

Sumar

1. Crearea unui proiect Maven cu Serenity și JUnit în IntelliJ	1
2. Structura proiectului Maven cu Serenity și JUnit.....	4
3. Setarea browser-ului web pentru rularea testelor	5
4. Execuția testelor	5
5. Generarea raportului Serenity pentru testele executate.....	6
1. Varianta 1	6
1. Varianta 2	7
6. Vizualizarea raportului Serenity	7
7. Data Driven Testing	7

Lista de Figuri

Figure 1 Crearea unui proiect Maven cu Serenity și JUnit	2
Figure 2 Adăugarea tipului de proiect Serenity cu JUnit în lista de tipuri de proiecte Serenity	2
Figure 3. Completarea numelui pachetului root și a numelui proiectului Serenity	3
Figure 4. Vizualizarea și modificarea configurărilor proiectului Maven.....	3
Figure 5. Alegerea folderului proiectului Maven	3
Figure 6. Alegerea opțiunii Enable Auto-Import	4
Figure 7. Structura proiectului Maven cu Serenity și JUnit	4
Figure 8. Setarea implicită a browser-ului web Mozilla Firefox	5
Figure 9. Setarea în variabila <i>Path</i> a folderului care conține driverele pentru browser-ele web folosite la testare	5
Figure 10. Fereastra Maven Projects și comenzile Maven	6
Figure 11. Fereastra Maven Projects și opțiunea Execute Maven Goal - generarea raportului Serenity.....	6
Figure 12. fereastra de comenzi Maven pentru generarea raportului de testare	6
Figure 13. Fereastra Terminal cu execuția comenzii de generare a raportului de testare	7
Figure 14 Vizualizarea raportului Serenity	7
Figure 15 Crearea fișierului cu date de test	8
Figure 16 Clasă de test parametrizată, folosind un fișier .csv.....	8
Figure 17 Raportul Serenity pentru testele din fișierul .csv	8

1. Crearea unui proiect Maven cu Serenity și JUnit în IntelliJ

1. în meniul **File** ---> **New** ---> **Project**;
2. se selectează din tipul de proiect **Maven**;
3. se alege o versiune un SDK, se recomandă **versiunea 11.0.4**;

4. se bifează opțiunea **Create from archetype**;
5. se alege din lista de tipuri de proiecte **serenity-junit-archetype** (vezi Figure 1) versiunea **1.8.4**, apoi **Next**;
6. dacă tipul de proiect nu se găsește în listă, se va adăuga prin click pe butonul **Add Archetype...** și se completează (vezi Figure 2):
 - **GroupId**: **net.serenity-bdd**
 - **ArtifactId**: **serenity-junit-archetype**
 - **Version**: **1.8.4**
 apoi **OK**;
7. Lista de proiecte Maven bazate pe Serenity poate fi accesată de la adresa web: <https://mvnrepository.com/artifact/net.serenity-bdd>.

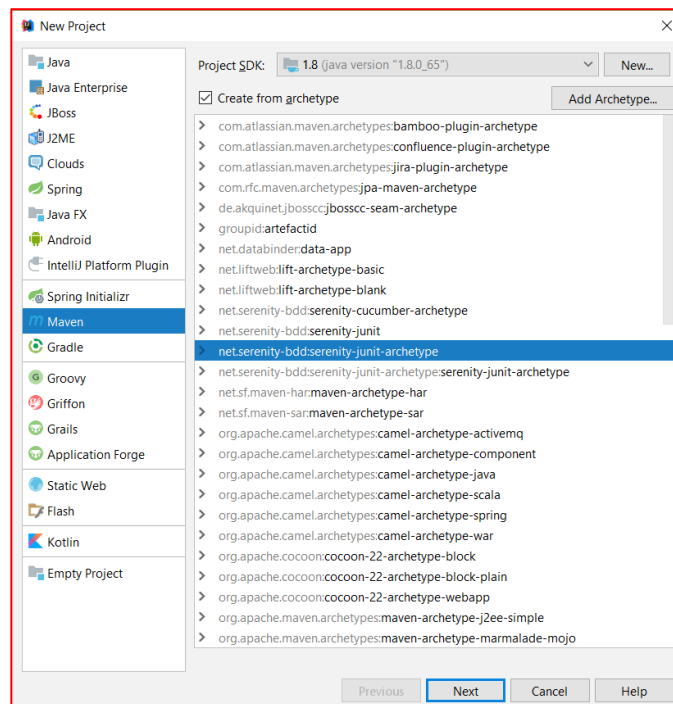


Figure 1 Crearea unui proiect Maven cu Serenity și JUnit

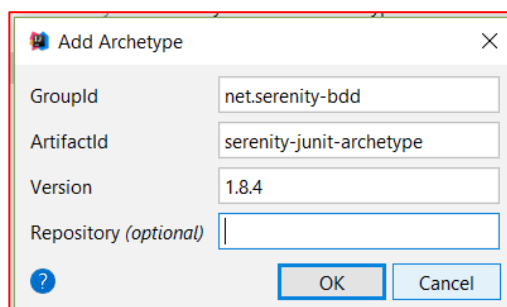


Figure 2 Adăugarea tipului de proiect Serenity cu JUnit în lista de tipuri de proiecte Serenity

