Project Summary – Swagger UI

Table of Contents

[Introduction 1](#_Toc423384617)

[Rest 1](#_Toc423384618)

[What is Swagger UI? 2](#_Toc423384619)

[Swagger documentation 3](#_Toc423384620)

[Modeling 3](#_Toc423384621)

[Sequence Diagram 3](#_Toc423384622)

[Activity Diagram 4](#_Toc423384623)

[Class Diagram 5](#_Toc423384624)

[Challenges for Swagger UI project 6](#_Toc423384625)

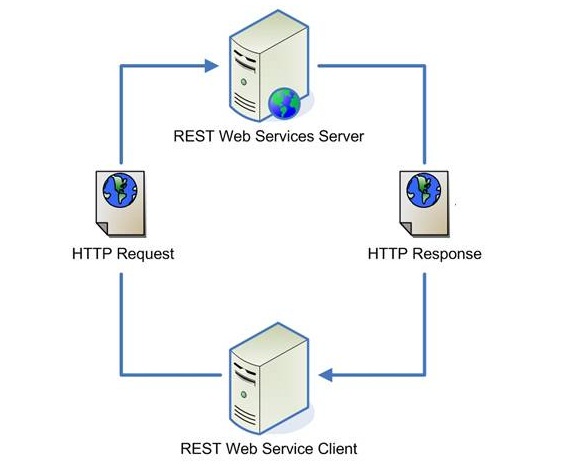
[Competitors comparison 7](#_Toc423384626)

[Swagger UI main contributors 8](#_Toc423384627)

# Introduction

## Rest

Representational state transfer (REST) is a software architecture style consisting of guidelines and best practices for creating scalable web services.



* **Problem**: describing the different APIs url and parameters.
* **Solution**: Swagger UI

# What is Swagger UI?

* Define a standard interface for describing APIs.
* Allows both humans and computers to discover and understand the capabilities of the service without access to source code.
* Similar to what interfaces have done for lower-level programming, Swagger removes the guesswork in calling the service.
* Swagger is made up of three components:
* Server: hosts the REST APIs that you want to use.
* Swagger Server: holds the API specification.
* Swagger UI Client: Reads a description of the APIs from the Swagger Server and renders it as a web-page and an interactive sandbox to play with the APIs.

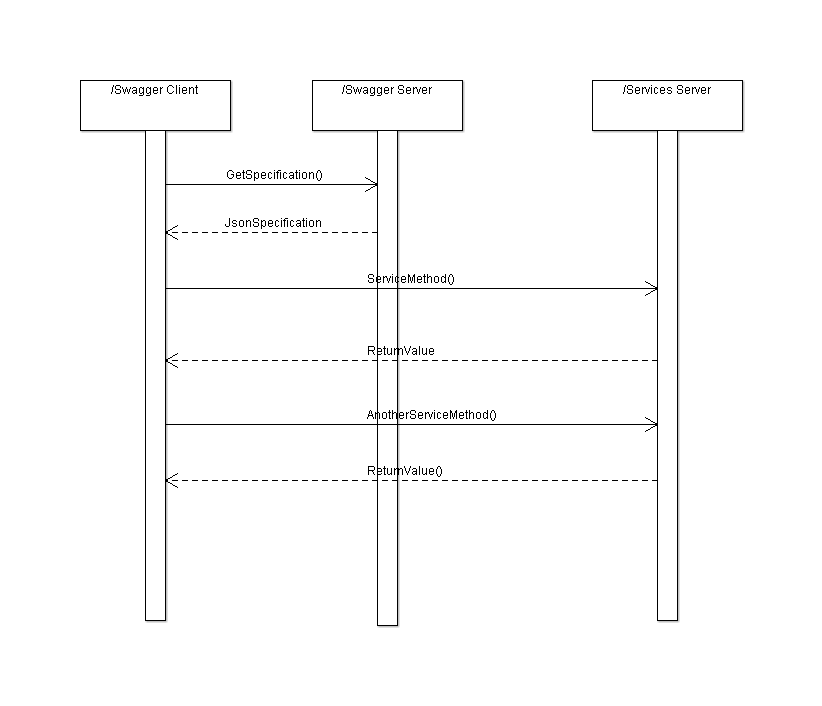
# Swagger documentation

* Swagger specification repository – provides references for swagger related documentation and tools.
* Repository – src/main/html/index.html - the main swagger ui html page.
* Src/main/javascript/Swaggerui.js – the main javascript file for swagger.

