

Selenium WebDriver

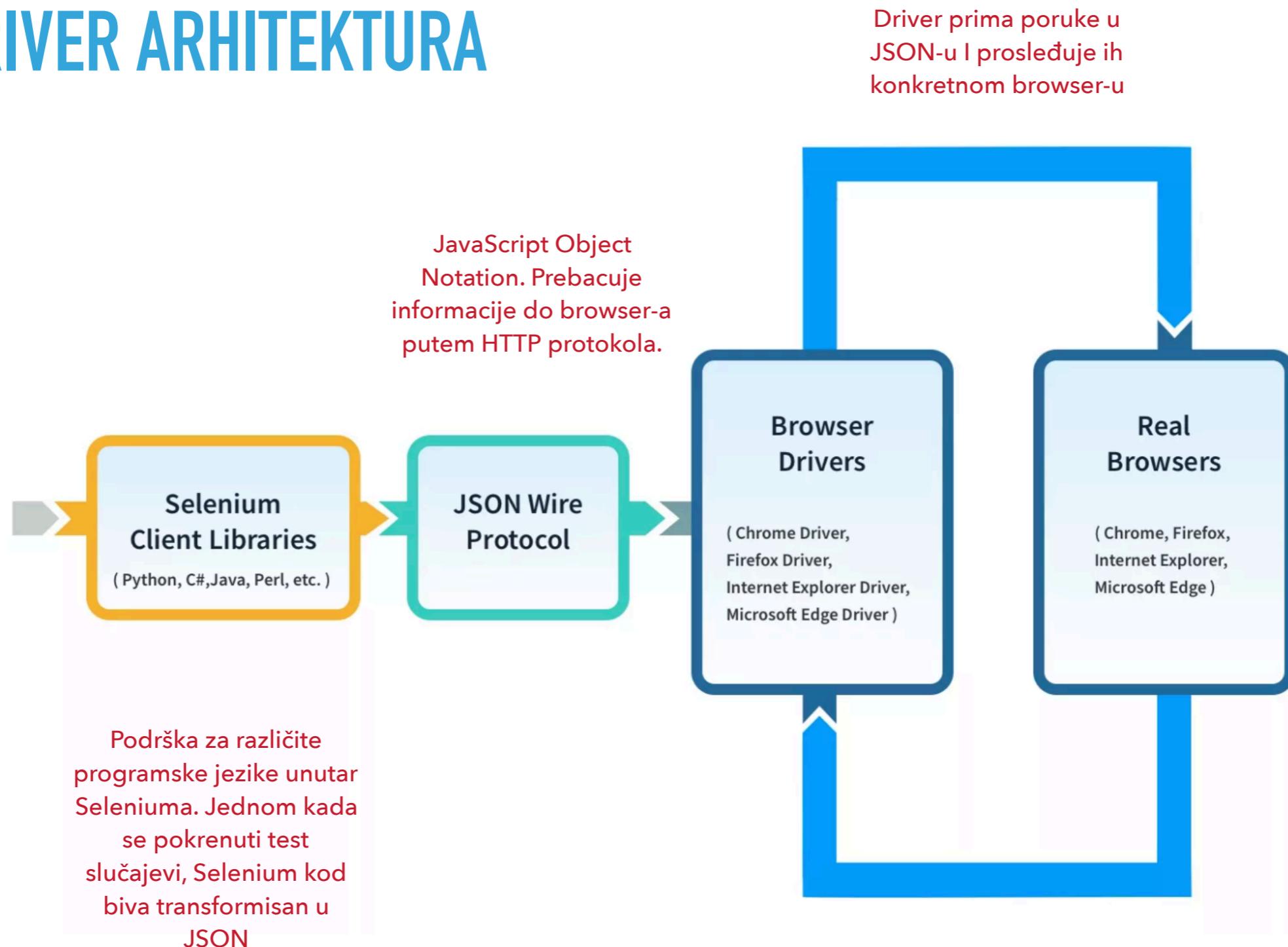
SADRŽAJ

- ▶ WebDriver
- ▶ Komande
- ▶ Lokatori
- ▶ Interakcije
- ▶ DB Dump

WEBDRIVER

- ▶ Selenium Web Driver predstavlja jednu od najvažnijih komponenti Selenium Tool Suite-a
- ▶ Poznatiji je pod nazivom Selenium 2.0
- ▶ Poseduje ugrađenu podršku za Firefox, dok je za druge browser-e potrebno omogućiti odgovarajuće driver-e
- ▶ Selenium Web Driver API omogućava komunikaciju između programskog jezika i browser-a
 - ▶ Java
 - ▶ .Net
 - ▶ PHP
 - ▶ Python
 - ▶ Perl
 - ▶ Ruby
- ▶ Driver za [Chrome](#)
- ▶ Driver za [Mozzilu](#)

