Lecture 14

Introduction to the Play Framework

SOEN 6441, Summer 2018

René Witte
Department of Computer Science
and Software Engineering
Concordia University

Outline

René Witte
Concordia University

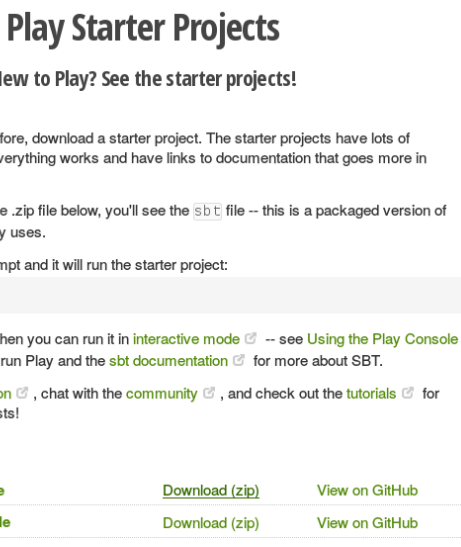
Outline

René Witte

Your First Play Application

Download Starter Application

https://playframework.com/download#starters



Play Starter Projects

New to Play? See the starter projects!

If you've never used Play before, download a starter project. The starter projects have lots of comments explaining how everything works and have links to documentation that goes more in depth.

If you download and unzip the .zip file below, you'll see the `sbt` file -- this is a packaged version of sbt , the build tool that Play uses.

Type the following at the prompt and it will run the starter project:

```
./sbt run
```

If you type `./sbt` by itself, then you can run it in interactive mode -- see Using the Play Console for more detail of how to run Play and the sbt documentation for more about SBT.

Please see the documentation , chat with the community , and check out the tutorials for more examples and blog posts!

Play 2.6.x Starter Projects

Play Java Starter Example	Download (zip)	View on GitHub
Play Scala Starter Example	Download (zip)	View on GitHub

René Witte

Play Starter Application

```
rene@matrix:~/play/play-java-starter-example> ll
total 68
drwxr-xr-x 6 rene users 4096 Feb 15  2018 app
-rw-r--r-- 1 rene users 1050 Feb  1 04:47 build.gradle
-rw-r--r-- 1 rene users  613 Feb  1 04:47 build.sbt
drwxr-xr-x 2 rene users 4096 Feb 15  2018 conf
drwxr-xr-x 3 rene users 4096 Feb 15  2018 gradle
-rw-r--r-- 1 rene users 5296 Feb  1 04:47 gradlew
-rw-r--r-- 1 rene users 2260 Feb  1 04:47 gradlew.bat
-rw-r--r-- 1 rene users  439 Feb  1 04:47 LICENSE
drwxr-xr-x 2 rene users 4096 Feb 15  2018 project
drwxr-xr-x 5 rene users 4096 Feb 15  2018 public
-rw-r--r-- 1 rene users 1453 Feb  1 04:47 README.md
-rwxrw-r-- 1 rene users   44 Jan 15 18:59 sbt
-rwxrw-r-- 1 rene users   55 Jan 15 18:59 sbt.bat
drwxr-xr-x 4 rene users 4096 Feb 15  2018 sbt-dist
drwxr-xr-x 2 rene users 4096 Feb 15  2018 scripts
drwxr-xr-x 2 rene users 4096 Feb 15  2018 test
```

René Witte

Compiling

```
rene@matrix:~/play/play-java-starter-example> ./sbt compile
Getting org.scala-sbt sbt 1.1.0 ...
downloading https://repo1.maven.org/maven2/org/scala-sbt/sbt/
    1.1.0/sbt-1.1.0.jar ...
        [SUCCESSFUL ] org.scala-sbt#sbt;1.1.0!sbt.jar (77ms)
downloading https://repo1.maven.org/maven2/org/scala-lang/
    scala-library/2.12.4/scala-library-2.12.4.jar ...
        [SUCCESSFUL ] org.scala-lang#scala-library;
    2.12.4!scala-library.jar (805ms)

...

[info] Compiling 7 Scala sources and 9 Java sources to /home/rene/
    play/play-java-starter-example/target/scala-2.12/classes ...
[info] Non-compiled module 'compiler-bridge_2.12' for
    Scala 2.12.4. Compiling...
[info]   Compilation completed in 6.189s.
[info] Done compiling.
[success] Total time: 25 s, completed Feb 15, 2018 9:52:01 AM
```

