

Vaadin

Oliver Laschke

Nikiema Mohamed

Mohammad Sawaya

Matthias Baidinger



Testverfahren

Gate Testbarkeit mittels JUnit
Tool von Vaadin namens TestBench
Alternative dazu ist Selenium

Live-Coding

Codebeispiele

System
Funktion
Form
Tabelle

Einwohlergebiete

Einwohler
Gebiete

Geschichte

- Vaadin: finnisches Wort für weibliches Rentier oder "ich bohre"
- Entstand 2002 als Adapter für das Millstone UI Framework
- Ab 2006 Entwicklung als eigenständiges Projekt
- 2007 Umbenennung zu IT Mill Toolset und Release Ver. 4
- 2009 Umbenennung zu Vaadin und Release Ver. 6
- Aktuelle Version 8 vom 22.2.2017

Gliederung

1. Einleitung
2. Inhalt
3. 1. Semester
4. 2. Semester
5. Anwendungsphase
6. Prüfungsphase
7. Zusammenfassung
8. Literaturverzeichnis

Technik

htw.

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences



Vaadin

Oliver Laschke

Nikiema Mohamed

Mohammad Sawaya

Matthias Baidinger



Testverfahren

Gate Testbarkeit mittels JUnit
Tool von Vaadin namens TestBench
Alternative dazu ist Selenium

Live-Coding

Codebeispiele

System
Funktion
Form
Tabelle

Einwohlergebiete

Einwohler
Gebiete

Geschichte

- Vaadin: finnisches Wort für weibliches Rentier oder "ich bohre"
- Entstand 2002 als Adapter für das Millstone UI Framework
- Ab 2006 Entwicklung als eigenständiges Projekt
- 2007 Umbenennung zu IT Mill ToolKit und Release Ver. 4
- 2009 Umbenennung zu Vaadin und Release Ver. 6
- Aktuelle Version 8 vom 22.2.2017

Gliederung

1. Einleitung
2. Inhalt
3. 1. Semester
4. 2. Semester
5. Anwendungsphase
6. Prüfungsphase
7. Zusammenfassung
8. Literaturverzeichnis