WEBDRIVER ARHITEKTURA



OSNOVNE KARAKTERISTIKE

- ▶ Osnovne karakteristike Selenium Web Driver-a:
 - ▶ Omogućava podršku za različite web browser-e
 - ▶ Chrome, Firefox, Opera, Safari, Internet Explorer...
 - ▶ Omogućava podršku za različite programske jezike
 - ▶ Windows, Mac, Linux, Unix...
 - ▶ Brzina (obraća se direktno browser-u)
 - ▶ Jednostavne komande

KOMANDE

Komanda

Opis

driver.get("URL")

To navigate to an application.

element.sendKeys("inputtext")

Enter some text into an input box.

element.clear()

Clear the contents from the input box.

select.deselectAll()

Deselect all OPTIONS from the first SELECT on the page.

select.selectByVisibleText("some text")

Select the OPTION with the input specified by the user.

driver.switchTo().window("windowName")

Move the focus from one window to another.

driver.switchTo().frame("frameName")

Switch from frame to frame.

driver.switchTo().alert()

Helps in handling alerts.

driver.navigate().to("URL")

Navigate to the URL.

driver.navigate().forward()

To navigate forward.

driver.navigate().back()

To navigate back.

driver.close()

Closes the current browser associated with the driver.

driver.quit()

Quits the driver and closes all the associated window of that driver.

driver.refresh()

Refreshes the current page.

LOKATORI

- ▶ Jedna od najvažnijih stvari u korišćenju WebDriver-a je pronalaženje elemenata na stranici
- ▶ Lociranje jednog elementa

```
WebElement cheese = driver.findElement(By.id("cheese"));
```

- ▶ Pronalaženje elementara se vrši pomoću instance **WebDriver** objekta
- ▶ Metoda **findElement(By)** vraća rezultat u vidu **WebElement** objekta
- ▶ Baca izuzetak ako ne pronađe element za zadati kriterijum
- ▶ By interface nudi veliki broj strategija za pronalaz elemenata
- ▶ Lociranje više elemenata

```
List<WebElement> muchoCheese = driver.findElements(By.cssSelector("#cheese li"));
```

- ▶ Metoda findElements(By) vraća rezultat u vidu liste WebElement objekata
- ▶ Ako ne postoji elementi koji zadovoljavaju kriteriju, vraća praznu listu

LOKATORI

Locator	Description
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
css selector	Locates elements matching a CSS selector
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
tag name	Locates elements whose tag name matches the search value
xpath	Locates elements matching an XPath expression

LOKATORI

- ▶ Lociranje na osnovu ID-a

```
<a id="change-password" class="btn" href="#">Forgotten Password</a>
```

```
driver.findElement(By.id("change-password"));
```

- ▶ Lociranje na osnovu NAME atributa

```
<input class="form-control" name="email" type="text" placeholder="Email" />
```

```
driver.findElement(By.name("email"));
```

LOKATORI

- ▶ Lociranje na osnovu naziva taga

```
<h1>Welcome to SDET World!</h1>
```

```
driver.findElement(By.tagName("h1"));
```

- ▶ Lociranje na osnovu CSS selektora

```
<a id="change-password" class="btn" href=".//some-link">Forgotten Password</a>
```

```
driver.findElement(By.cssSelector("#change-password"));
```

LOKATORI

- ▶ Lociranje na osnovu XPath-a

```
<a id="change-password" href="#">Change Password</a>
```

```
driver.findElement(By.xpath("//a[@id='change-password']"));
```

- ▶ Lociranje na osnovu klase

```
<a id="change-password" class="btn" href=".//some-link">Forgotten Password</a>
```

```
driver.findElement(By.className("btn"));
```

LOKATORI

- ▶ Lociranje na osnovu teksta linka

```
<a id="change-password" class="btn" href=".//some-link">Forgotten Password</a>
```

```
driver.findElement(By.linkText("Forgotten Password"));
```

