**Exemplu de project**

* Fie User:

@NoArgsConstructor  
@AllArgsConstructor  
@Getter  
@Setter  
@ToString  
@Builder  
@EqualsAndHashCode  
public class User {  
 private String firstName;  
 private String lastName;  
 private Integer age;  
}

* Acum fie RestControllerul:

@RestController  
@RequestMapping("/users")  
public class RestAPI {  
 @GetMapping  
 public List<User> getAll(){  
 return List.*of*(User.*builder*().firstName("Mititiuc").lastName("Eduard").age(21).build(),  
 User.*builder*().firstName("Goncear").lastName("Alex").age(21).build(),  
 User.*builder*().firstName("Turcan").lastName("Petru").age(20).build());  
  
 }  
 @GetMapping("/{age}")  
 public List<User> getByAge(@PathVariable int age){  
 return List.*of*(User.*builder*().firstName("Mititiuc").lastName("Eduard").age(21).build(),  
 User.*builder*().firstName("Goncear").lastName("Alex").age(21).build());  
 }  
}

* Acum, test va avea asa structura:



DemoApplicationTests va fi asa:

@SpringBootTest  
@RunWith(Cucumber.class)  
@CucumberOptions(  
 glue = "com.example.demo.glue",  
 features = "classpath:features"  
)  
class DemoApplicationTests {  
  
}

SpringBootRunCuncumberTests:

@SpringBootTest(webEnvironment = SpringBootTest.WebEnvironment.*DEFINED\_PORT*)  
@CucumberContextConfiguration  
public class SpringBootRunCucumberTests {  
 @DataTableType  
 public User convertToUser(Map<String,String> user){  
 return User.*builder*()  
 .firstName(user.get("firstName"))  
 .lastName(user.get("lastName"))  
 .age(Integer.*valueOf*(user.get("age")))  
 .build();  
 }  
}

Si UserController pentru tests(se va crea un bean pentru clasa asta, de aceea merge @Autowired)

public class UserController {  
 private List<User> expected;  
 private List<User> response;  
 @Autowired  
 private TestRestTemplate testRestTemplate;  
 @When("^A user uses (.\*) endpoint$")  
 public void setReturnedUsersList(String response) throws Exception {  
 this.response = List.*of*(testRestTemplate.getForObject(response,User[].class));  
 System.*out*.println(this.response);  
 }  
 @Then("It should receive this data")  
 public void compareResults(List<User> expected){  
 this.expected = expected;  
 System.*out*.println(this.expected);  
  
 Assertions.*assertIterableEquals*(this.expected,this.response);  
 }  
  
}

**E mai bine sa punem ^ la inceput si $ la final si (.\*) sau (.+) langa ele ori prin mijloc**

feature1:

Feature: Test User Controller  
  
 Scenario: A user accesses the /users endpoint  
 When A user uses /users endpoint  
 Then It should receive this data  
 | firstName | lastName | age |  
 | Mititiuc | Eduard | 21 |  
 | Goncear | Alex | 21 |  
 | Turcan | Petru | 20 |  
  
 Scenario: A user accesses the /users/21 endpoint  
 When A user uses /users/21 endpoint  
 Then It should receive this data  
 | firstName | lastName | age |  
 | Mititiuc | Eduard | 21 |  
 | Goncear | Alex | 21 |

**Acum, desi avem 2 scenarii, avem doar un singur when() si then(), dar vedem ca avem 2 teste, caci sunt mai multe scenarii ce corespund la when() si then()**



**Background**

Feature: Test User Controller  
  
 Background: Initialize lists  
 Given Our Lists are not null  
  
 Scenario: A user accesses the /users endpoint  
 When A user uses /users endpoint  
 Then It should receive this data  
 | firstName | lastName | age |  
 | Mititiuc | Eduard | 21 |  
 | Goncear | Alex | 21 |  
 | Turcan | Petru | 20 |  
  
 Scenario: A user accesses the /users/21 endpoint  
 When A user uses /users/21 endpoint  
 Then It should receive this data  
 | firstName | lastName | age |  
 | Mititiuc | Eduard | 21 |  
 | Goncear | Alex | 21 |

public class UserController {  
 private List<User> expected;  
 private List<User> response;  
 @Autowired  
 private TestRestTemplate testRestTemplate;  
 @Given("Our Lists are not null")  
 public void initializeLists(){

System.*out*.println("Initialize lists");  
 response = new ArrayList<>();  
 expected = new ArrayList<>();  
 }  
 @When("^A user uses (.\*) endpoint$")  
 public void setReturnedUsersList(String response) throws Exception {  
 this.response = List.*of*(testRestTemplate.getForObject(response,User[].class));  
 }  
 @Then("It should receive this data")  
 public void compareResults(List<User> expected){  
 this.expected = expected;  
  
 Assertions.*assertIterableEquals*(this.expected,this.response);  
 }  
  
}

**Deci, pentru fiecare Scenariu s-au initializat listele si s- a afisat mesajul in consola.**