Note

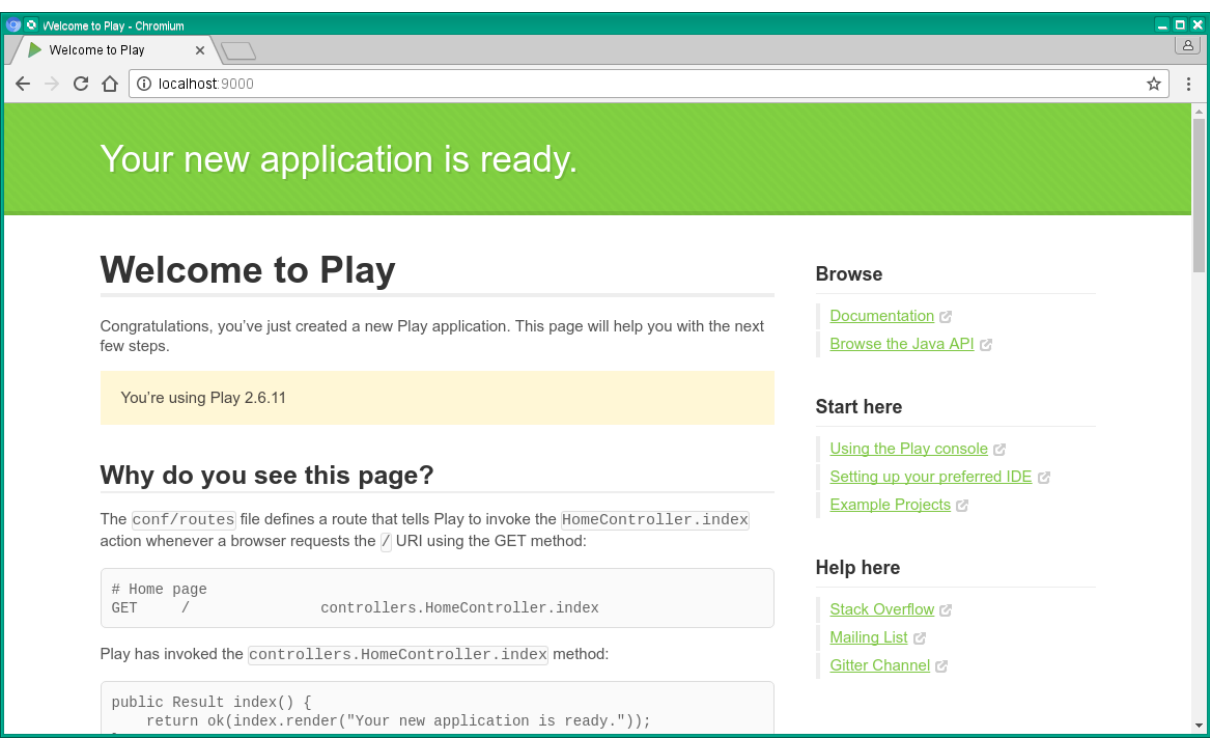
The previously used `activator` command has been replaced by `sbt`

René Witte

Running

René Witte
Concordia University


```
rene@matrix:~/play/play-java-starter-example> ./sbt run
[info] Loading settings from plugins.sbt ...
[info] Loading project definition from /home/rene/play/
    play-java-starter-example/project
[info] Loading settings from build.sbt ...
[info] Set current project to play-java-starter-example (in
    build file:/home/rene/play/play-java-starter-example/)

--- (Running the application, auto-reloading is enabled) ---

[info] p.c.s.AkkaHttpServer - Listening for HTTP on
    /0:0:0:0:0:0:0:0:9000

(Server started, use Enter to stop and go back to the console...)
```


