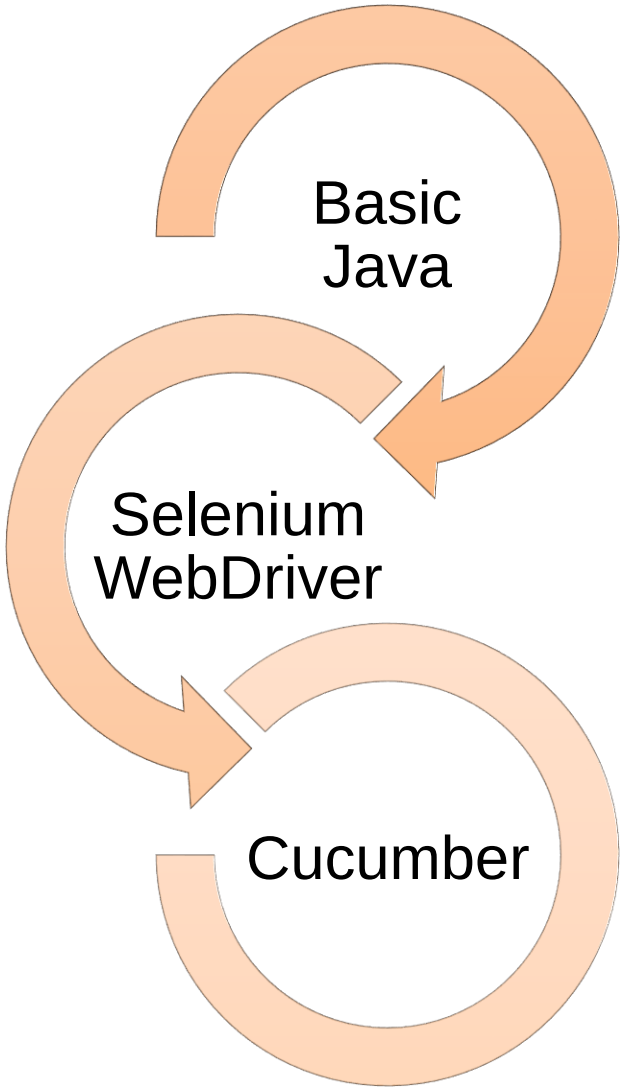
accenture
Latvia ATC

Test Automation Engineering Fundamentals: Cucumber

Cucumber intro

Looking back at the “Plan for the course”

accenture
Latvia ATC



Cucumber

accenture
Latvia ATC

Test Automation Engineering Fundamentals

Reminder on Cucumber

accenture
Latvia ATC

Cucumber is a software tool that computer programmers use for testing other software. It runs automated acceptance tests written in a behavior-driven development (BDD) style.

What is BDD?

accenture
Latvia ATC

Behavior-driven development (BDD) is a software development methodology in which an application is specified and designed by describing how its behavior should appear to an outside observer.

Therefor first of all we explain how software should behave.

What is BDD?

accenture
Latvia ATC

Often expected behavior is written as “user story”, which is written in format:

“As a _____, I want _____, so that ____.”

And then there are “acceptance criteria” which specifies if the story is implemented correctly.

Acceptance criteria, can be in a format of:

- **Checklist;**
- **Scenario(s) (e.g. in gherkin format using such keywords as Given, When, Then and And).**

Gherkins

accenture
Latvia ATC

```
Scenario: The name of the scenario
  Given an initial condition
  When something happens
  Then this is the result
```

The word 'And' chips in to expand the complexity of the scenario if required:

```
Scenario: The name of the scenario
  Given an initial condition
  And another condition
  When something happens
  And something else happens
  Then this is the result
  And this is also the result
```

User story example

accenture
Latvia ATC

Title: Order Cheque Book
As a **customer**
I want **to access my online account**
So that **I can order a new cheque book**

Acceptance Criteria

1) The bank will only send out a cheque book if the customer is in credit.
2) The user can only request a cheque book if the bank has his/her complete address details.