8. se completează **GroupId**, i.e., numele pachetului root (vezi Figure 3);
9. se completează **ArtifactId**, i.e., numele proiectului creat (vezi Figure 3), apoi **Next**;

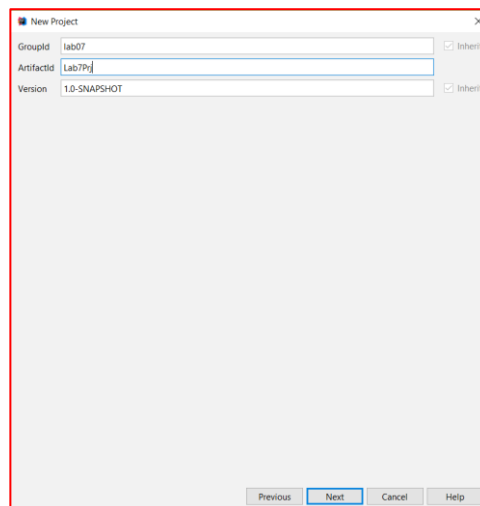


Figure 3. Completarea numelui pachetului root și a numelui proiectului Serenity

10. se poate configura repository-ul implicit și pot vizualiza configurările realizate (vezi Figure 4), apoi **Next**;

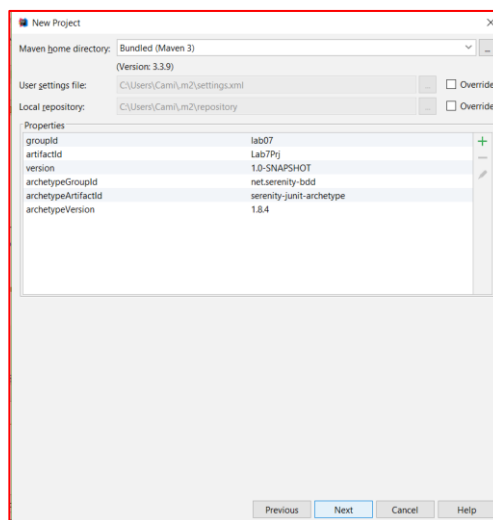


Figure 4. Vizualizarea și modificarea configurărilor proiectului Maven

11. se completează numele folderului în care proiectul va fi creat (vezi Figure 5), apoi **Finish** pentru a finaliza crearea proiectului Maven.

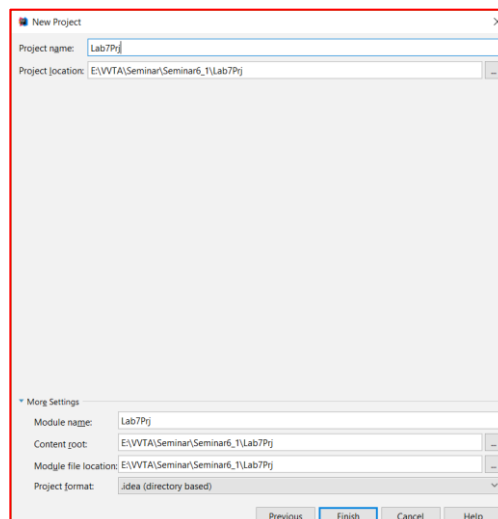


Figure 5. Alegerea folderului proiectului Maven

2. Structura proiectului Maven cu Serenity și JUnit

1. după crearea proiectului, se bifează opțiunea **Enable Auto-Import** (vezi Figure 6);

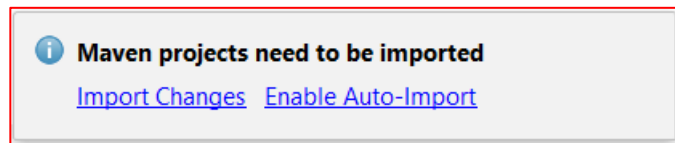


Figure 6. Alegerea opțiunii Enable Auto-Import

2. după creare, proiectul are o structură similară cu a unui proiect Maven care conține doar pachetul **test/java** (vezi Figure 7);
3. particularitățile proiectului Maven Serenity cu JUnit sunt reprezentate de:
 - pachetul **main/java** nu este generat deoarece aplicația care urmează să fie testată este accesată prin intermediul unei adrese web, așadar nu este necesar să adăugăm cod sursă într-un asemenea pachet;
 - pachetul **test/java** conține câteva subpachete predefinite, i.e., **features/search**, **pages**, **steps**, și clase generate care implementează șablonul pentru testarea aplicațiilor web **Pages-Steps-Tests**;