Welcome to Play - Chromium
Welcome to Play
localhost:9000
Your new application is ready.
Welcome to Play
Congratulations, you've just created a new Play application. This page will help you with the next few steps.
You're using Play 2.6.11
Why do you see this page?
The conf/routes file defines a route that tells Play to invoke the HomeController.index action whenever a browser requests the / URI using the GET method:
Home page
GET / controllers.HomeController.index
Play has invoked the controllers.HomeController.index method:
public Result index() {
return ok(index.render("Your new application is ready."));
Browse
Documentation
Browse the Java API
Start here
Using the Play console
Setting up your preferred IDE
Example Projects
Help here
Stack Overflow
Mailing List
Gitter Channel

Creating an Eclipse Project

René Witte
Concordia University

Step 1: Add the sbteclipse Plugin

Open `project/plugins.sbt`, add two lines:

```
// Eclipse plugin, see https://github.com/typesafehub/sbteclipse
addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "5.2.4")
```

Step 2: Generate Eclipse Files

Run `sbt eclipse` (must be done *after* compile!)

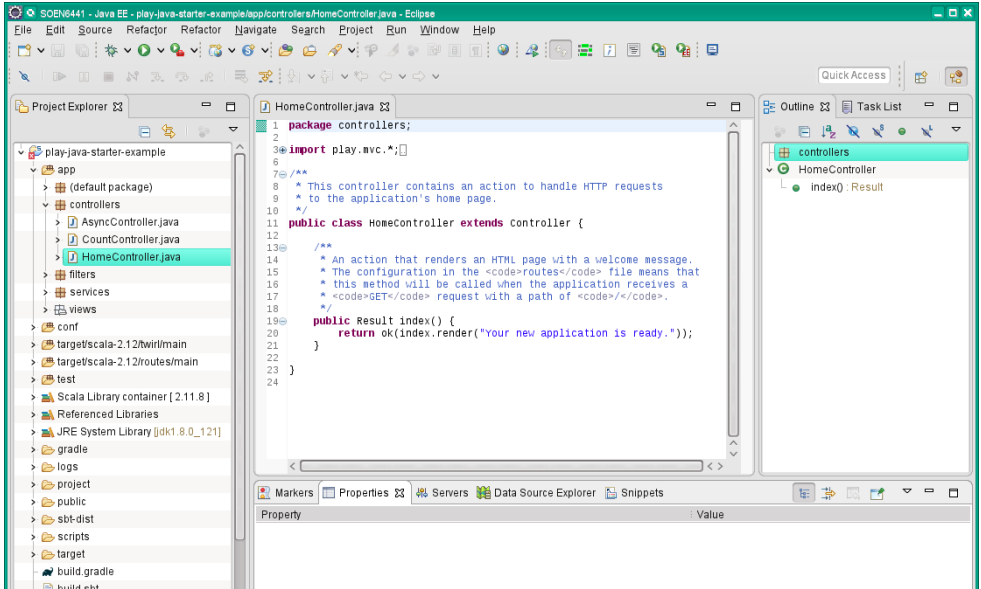
```
rene@matrix:~/play/play-java-starter-example> ./sbt eclipse
[info] Loading settings from plugins.sbt ...
...
[info] About to create Eclipse project files for your project(s).
...
[info] Successfully created Eclipse project files for project(s):
[info] play-java-starter-example
```

See https://www.playframework.com/documentation/2.6.x/IDE

Creating an Eclipse Project (II)

Step 3: Import Project into Eclipse Workspace

Choose *File* → *Import* (General/Existing project); select Play starter project directory