User story example

accenture
Latvia ATC

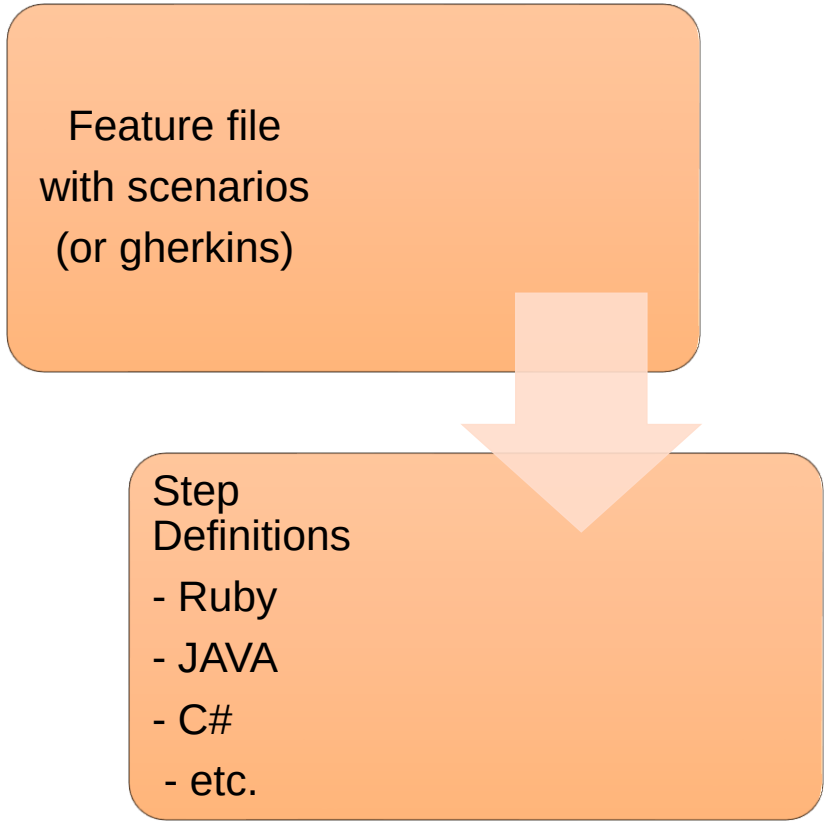
```
Scenario: Order a Cheque Book
  Given the account is in credit
  And the user has been authenticated
  And the user's address is available
  When the user clicks on 'order a cheque book’
  Then send cheque book to user

Scenario: Order a Cheque Book, no address available
  Given the account is in credit
  And the user has been authenticated
  And the bank does not have complete user address details
  When the user clicks on 'order a cheque book’
  Then ask user to complete address
```

Cucumber

accenture
Latvia ATC

And in cucumber we can automate those scenarios to run as tests by defining each step in a language of our choose.



Setting up project

accenture
Latvia ATC

Test Automation Engineering Fundamentals

Setting up project - installing Maven

accenture
Latvia ATC

- **Check if maven is installed:**
  1. Go to Help -> Eclipse Marketplace
  2. “Installed” tab
  3. Check if “Maven Integration for Eclipse” is there
- **If not then install maven by:**
  1. Go to Help -> Eclipse Marketplace
  2. Search by Maven
  3. Click “Install” button at “Maven Integration for Eclipse”
  4. Follow the instruction step by step
  5. Restart Eclipse

