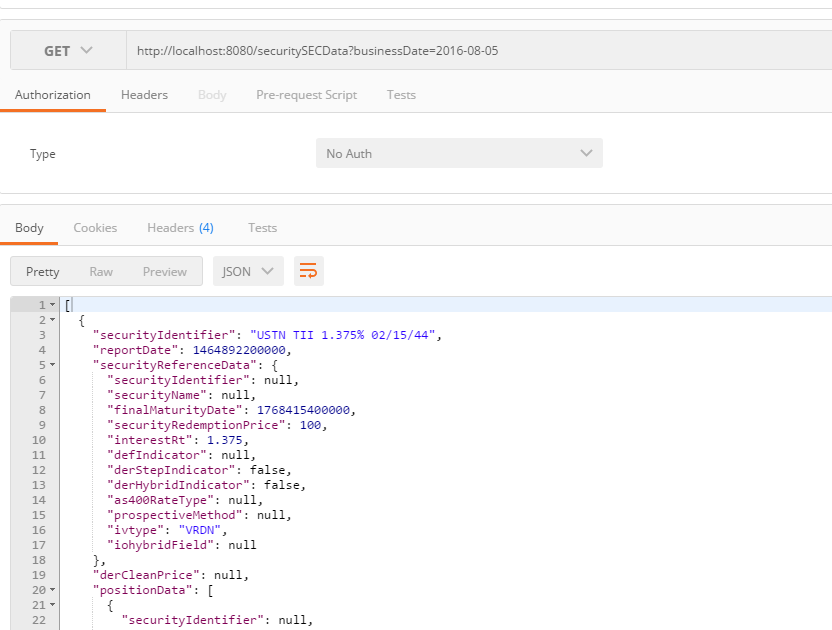
$ mvn spring-boot:run



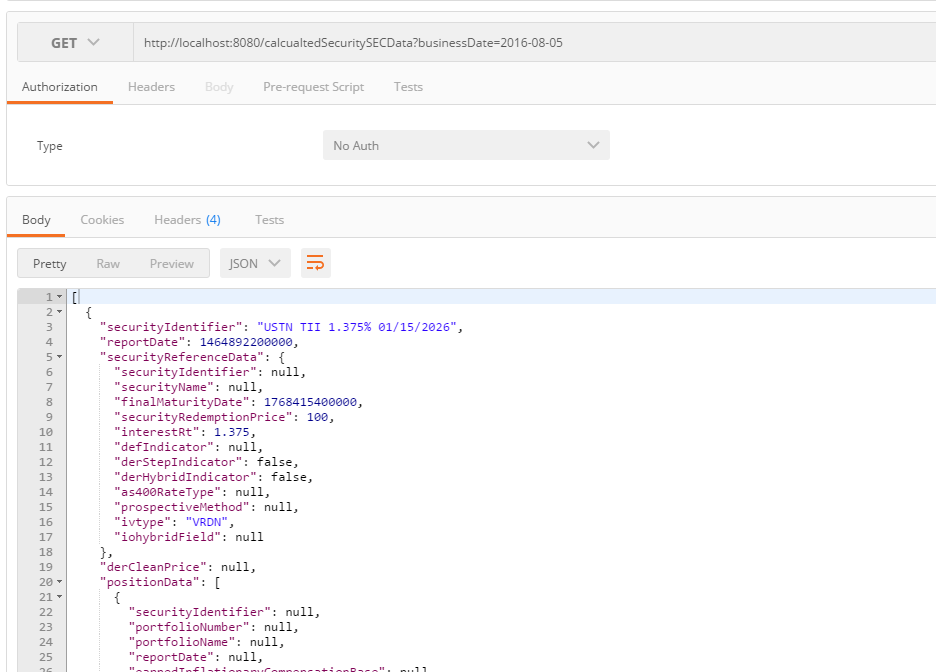
# Verification

Open docs/sec\_yield\_api\_postman\_collection.json in POSTMAN

(/securitySECDate)

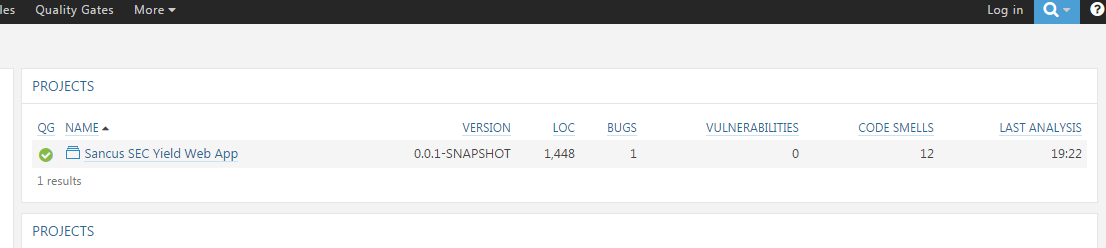


(/calculatedSecuritySECData)



# Sonar

$ mvn sonar:sonar



# Contact

|  |  |
| --- | --- |
| Topcoder |  |