René Witte

Outline

René Witte

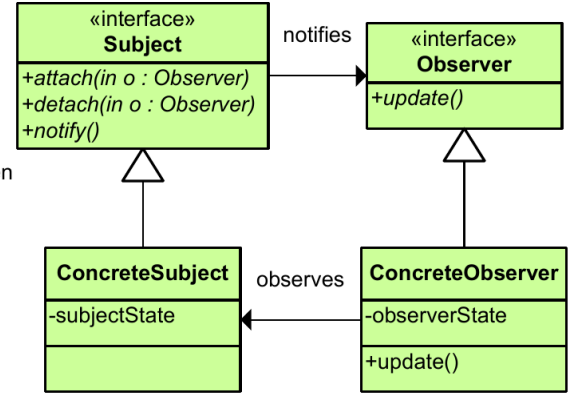
Concordia University

René Witte
Concordia University

Observer

Type: Behavioral

What it is:
Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.



Observer Pattern Implementation

René Witte

`Observer` Interface

```
interface Observer {
  void notify(String tweet);
}
```

Observer Pattern Implementation (II)

Implementing `Observers`

```
class NYTimes implements Observer{
  public void notify(String tweet) {
    if(tweet != null && tweet.contains("money")){
      System.out.println("Breaking␣news␣in␣NY!␣" + tweet);
    }
  }
}
class Guardian implements Observer{
  public void notify(String tweet) {
    if(tweet != null && tweet.contains("queen")){
      System.out.println("Yet␣another␣news␣in␣London...␣" + tweet);
    }
  }
}
class LeMonde implements Observer{
  public void notify(String tweet) {
    if(tweet != null && tweet.contains("wine")){
      System.out.println("Today␣cheese,␣wine␣and␣news!␣" + tweet);
    }
  }
}
```

René Witte

Observer Pattern Implementation (III)

René Witte

`Subject` Interface

```
interface Subject{
  void registerObserver(Observer o);
  void notifyObservers(String tweet);
}
```

Observer Pattern Implementation (IV)

René Witte
Concordia University

`Subject` Implementation

```
class Feed implements Subject{

  private final List<Observer> observers = new ArrayList<>();

  public void registerObserver(Observer o) {
    this.observers.add(o);
  }

  public void notifyObservers(String tweet) {
    observers.forEach(o -> o.notify(tweet));
  }
}
```

Observer Pattern Implementation (V)

René Witte
Concordia University

Demo application

```
Feed f = new Feed();
f.registerObserver(new NYTimes());
f.registerObserver(new Guardian());
f.registerObserver(new LeMonde());
f.notifyObservers("The queen said her favourite course is SOEN6441!");
```

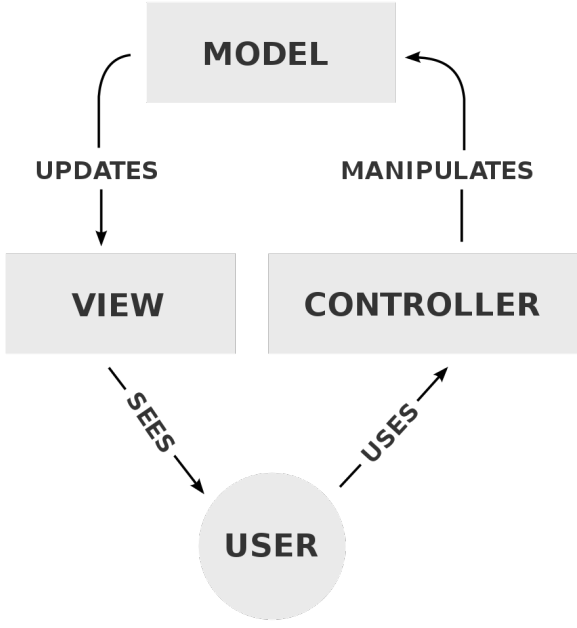
Observer Pattern Implementation (VI)

René Witte
Concordia University

Using lambdas

```
f.registerObserver((String tweet) -> {
  if(tweet != null && tweet.contains("money")){
    System.out.println("Breaking news in NY! " + tweet);
  }
});

f.registerObserver((String tweet) -> {
  if(tweet != null && tweet.contains("queen")){
    System.out.println("Yet another news in London... " + tweet);
  }
});
```

