Server for Asynchronous Protocols Mocking

1. Introduction

The reasons for project:

* Mocking reduces dependencies between teams (<https://learning.getpostman.com/docs/postman/mock_servers/intro_to_mock_servers/>, <https://medium.com/billie-finanzratgeber/mock-server-save-time-and-money-during-development-and-testing-bf1d74364d84>)
* Mocking helps to prototype (<https://learning.getpostman.com/docs/postman/mock_servers/intro_to_mock_servers/>)
* Mocking helps while testing (<https://medium.com/fabricgroup/using-mock-server-with-refit-to-test-restful-api-integration-802f377c4c6e>, <https://medium.com/billie-finanzratgeber/mock-server-save-time-and-money-during-development-and-testing-bf1d74364d84>)
* No flexible mock servers for asyncronous protocols like websocket or sse (<https://groups.google.com/forum/#!topic/mountebank-discuss/t19u-sc8b6E>)

Description of outcome

Outline the structure

1. Main body
   1. Terms

Here will be the definition of main terms (server, mocking, mock server…)

* 1. Analogs for HTTP mocking

Here will be a small description of main mock servers for mocking HTTP servers (<http://www.mbtest.org>, <http://www.mock-server.com> )

* 1. Analogs for Asyncronous mocking

The failure to find an implemented server for mocking SSE and Web Socket protocols. Ways to imitate them (<https://groups.google.com/forum/#!topic/mountebank-discuss/t19u-sc8b6E>) and why they are not appropriate in some cases

* 1. Main requirements to the outcome
* Dockerization
* Configuration via REST API
* Actual mocking
* Configuration via JSON
  1. Implementation notes
* Development stack choice
* Components decomposition

1. Conclusion

Sum up the expected outcomes, benefits development teams are going to get when the first implementation is ready.