- ▶ Lociranje na osnovu delimičnog teksta linka

```
<a id="change-password" class="btn" href=".//some-link">Forgotten Password</a>
```

```
driver.findElement(By.partialLinkText("Forgotten "));
```

LOKATORI ZADATAK

- ▶ Pogledati primer LocatorsExample.java
- ▶ TODO:

Calculator.net

FINANCIAL FITNESS & HEALTH MATH OTHER

home / math / percentage calculator

Percentage Calculator

Please provide any two values below and click the "Calculate" button to get the third value.

% of =
Calculate **Clear**

Percentage Calculator in Common Phrases

what is

% of

Calculate

_____ is what % of _____

Calculate

_____ is _____ % of what

Calculate

Percentage Difference Calculator

Value 1



Search

Math Calculators

Scientific	Fraction
Percentage	Triangle
Volume	Standard Deviation
Random Number Generator	More Math Calculators

[Financial](#) | [Fitness and Health](#) | [Math](#) | [Other](#)

1. Nakon ispisa vrednosti na konzolu obrisati unete vrednosti iz forme
2. Dodati asertaciju kojom ćete potvrditi da kalkulator radi ispravno.
3. Kreirati test slučaj sa praznim poljima forme i uz njega odgovarajuću asertaciju
 - ▶ Kako biste reorganizovali kod tako da imamo što manje redundantnosti?
4. Ispisati na konzolu nazive svih drugih tipova kalkulatora koji se nalaze na stranici
 - ▶ Da li možemo iskoristiti for each petlju?

INTERAKCIJA

- ▶ Selenium WebDriver nam omogućava direktnu interakciju sa elementima HTML stranice

1. Text Box Interaction
 2. Radio Button Selection
 3. Drop Down Item Selection
 4. Check Box Selection
 5. Drag & Drop
 6. Keyboard Actions
 7. Mouse Actions
 8. Multi Select
 9. Find All Links
-
- { InteractionExample1.java
- } InteractionExample2.java
- } InteractionExample3.java
- } InteractionExample7.java
- } InteractionExample6.java
- } InteractionExample4.java
- } InteractionExample5.java

INTERAKCIJA ZADATAK

- ▶ InteractionExample1.java TODO:
 1. Popraviti TextBox interakciju, tako da ne postoji vodeća nula
 2. Dodajte asertacije koje potvrđuju da su izmenjena polja
 3. Popuniti sva polja unutar Deductions & Credits sa vrednošću 100.
- ▶ InteractionExample2.java TODO:
 1. Postaviti Mozilla driver
- ▶ InteractionExample3.java TODO:
 1. Asertacijom potvrdite da je element pomeren
- ▶ InteractionExample4.java TODO:
 1. Asertacijom potvrdite da su selektovana 3 elementa

INTERAKCIJA

- ▶ Keyboard Actions se izvodi pomoću jedne od 3 metode:
 - ▶ sendKeys – simulacija unosa teksta preko tastature. Posebni karakteri koji nisu tekst, takođe mogu biti uneti putem ove metode
 - ▶ pressKey – simulacija pritiskanja karaktera koji nisu tekst. Pod pressKey podrazumevamo korišćenje dugmića kao što su "F1", "F2", "Tab", "Control" itd.
 - ▶ releaseKey – simulacija otpuštanja dugmeta na tastaturi nakon keyPress događaja

```
void sendKeys(java.lang.CharSequence keysToSend)
void pressKey(java.lang.CharSequence keyToPress)
void releaseKey(java.lang.CharSequence keyToRelease)
```

