Lot 8 : Test Automation

UI Testing

Workshop on October 30th 2020

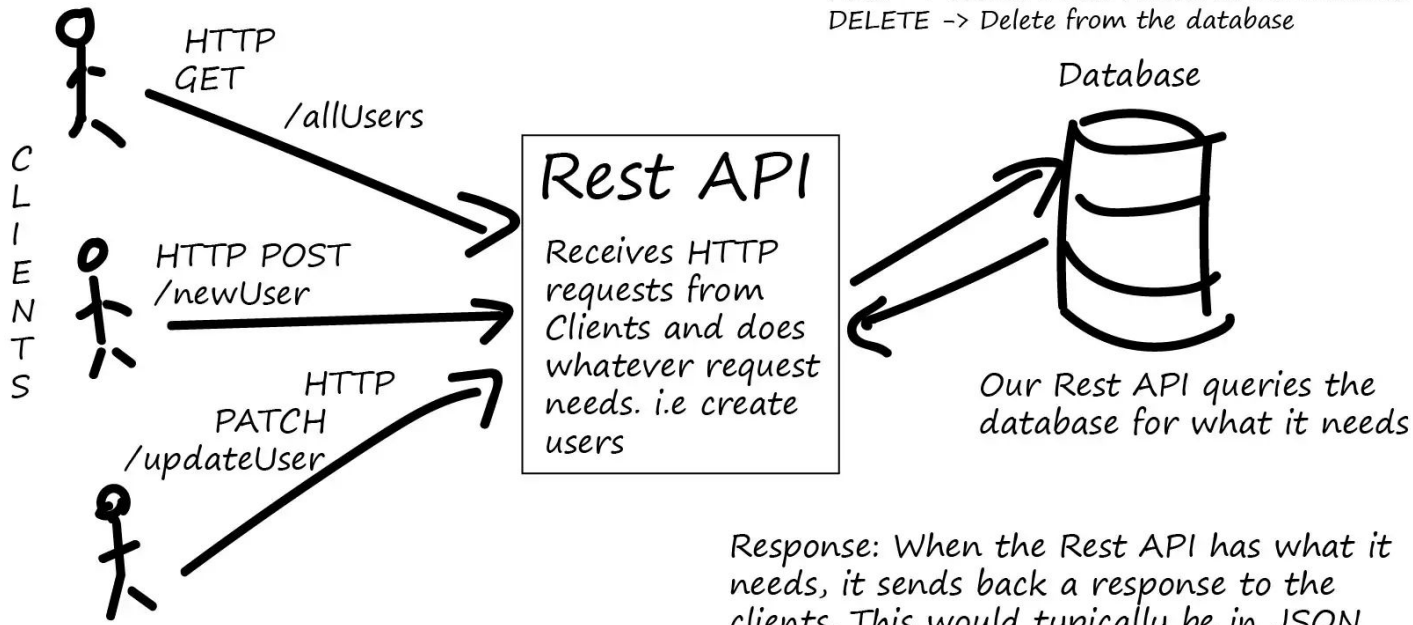
Agenda

1. Recap Day 2
2. REST-Assured Workshop
3. UI Testing
4. Selenium Workshop

Recap Day 2

Rest API Basics

Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database



Our Rest API queries the database for what it needs

Our Clients, send HTTP Requests and wait for responses

Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.

REST-Assured Workshop

Test Automation

- Test Automation helps to ease the repetitive tasks of Manual testers.
- Test Automation covers many scenarios within a little time
- Does NOT replace manual testing


Core Functional Testing
GUI Testing
DB Testing
Usability Testing
Regression Testing
Sanity Testing
Cross Browser Testing
E2E Testing
Business Process Testing
Se

Selenium

- Open source automation test suite of different tools (language agnostic) focused to automate web application testing
- Consists of
  - Selenium IDE
  - Selenium Remote Control
  - Selenium WebDriver
  - Selenium Grid

Selenium WebDriver

- Unlike other selenium tools, it does not rely on Javascript for Automation. A direct communication is established with the browser
- WebDriver can only support web based applications
- Supported languages
  - Java
  - C#
  - Ruby
  - Python
  - Javascript

Selenium WebDriver - Basic Commands

- Getting a web page
  - driver.get(“www.target.url”)
  - driver.Navigate().to(“www.target.url”)
- Browser events
  - driver.Navigate().back()
  - driver.Navigate().forward()
  - driver.Navigate().refresh()

Selenium WebDriver - Basic Commands

- Locating HTML elements
  - driver.findElement(By.id(“id”))
  - driver.findElement(By.name(“id”))
  - driver.findElement(By.xpath(“id”))
  - driver.findElement(By.cssSelector(“id”))
  - driver.findElement(By.tagName(“id”))
  - driver.findElement(By.className(“id”))
- Input fields commands

  driver.findElement(By.id(“id”))
  - sendKeys()
  - clear()
  - getText()
  - click()

Selenium WebDriver - Basic Commands

- Closing the browser
  - driver.close();
- Close all the browser instances associated with the driver
  - driver.quit()

Selenium WebDriver - Basic Commands

- Worth mentioning
  - Actions
  - Switches
  - getWindowHandles()/getWindowHandle()
  - Handing alerts

Selenium WebDriver - Wait Commands

- Implicit wait
  - Applied for the driver instance
  - Wait before it throws a “No Such Element Exception”
    driver.manage().timeouts().implicitWait(<Time to wait>, TimeUnit.SECONDS)
    // Can be seconds, milliseconds, minutes …

- Explicit wait
  - Wait for certain condition (Expected conditions) to be :
    - True within the given time
    - False when the given time exceeds
  - Applied for specific element

Selenium WebDriver - Wait Commands

- Fluent wait
  - Wait for a condition, as well as the frequency with which we want to check the condition before throwing an Exception

    ```
    Wait wait = new FluentWait(WebDriver reference)
            .withTimeout(timeout, SECONDS)
            .pollingEvery(timeout, SECONDS)
            .ignoring(Exception.class);
    ```

- Thread.sleep(<time in ms>)
  - Helps to sleep / suspend the test execution for the given time. Should only be reserved for debugging, DO NOT PUBLISH

Selenium WebDriver - Assert and Verify

- Used to find whether a given input is present or not on the webpage

- When an “assert” command fails, the test execution will be aborted.
  So when the Assertion fails, all the steps after that line of code are skipped

- When a “verify” command fails, the test will continue executing and logging the failure

Selenium Workshop