Mahisoft

API Training

Workshop 1 - Rest

January 2016

Overview

The current workshop exercise one aims to introduce participants into the API world. As part of the training participants should exercise their basic to advanced skills in RESTFUL APIs, Contract Oriented programming, Service Oriented Software, Software Development Patterns and Refactoring.

Goals

1. **Self assessment of Proficiency.**
2. **Self assessment of knowledge base.**
3. **Identification of areas of improvement.**

Workshop

For the first API workshops it is expected of the trainee to investigate and practice about REST and Services as a concept and then implement examples in the technology of the project with priority for the trainee.

Tasks

1. Investigation and questions:
    a. Terminology research (for each term please give a introduction and a brief explanation of their concept and use)
        i. REST
        ii. HTTP
        iii. TCP/IP
        iv. HTTP Request
        v. HTTP Response
        vi. REST Resource
    b. Explain the use of 1XX, 2XX, 3XX, 4XX and 5XX HTTP response codes
    c. How is REST different from other alternatives like RPC or Web Services (WSDL SOAP etc) ?
    d. Which software patterns are commonly built with REST API endpoints? .

e. (For functional enthusiasts) Find/Research functional practices advantages/disadvantages within an API development context.

2. Some agreements:

a. Workshop activities are delivered through a **git** public repository owned by the participant. Each workshop should have a PR to the master branch of the public repository for the workshop. Part of the facilitators assessment will be to review the workshop PR.

b. If there a things needed to be installed to run code examples. The PR should have a bash script for installation.

c. Every activity of the workshop must have a corresponding script that shows an example of the activity being done. Postman scripts and exports are valid ways to deliver activities.

d. The default folder of work in linux for testing should be **$HOME/work/workshop-api**

3. Reviewing time

All API workshops will be reviewed and given feedback to close to the presentation of the next workshop. There are two components of the evaluation a Self Assessment one a the Facilitators comments.

For the self assessment part we require a comment from the participant explaining the difficulties and challenges of the workshop, the references used or links and what was the most important learning experience of the workshop.

The facilitator comments will be given to in two ways. One is in the PRs for every workshop the second is in the next workshop presentation (meeting).

Concepts/Technologies to practice:

1) JAVA 1.8
2) Spring Framework - Spring Boot 1.4
3) Maven / Gradle
4) Postman
5) RESTFUL API

Exercises:

NOTE: The design granularity and complexity goes as far as the participant needs or wants.

1) Build a header checker software middleware that implements the validator pattern. (For functional enthusiasts: Use only lambdas for checks, Beware of exception handling in functional context).

EXTRA: **(NOTE: Any extra not covered may appear later on in other workshops.)**

2) Build a spring configuration class for that sets up an environment in terms of:
    a) The precision and comparison method (round up, round down, etc.) that should be used for every BigDecimal comparison in the system.
    b) A library of third party urls by hardwire name codes.

Software Patterns of the Workshop:

1) Creational - Factory
2) Behavioural - Validator
3) Structural - Adapter

For each pattern identify:

a) Where have you seen in code this pattern?
b) What are the advantages and disadvantages if any?
c) How can you apply it in at least 2 real life scenarios?

Extra:

d) Show code examples of at least 1.