INTERAKCIJA

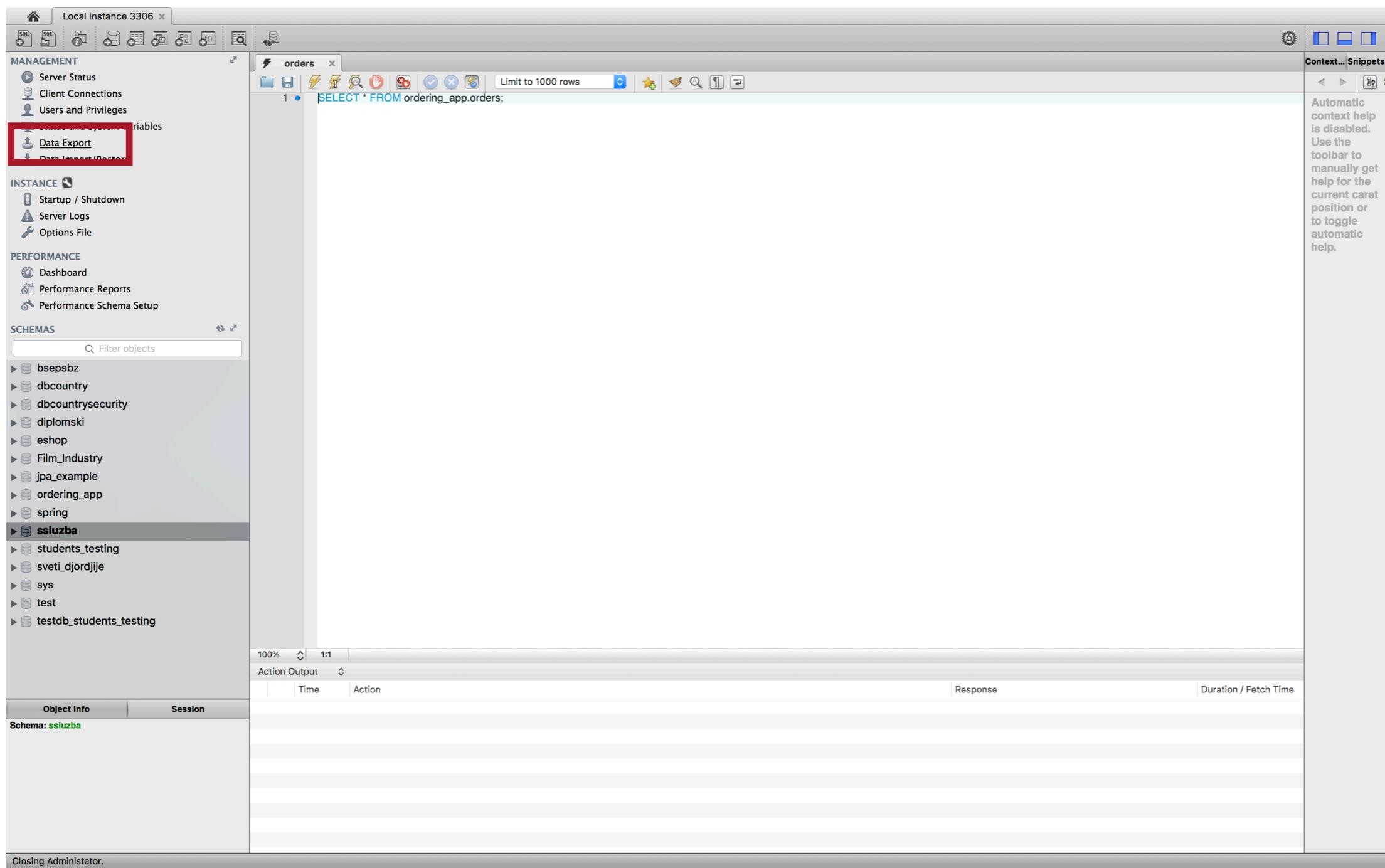
- ▶ Mouse Actions se izvodi pomoću jedne od 6 metoda:
 - ▶ click – simulira klik na element ili na tačno određene koordinate
 - ▶ contextClick – konkteksni, desni klik na element ili na tačno određene koordinate
 - ▶ doubleClick – simulira dvoklik na element ili na tačno određene koordinate
 - ▶ mouseDown – simulira pritisak na taster miša
 - ▶ mouseMove – simulira pomeranje miša
 - ▶ mouseUp – simulira otpuštanje tastera miša

```
void click(WebElement onElement)
void contextClick(WebElement onElement)
void doubleClick(WebElement onElement)
void mouseDown(WebElement onElement)
void mouseUp(WebElement onElement)
void mouseMove(WebElement toElement)
void mouseMove(WebElement toElement, long xOffset, long yOffset)
```