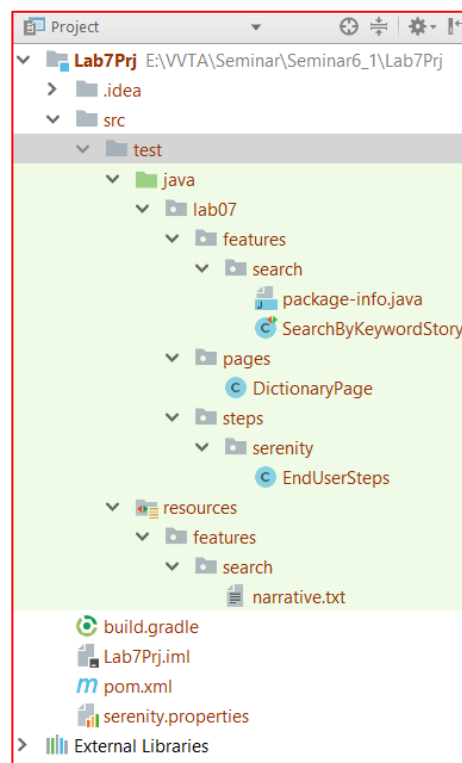


Figure 7. Structura proiectului Maven cu Serenity și JUnit

4. în fișierul **pom.xml** se adaugă în secțiunea `<dependencies>...</dependencies>` următoarea dependență:

```
<dependency>
  <groupId>javax.xml.bind</groupId>
  <artifactId>jaxb-api</artifactId>
  <version>2.3.0</version>
</dependency>

<dependency>
  <groupId>com.sun.xml.bind</groupId>
  <artifactId>jaxb-impl</artifactId>
  <version>2.3.0</version>
</dependency>
```

```

<dependency>
  <groupId>javax.activation</groupId>
  <artifactId>activation</artifactId>
  <version>1.1.1</version>
</dependency>

<dependency>
  <groupId>com.sun.xml.bind</groupId>
  <artifactId>jaxb-core</artifactId>
  <version>2.3.0</version>
</dependency>

```

3. Setarea browser-ului web pentru rularea testelor

1. la crearea proiectului Maven cu Serenity și JUnit, browser-ul web **Mozilla Firefox** este setat ca browser implicit în fișierul **pom.xml**, i.e., **firefox** (vezi Figure 8);
2. driverele pentru browser-ele web folosite la testare:
 - **Firefox:**
 - adresa web: <https://github.com/mozilla/geckodriver/releases>;
 - **Chrome:**
 - adresa web: <https://sites.google.com/a/chromium.org/chromedriver/downloads>;
3. driverele se descarcă, se dezarchivează și se salvează într-un folder, e.g., **c:\drivers**, de unde pot fi folosite ulterior de orice proiect de testare;
4. în variabila de mediu **Path** se adaugă calea către folderul care conține driverele pentru browser-ele web, i.e., **c:\drivers** (vezi Figure 9), **apoi se restartează sistemul de operare**;
5. pentru modificarea browser-ului folosit la rularea testelor, se modifică în fișierul **pom.xml** tipul driverului, i.e., **chrome** sau **firefox**;

```

<name>Serenity project with JUnit and WebDriver</name>

<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <serenity.version>1.8.3</serenity.version>
  <webdriver.driver>firefox</webdriver.driver>
</properties>

```

Figure 8. Setarea implicită a browser-ului web Mozilla Firefox

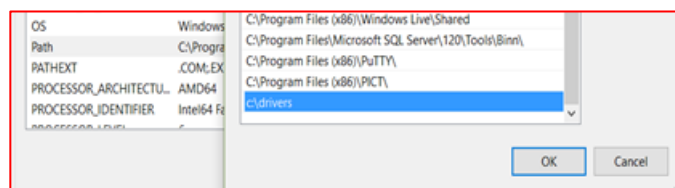


Figure 9. Setarea în variabila *Path* a folderului care conține driverele pentru browser-ele web folosite la testare

4. Execuția testelor

1. click dreapta pe un test sau o clasă de test în **Project Explorer**, e.g., **SearchByKeywordStory**;
2. se selectează **Run SearchByKeywordStory**;