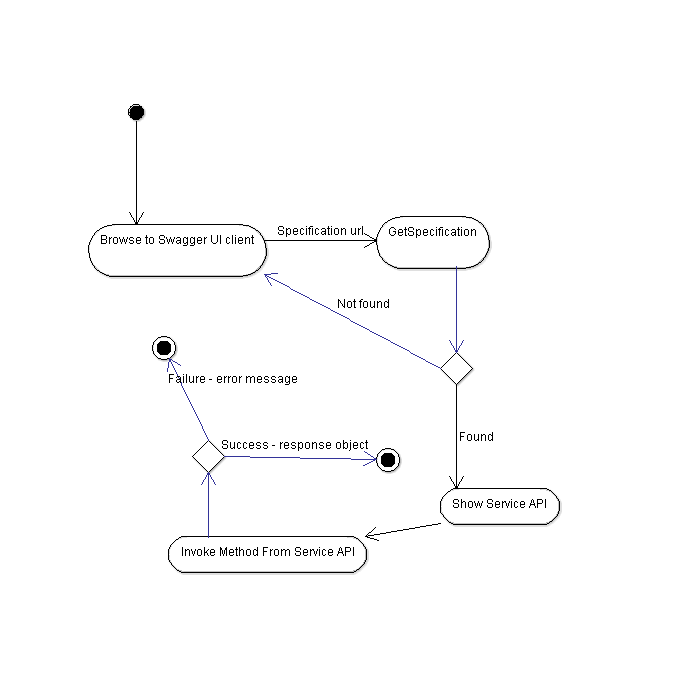
# Modeling

Modeling the Swagger UI helped us to understand better the design, flow and components of Swagger UI.

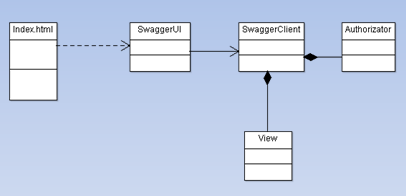
## Sequence Diagram

****

## Activity Diagram



## Class Diagram

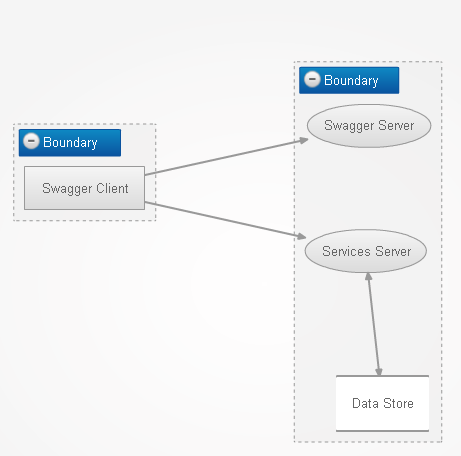


## Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an [information system](https://en.wikipedia.org/wiki/Information_system), modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated.