Technik

htw

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences



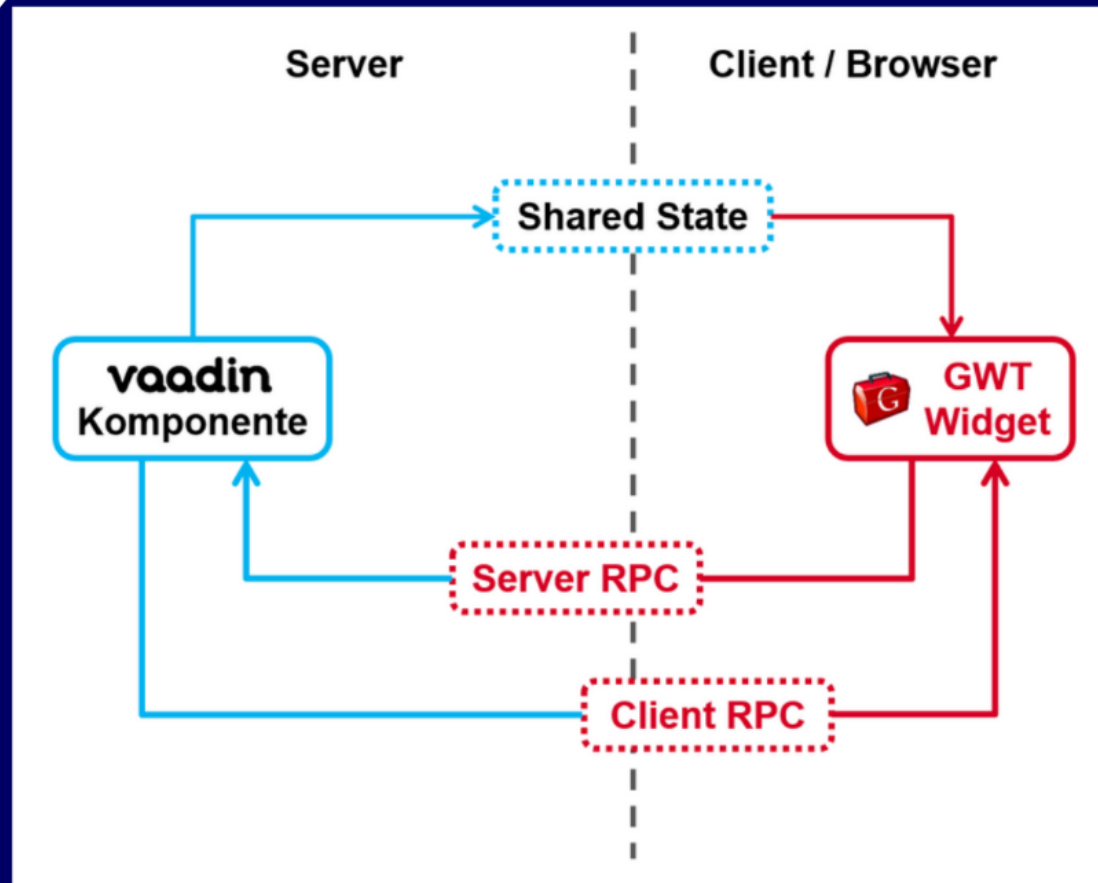
Gliederung

- 1. Geschichte*
- 2. Technik*
 - 2.1 Server*
 - 2.2 Client*
- 3. Einsatzgebiete*
- 4. Codebeispiele*
- 5. Testverfahren*
- 6. Live-Coding*

Geschichte

- *Vaadin: finnisches Wort für weibliches Rentier oder "ich beharre"*
- *Entstand 2002 als Adapter für das Millstone UI Framework*
- *Ab 2006 Entwicklung als eigenständiges Projekt*
- *2007 Umbenennung zu IT Mill Toolkit und Release Ver. 4*
- *2009 Umbenennung zu Vaadin und Release Ver. 6*
- *Aktuelle Version 8 vom 22.2.2017*

Technik



Server

- Erstellen der UI
- Bearbeitet Events mit Listeners
- Verwaltet Bilder und andere Ressourcen
- Fehlerverwaltung
- Benachrichtigungen



Client

GWT - Widgets



Client

GWT - Widgets

Einsatzgebiete

Vorteile

- Geringe Lernkurve
- Entwicklungsgeschwindigkeit
- komplettes Java Ökosystem
- Vielen Veranstaltungen, Communitygröße
- Controls und vorhandene AddOns
- Sehr gut dokumentiert

Nachteile

- Integration weiterer Frameworks
- Implementierung eines Event Handlings
- Schlechte Skalierbarkeit

Vorteile

- Geringe Lernkurve*
- Entwicklungsgeschwindigkeit*
- komplettes Java Ökosystem*
- Vielen Veranstaltungen, Communitygröße*
- Controls und vorhandene AddOns*
- Sehr gut dokumentiert*

Nachteile

- Integration weiterer Frameworks*
- Implementierung eines Event Handlings*
- Schlechte Skalierbarkeit*

Button & Click-Event

```
Button button = new Button("Click");  
button.addActionListener(event ->  
Notification.show("The button was clicked",  
Type.TRAY_NOTIFICATION));
```

TextArea & Listener-Event

```
TextArea txtArea = new TextArea();  
txtArea.setValue("Hier steht ein Text");  
txtArea.setRows(10);  
txtArea.setSizeFull();  
txtArea.addValueChangeListener(event ->  
Notification.show("Text hat sich geändert: ",  
String.valueOf(event.getValue()),  
Type.TRAY_NOTIFICATION));
```

Panel

```
Panel panel = new Panel("Panel Überschrift");  
panel.setHeight(100.0f, Unit.PERCENTAGE);  
final VerticalLayout contentLayout = new VerticalLayout();  
contentLayout.setWidth(500, Unit.PIXELS);  
contentLayout.setSpacing(false);  
contentLayout.addComponent(new Label(ExampleUtil.lorem,  
ContentMode.HTML));  
  
panel.setContent(contentLayout);
```

Table

```
final Window mainWindow = new Window("Tabellentest");

    Table table = new Table("Namensliste");
    table.addContainerProperty("Vorname", String.class, null);
    table.addContainerProperty("Nachname", String.class, null);
    /* Items zur Tabelle hinzufuegen*/

    table.addItem(new Object[] {"Vorname1","Nachname1"}, new
Integer(1));
    table.addItem(new Object[] {"Vorname2","Nachname2"}, new
Integer(2));

    mainWindow.addComponent(table);
    setMainWindow(mainWindow);

    Panel panel = new Panel("Panel");
    panel.setWidth("400px"); // width definieren
    panel.addComponent(table);
    mainWindow.addComponent(panel);

    setMainWindow(mainWindow);
```