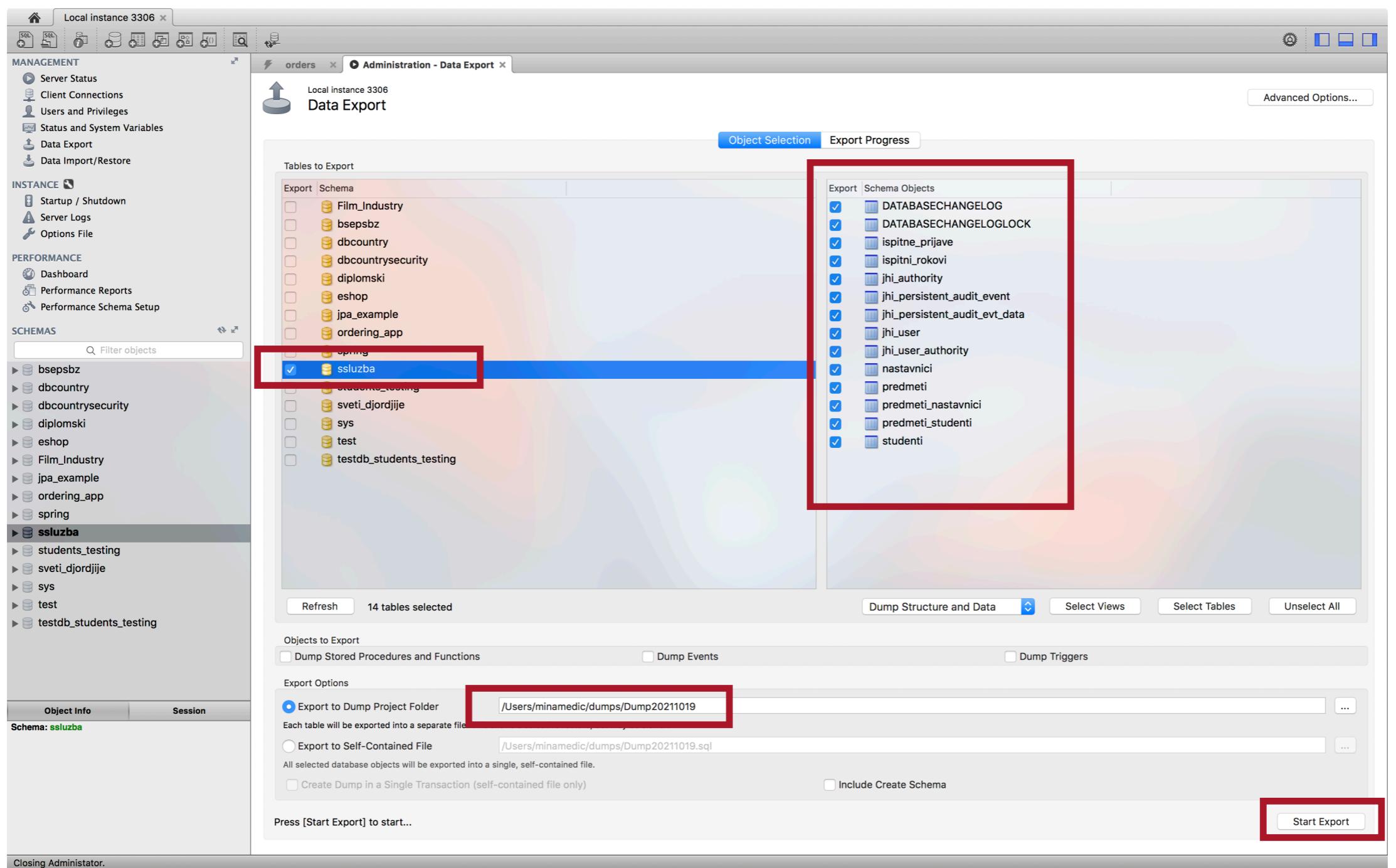
DB SNAPSHOT

- ▶ Pre pokretanja testova potrebno je poznavati trenutno stanje aplikacije (podataka u njoj)
- ▶ Ne poklapanje očekivanog stanja u testovima i same aplikacije dovodi do pucanja testova
- ▶ Testovi menjaju stanje aplikacije, pa je ponekad potrebno vratiti aplikaciju u stanje pre početka izvršavanja testova
- ▶ Dump - backup baze podataka
- ▶ Restore - vraćanje baze u neko od prethodnih stanja

DB DUMP



DB DUMP



DB RESTORE