Model-View-Controller (MVC)



René Witte
Concordia University

MVC in Play

René Witte
Concordia University

Model

Application classes, typically under `models/`

- E.g., products in a web shop
- Typically connected to a DB layer (e.g., using Ebean)

(please do not confuse *models* with *modules*, which are used for dependency injection)

View

Views, typically under `views/`

- either application-generated, or through a library
- Play supports view templates (e.g., to show a product in a web shop)

Controller

Controllers, in `controllers/`

- Event-handling (HTTP requests)
- Execute actions that return a result (HTTP response)

View Example

```
@*
 * This template takes a single argument, a String containing a
 * message to display.
 *@
@(message: String)

@*
 * Call the `main` template with two arguments. The first
 * argument is a `String` with the title of the page, the second
 * argument is an `Html` object containing the body of the page.
 *@
@main("Welcome to Play") {
    @*
     * Get an `Html` object by calling the built-in Play welcome
     * template and passing a `String` message.
     *@
    @welcome(message, style = "java")
}
```

René Witte
Concordia University

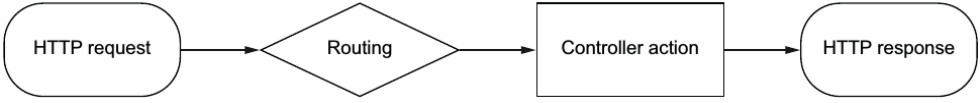
Outline

René Witte

Concordia University

The Request Lifecyle

René Witte

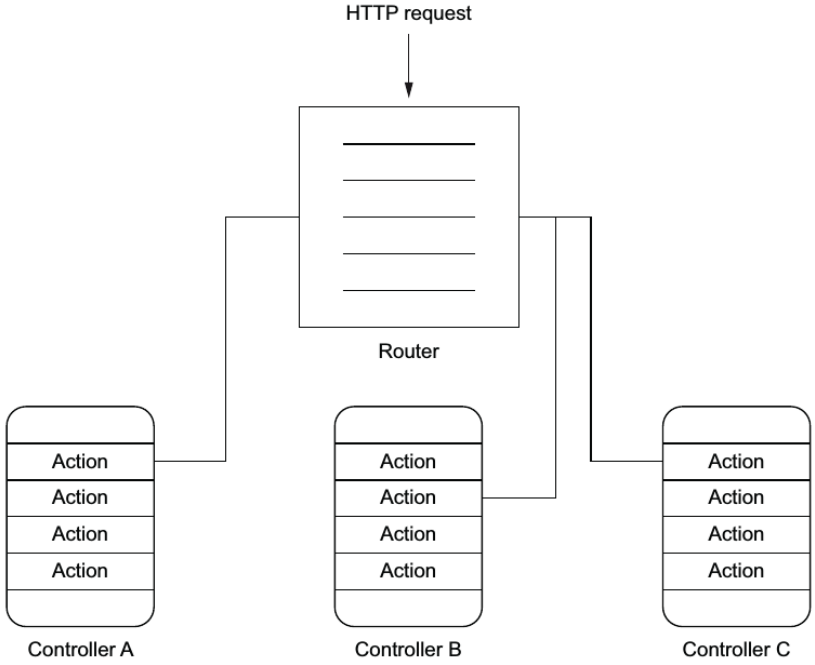


HTTP Request Example

```
GET /welcome HTTP/1.1
Host: localhost
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/36.0.1985.125
Safari/537.36
DNT: 1
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8,de;q=0.6,fr;q=0.4,nl;q=0.2
```

