Aswat – Interview candidate test: API development

Document	Aswat-ICTAD-20150617
Author(s)	Loïc Tosser <loic.tosser@aswat-telecom.com> Rajesh Vaya <rajesh.vaya@aswat-telecom.com>
Version	1.0
Status	Draft

Table of Contents

Project case

Develop a simple REST API (keeping mind the RFC standards) to facilitate a simple shopping experience.

Brief semantic requirements

You must design the architecture around the 3 main entities

- Customers
- Products
- Cart

API features

- The API should be able to register / update customer
- The API should be able to login / logout the customer (must manage the session handling)
- The API should return the list of products and its available quantity (The products data can be a static database, JSON file etc. or whatever suits your development cycle)
- The API should be able to add / remove items to the logged in customer's cart
- The API should be able to fetch the cart of the logged in customer

API performance

- The API should be able to handle millions of customer profiles / cart data
- The API should be able to handle multiple concurrent requests
- The API should be able to respond to the request in ~10 to ~100 milliseconds
- The API should not be at risk when load testing / injection attacks are done

Documentations

- The code should be well documented
- Code coverage report should be produced

Testing

- Best to have automated unit test cases
- Should be testable for load testing
- Should be testable against injection attacks

Technology

- Use any backend and database you are comfortable with, we will only judge on the architecture setup, but choosing the right technology dictates a lot about architecture

Show-off

- Show-off your skills in whatever way possible.
- If you think you can add more features (e.g. admin, ACL, caching), then do so
- Innovative ways of documentation, testing, automation etc
- How much ever you do, do it properly - Don’t commit incomplete code or buggy code

Submission requirements

- The entire project should be hosted on github with proper README file containing every detail to deploy, test, list of URLs and any other documentations which may be need to setup and run the API
- Every feature / change / fix should belong to a commit and commit messages should be descriptive enough (Don’t commit your entire project / progress in just few commits). We should be able to see the evolution of the code through commits
- You are allowed to use any framework / language, however copying chunks of code from online will not be appreciated as your code will be checked against plagiarism
- Links to your test environment would be helpful (but not required)