TestBench

- Automatisiertes UI-Testing
- Basiert auf Selenium
- Tests werden als *Flow* Tests erstellt
- Testklassen erben von der Klasse `com.vaadin.testbench.TestBenchTestCase`
- Kontextpflichtig

```
public class ExampleTestCase extends TestBenchTestCase {
    @Before
    public void before() {
        WebDriver driver = getDriver();
        getDriver().get("http://vaadin.com/");
    }

    @After
    public void tearDown() {
        getDriver().quit();
    }
}

@Test
public void testCaption() {
    WebElement element = $("caption").first();
    assertEquals("Caption", element.getText());
}

@Test
public void testCaption2() {
    WebElement element = $("caption").first();
    assertEquals("Caption", element.getText());
}
```

Testverfahren

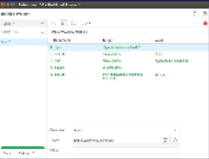
Gute Testbarkeit mittels JUnit

Tool von Vaadin namens TestBench

Alternative dazu ist Selenium

Selenium

- Automatisierte Tests für Webanwendungen
- Interaktion mit Webseiten durch aufrufbarer und beliebig oft wiederholbar



```
WebDriver driver = new ChromeDriver();
driver.get("https://www.bbc.com/");

WebElement searchBox = driver.findElement(By.name("-"));
searchBox.sendKeys("ChromeDriver");
searchBox.submit();

driver.quit();
```

TestBench

- *Automatisiertes UI-Testing*
- *Basiert auf Selenium*
- *Tests werden als JUnit Tests erstellt*
- *Testklassen erben von der Klasse
`com.vaadin.testbench.TestBenchTestCase`*
- *Kostenpflichtig*

```
public class ExampleTestCase extends TestBenchTestCase{
    @Before
    public void before(){
        setDriver(new FirefoxDriver());
        getDriver().get("http://localhost:8080/...");
    }
}
```

```
    @After
    public void teardown(){
        getDriver().quit();
    }
    ...
}
```

```
$(ButtonElement.class).caption("Speichern").first().click();
$(TextFieldElement.class).caption("Vorname").exists();
$(TextFieldElement.class).caption("Vorname").first().setValue
("Hans");
$(TableElement.class).first().getCell(0, 0).getText();
```

Selenium

- *Automatisierte Tests für Webanwendungen*
- *Interaktion mit Webanwendungen aufnehmbar und beliebig oft wiederholbar*

Selenium IDE - Untitled Project*

Untitled Project*

Tests +

Search tests...

test1*

https://www.htw-berlin.de

Command	Target	Value
1. open	https://www.htw-berlin.de/	
2. click at	name=query	30,27
3. type	name=query	Angewandte Informatik
4. submit	id=websuche	
5. click at	link=Angewandte Informatik - Bachelor	254,11

Command: open

Target: https://www.htw-berlin.de/

Value:

Runs: 1 Failures: 0

```
WebDriver driver = new ChromeDriver();  
driver.get("https://www.htw-berlin.de/");
```

```
WebElement searchBox = driver.findElement(By.name("..."));  
searchBox.sendKeys("ChromeDriver");  
searchBox.submit();
```

```
driver.quit();
```



Live-Coding



Vaadin

Oliver Laschke

Nikiema Mohamed

Mohammad Sawaya

Matthias Baidinger



- Gliederung
- 1. Geschichte
 - 2. Testverfahren
 - 3. Codebeispiele
 - 4. Einzugsgebiete
 - 5. Technik

Geschichte

- Vaadin: finnisches Wort für weibliches Rentier oder "ich bohre"
- Entstand 2002 als Adapter für das Millstone UI Framework
- Ab 2006 Entwicklung als eigenständiges Projekt
- 2007 Umbenennung zu IT Mill Toolset und Release Ver. 4
- 2009 Umbenennung zu Vaadin und Release Ver. 6
- Aktuelle Version 8 vom 22.2.2017

Testverfahren

Gate Testbarkeit mittels JUnit
Tool von Vaadin namens TestBench
Alternative dazu ist Selenium

Codebeispiele

- System
- Funktion
- Panel
- Table

Einzugsgebiete

- Java
- JavaScript

Technik

Live-Coding