Setting up project - installing Natural

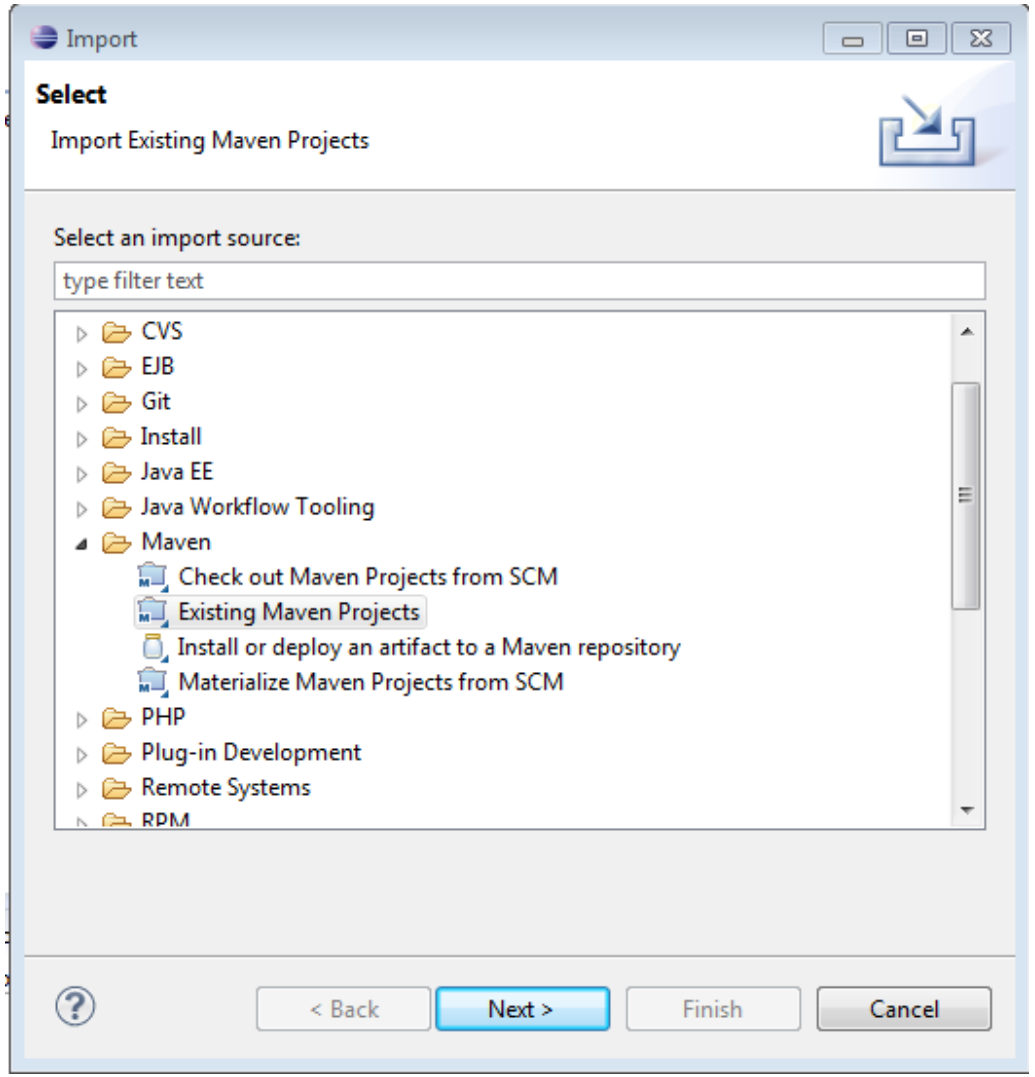
accenture
Latvia ATC

- **Check if maven is installed:**
  1. Go to Help -> Eclipse Marketplace
  2. “Installed” tab
  3. Check if “Natural” is there
- **If not then install Natural by:**
  1. Go to Help -> Eclipse Marketplace
  2. Search by Natural
  3. Click "Install" button at “Natural 0.7.6”
  4. Follow the instruction step by step
  5. Restart Eclipse

Setting up project – importing project

accenture
Latvia ATC

1. **Create a new project by selecting File -> Import**
2. **In the wizard, select Existing Maven Projects -> Next**
3. **“Browse…” to the Project**
4. **Finish**



Running project

accenture
Latvia ATC

- **Right-click on project -> “Run as…” -> Maven install**
- **Right-click on project -> “Run as…” -> JUnit Tests or Maven test**