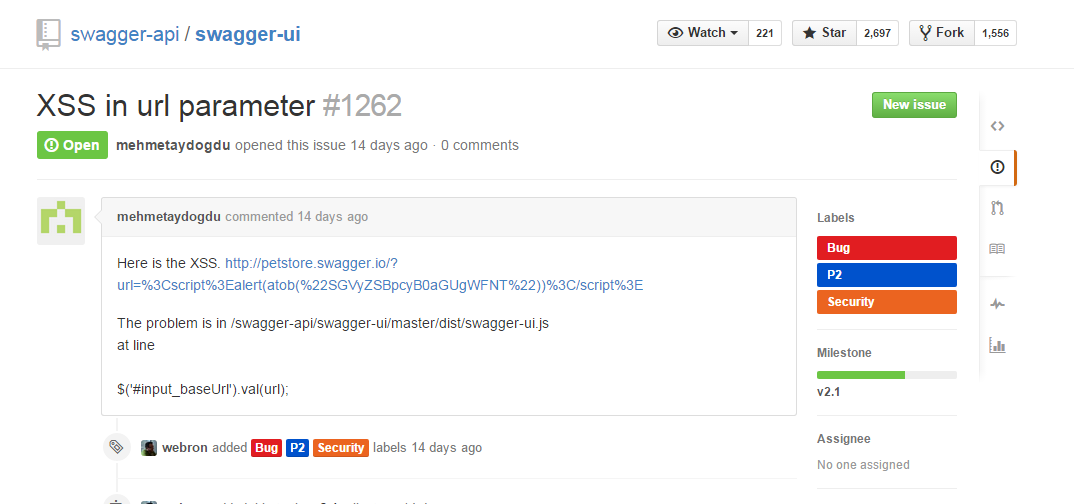
This is new diagram we have added to understand the processes in the system and the flow between it.

The Boundary is “Trust Boundary” in the perspective of security, which mean which processes can trust each other.

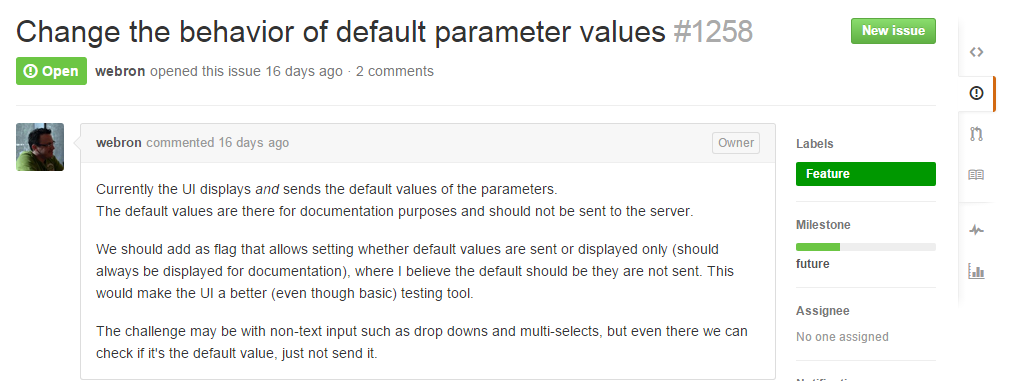


# Challenges for Swagger UI project

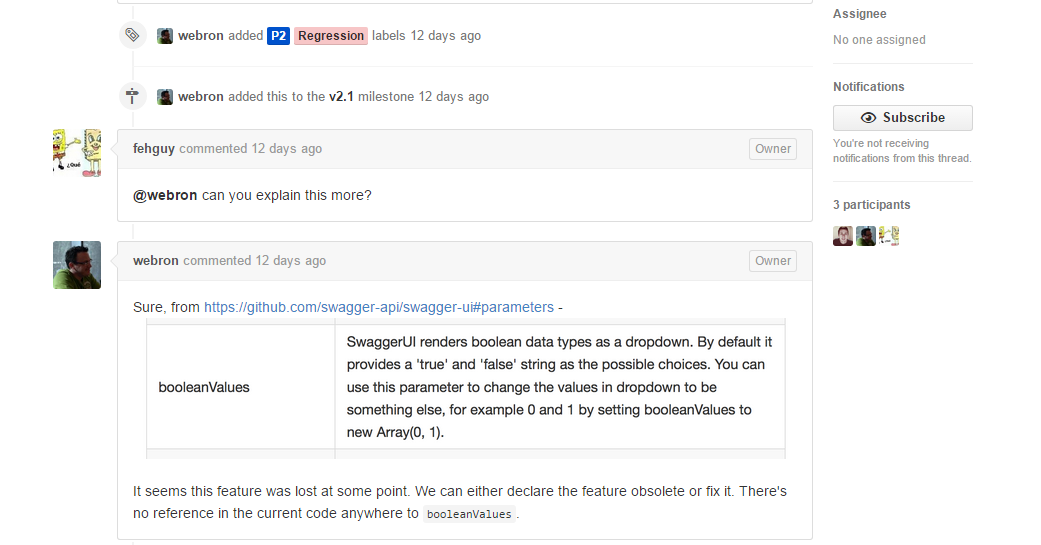
* **Security Bugs**: Cross Site Scripting(XSS).
* enables attackers to inject client side script into web pages viewed by other users.