5. Generarea raportului *Serenity* pentru testele executate

1. Varianta 1

1. execuția testelor nu implică și generarea raportului de testare;
2. din meniul **View** ---> **Tool Windows** ---> **Maven Projects** se deschide fereastra proiectelor gestionate cu Maven (vezi Figure 10);
3. se alege opțiunea **verify**, pentru rularea testelor;
4. din meniul ferestrei proiectelor gestionate cu Maven se alege opțiunea **Execute Maven Goal** (vezi Figure 11);
5. în fereastra de comenzi Maven se completează comanda `mvn serenity:aggregate` , apoi **Execute** (vezi Figure 12);
6. raportul generat va fi salvat în folderul proiectului în `\target\site\serenity`;

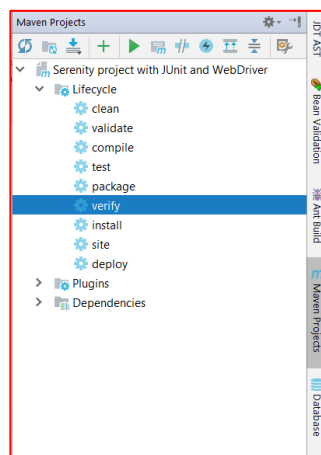


Figure 10. Fereastra Maven Projects și comenzile Maven

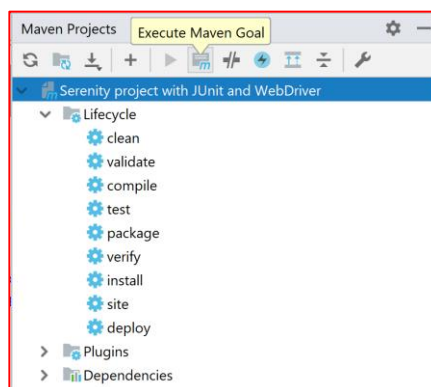


Figure 11. Fereastra Maven Projects și opțiunea Execute Maven Goal - generarea raportului Serenity

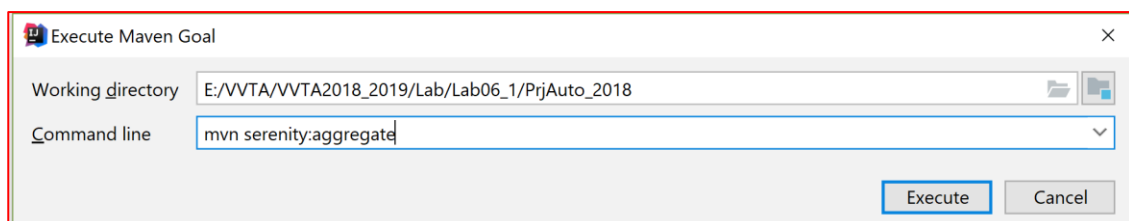


Figure 12. fereastra de comenzi Maven pentru generarea raportului de testare

1. Varianta 2

1. **presupune ca Maven sa fie instalat local** și folderul în care Maven este instalat să fie inclus în variabila de mediu **Path**;
2. din meniul **View** ---> **Tool Windows** ---> **Terminal** se deschide fereastra care permite execuția comenzilor din linia de comandă;
3. în această fereastră se execută comanda Maven:
`mvn serenity: aggregate` <enter> (vezi Figure 13);
4. raportul generat va fi salvat în folderul proiectului în `\target\site\serenity`;

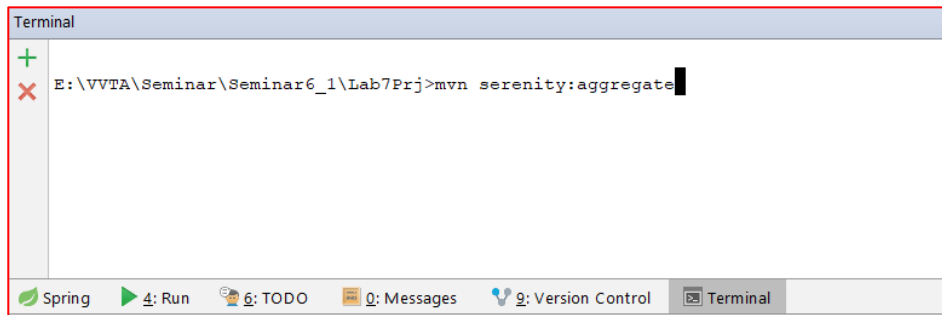


Figure 13. Fereastra Terminal cu execuția comenzii de generare a raportului de testare

6. Vizualizarea raportului Serenity

- din folderul `\target\site\serenity` se încarcă într-un browser web fișierul `index.html` (vezi Figure 14).



