Exercise 3

1) Sample-Bank: Do you remember the loan feature we specified in the first exercise? So, it is time to automate it. Create the story and scenarios to test the Loan Service and all its validations. For this test you should create the story file with the scenarios, the steps class and the service communication.

Remember the Sample Bank credit rules:

This service allows the customer to get a loan following some credit rules. Then to get a loan the customer have to:

* Have a minimum balance $2000 in his account.
* The customer can just get a loan of 30% of his actual balance.
* The customer can only get the loan once.

The REST Service is (Do not forget to run the app by running the class Application.java):

* [**http://localhost:8080/samplebank/api/accounts/credit/{ownerCpf}/loan?value={amount}**](http://localhost:8080/samplebank/api/accounts/credit/%7bownerCpf%7d/loan?value=%7bamount%7d)

**Parameters:**

* + **ownerCpf: CPF associated to the account using the credit.**
  + **loan: Requested loan amount.**

Extra Tips:

* Write a new story to test this service since is related to a new credit service.
* Test Scenarios:
  + Customer is able to get a loan.
  + Customer does not have the minimum balance to get a loan.
  + Customer tries to get a loan that exceeds the allowed 30% of his actual balance amount.
  + Customer already has a pending loan.
  + The validation return codes and messages that you will need in the assertions can be find in the Enum **ResponseMessage.java**.
  + Don’t forget that the accounts created are cleaned after the scenario. In the **ScenarioLifecycle** class, take a look in the method annotated with @AfterScenario.