* **Additional Features**:
* add a flag that allows setting whether default values are sent or displayed only.



* **Regression** - config for booleanValues gone.



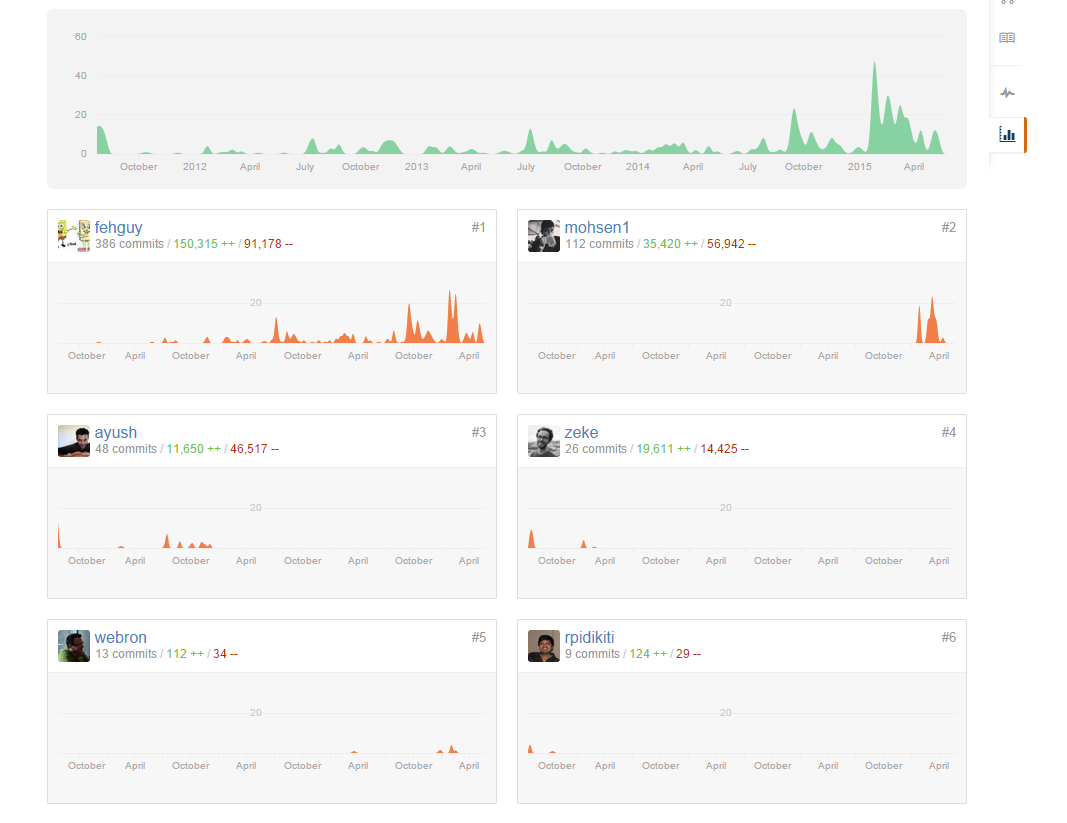
# Competitors comparison

Main competitor: **Mulesoft**

* **Not open source – can’t change the code to your needs.**
* **Cost money.**
* **Smaller community – need to wait sometimes couple of days for support from the support team.**
* **Better UI.**
* **Integration with more MuleSoft components.**
* **Dedicated support team.**

# Swagger UI main contributors

Overall 81 contributors in 4 years



# Summary

In our project we took an open source project and understood it by applying the modeling techniques we have learnt in the course.

The modeling part helped us a lot by understanding the components of the project in high lever as well as understanding the flow between the components.

Also we understood lower lever details about the code by applying class diagram which helped us understand the relationships between the different classes.