Figure 14 Vizualizarea raportului Serenity

7. Data Driven Testing

1. se creează fișierul **WikiTestData.csv** cu date de test (vezi Figure 15);
2. prima linie din fișier indică structura tabelului cu date de intrare;
3. următoarele linii conțin date de intrare pentru cazuri de testare individuale;

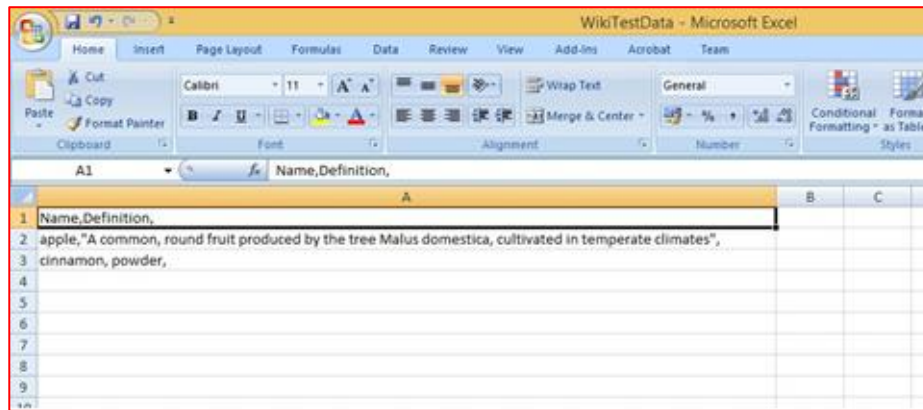


Figure 15 Crearea fișierului cu date de test

4. fișierul **WikiTestData.csv** se salvează în folderul **src/test/resources**;
5. clasa de test va fi executată folosind un *runner parametrizat*;
6. clasa de test reprezintă un șablon de test care va fi folosit pentru execuția fiecărui test pentru care se vor prelua datele de intrare din fiecare linie a fișierului .csv, fișierul fiind dat ca parametru (vezi Figure 16);
7. rularea testelor se realizează ca în Secțiunea 4;
8. generarea raportului Serenity se realizează ca în Secțiunea 5;
9. vizualizarea raportului Serenity indică execuția aceluiși test cu date de intrare diferite, preluate din fișierul .csv, dat ca parametru (vezi Figure 17).

```

20
21 @RunWith(SerenityParameterizedRunner.class)
22 @UseTestDataFrom("src/test/resources/WikiTestData.csv")
23 public class SearchByKeywordStoryDdt {
24
25     @Managed(uniqueSession = true)
26     public WebDriver webDriver;
27
28     @ManagedPages(defaultUrl = "http://en.wiktionary.org/wiki/wiktionary")
29     public Pages pages;
30
31     public String name;
32     public String definition;
33
34     @Qualifier
35     public String getQualifier() {
36         return name;
37     }
38
39     @Steps
40     public EndUserSteps endUser;
41
42     @Issue("#WIKI-1")
43     @Test
44     public void searchWikiByKeywordTestDDT() {
45         endUser.is_the_home_page();
46         endUser.looks_for(getName());
47         endUser.should_see_definition(getDefinition());
48     }

```

Figure 16 Clasă de test parametrizată, folosind un fișier .csv

Steps	Screenshot	Outcome	Duration
Google search test DDT		SUCCESS	0s
11 In: Name=apple, Definition=a common, round fruit produced by the tree Malus domestica, cultivated in temperate climates		SUCCESS	0s
12 Navigate to: http://www.google.com		SUCCESS	760s
13 Look for: def apple		SUCCESS	1.73s
14 Search for keyword: def apple		SUCCESS	0.5s
15 Click on search		SUCCESS	0.08s
16 Verify definition: a common, round fruit produced by the tree Malus domestica, cultivated in temperate climates		SUCCESS	0.06s
Google search test DDT		SUCCESS	0s
11 In: Name=cinnamon, Definition=an aromatic spice made from the peeled, dried, and rolled bark of a SE Asian tree		SUCCESS	0s
12 Navigate to: http://www.google.com		SUCCESS	718s
13 Look for: def cinnamon		SUCCESS	1.77s
14 Search for keyword: def cinnamon		SUCCESS	0.53s
15 Click on search		SUCCESS	0.06s
16 Verify definition: an aromatic spice made from the peeled, dried, and rolled bark of a SE Asian tree		SUCCESS	0.24s

Figure 17 Raportul Serenity pentru testele din fișierul .csv