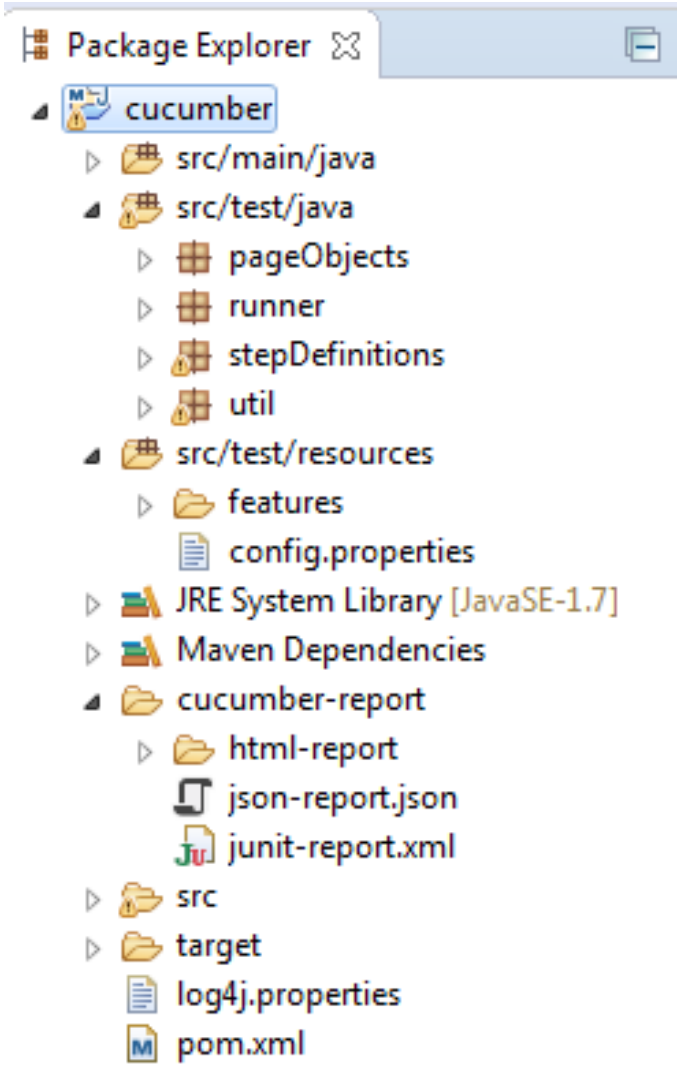
Cucumber project structure

accenture
Latvia ATC

We will be looking at the following parts of project:

- **Features;**
- **Step Definitions;**
- **Runner;**
- **Reports;**
- **Utils.**

Note: We already discussed what is "Page Object", so we will not looking at this part.



Cucumber project structure - Features

accenture
Latvia ATC

Feature file consists of user story and acceptance criteria in gherkin format.
Feature files have ".feature" extension.
And can be found in "src/test/resources" folder for our project.
Example of feature file:

```
Feature: Login
As a user I want to login to see welcome text

  Scenario: Successful login
    Given I have a user
    When I login
    Then I see Welcome text
```

Note: Cucumber can be used for testing all kind of application, not only web.

Cucumber project structure - Step Definitions example

accenture
Latvia ATC

Step definitions are saved in simple Java class.
And can be found in “src/test/java/stepDefinitions” folder for our project. In constructor of class driver and pages may be defined, e.g.:

```
public class loginSteps {
    LoginPage loginPage;
    WelcomePage welcomePage;
    String userName = "";
    String password = "";

    public loginSteps (Driver driver) {
        this.driver = driver;
        loginPage = PageFactory.initElements(driver, LoginPage.class);
        welcomePage = PageFactory.initElements(driver, WelcomePage.class);
    }
}
```

Cucumber project structure - Step Definitions example

accenture
Latvia ATC

```
@Given("^I have a user$")
public void I_have_a_user() throws Throwable {
    userName = user1;
    password = passw0rd;
}

@When("^I login$")
public void I_login() throws Throwable {
    loginPage.loginWithCredentials(userName, password);
}

@Then("^I see Welcome text$")
public void I_see_Welcome_text() throws Throwable {
    assertEquals("Welcome, " + userName, welcomePage.getHeader());
}
```

Cucumber project structure - Runner

accenture
Latvia ATC

In order for this project to run we need a runner class, e.g.:

```
@RunWith(Cucumber.class)
@CucumberOptions(
    features = "src/test/resources/features/login.feature",
    plugin = { "pretty", "html:cucumber-report/html-report",
               "junit:cucumber-report/junit-report.xml",
               "json:cucumber-report/json-report.json" },
    glue = { "stepDefinitions" })
public class CucumberRunner {}
```

Cucumber project structure – Reports and Utils

accenture
Latvia ATC

After each run reports are generated, which where specified in the runner.

Utils have a few helper classes, such as Driver and Properties.
Driver in return has some pre-defined things, which can help:

- **“**`isElementPresent(By)`**” or “**`isElementPresent(element)`**”**
- **“**`waitForInvisibilityOfElement(By)`**”.**

QUESTIONS

accenture
Latvia ATC

Activity

accenture
Latvia ATC

1. **Create a new feature file;**
2. **Add a scenario, like:**
    1. Opening a page (e.g. google.com);
    2. Checking that some elements are present or have correct text (e.g. if you opened google that input field, search and “I’m feeling Lucky” button are present and have correct text).
3. **Create a new java steps file, where you define steps above;**
4. **Change in Runner which feature is being run and run the tests, fix them if something is failing;**
5. **Look at what kind of reports are generated after test run.**

Note: for this activity page object is optional, not mandatory.