René Witte

Request Routing



René Witte

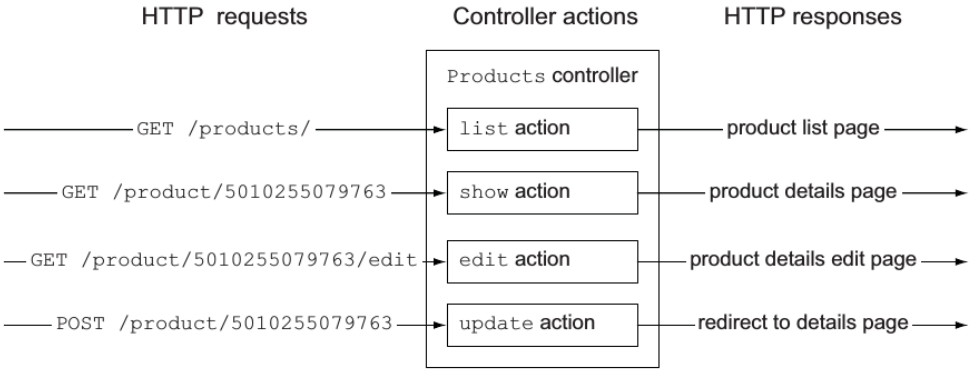
Routes configuration

Defined in `conf/routes`

René Witte
Concordia University


```
# Routes
# This file defines all application routes (Higher priority routes first)
# ~~~~

# An example controller showing a sample home page
GET     /               controllers.HomeController.index

# An example controller showing how to use dependency injection
GET     /count          controllers.CountController.count

# An example controller showing how to write asynchronous code
GET     /message        controllers.AsyncController.message

# Map static resources from the /public folder to the /assets URL path
GET     /assets/*file   controllers.Assets.versioned(path="/public", file: Asset)
```

Controllers



Play Controllers

- Class extending `play.mvc.Controller`
- Contains several action methods
- Returns `play.mvc.Result`

René Witte

Example Controller

```
package controllers;

import play.mvc.*;
import views.html.*;

/**
 * This controller contains an action to handle HTTP requests
 * to the application's home page.
 */
public class HomeController extends Controller {

    /**
     * An action that renders an HTML page with a welcome message.
     * The configuration in the <code>routes</code> file means that
     * this method will be called when the application receives a
     * <code>GET</code> request with a path of <code>/</code>.
     */
    public Result index() {
        return ok(index.render("Your new application is ready."));
    }
}
```

René Witte

Results

René Witte

Helper class `play.mvc.Results`

```
Result ok = ok("Hello world!");

Result notFound = notFound();

Result pageNotFound
    = notFound("<h1>Page not found</h1>").as("text/html");

Result badRequest
    = badRequest(views.html.form.render(formWithErrors));

Result oops = internalServerError("Oops");

Result anyStatus = status(488, "Strange response type");
```

Redirects

```
public Result index() {
    return redirect("/user/home");
}

public Result index() {
    return temporaryRedirect("/user/home");
}
```

René Witte

Working with asynchronous results

René Witte

Using `CompletionStage<Result>`

```
CompletionStage<Double> futurePIValue = computePIAsynchronously();
// thenApply runs in same thread
CompletionStage<Result> futureResult = futurePIValue.thenApply(
    pi -> ok("PI value computed: " + pi)
);
```

Note

Only the `"*Async"` methods from `CompletionStage` provide asynchronous execution (in a different thread).

See https://www.playframework.com/documentation/2.6.x/JavaAsync

Outline

Running Unit Tests

```
rene@matrix:~/play/play-java-starter-example> ./sbt test
[info] Loading settings from plugins.sbt ...
[info] Loading project definition from /home/rene/play/
    play-java-starter-example/project
[info] Loading settings from build.sbt ...
...
[info] Done compiling.
[info] Test run started
[info] Test UnitTest.simpleCheck started
[info] Test UnitTest.testAsync started
[info] Test UnitTest.testCount started
[info] Test run finished: 0 failed, 0 ignored, 3 total, 2.076s
[info] Test run started
[info] Test FunctionalTest.renderTemplate started
...
[info] Test BrowserTest.test started
...
[info] application - ApplicationTimer demo: Stopping application at
    2018-02-15T17:19:19.422Z after 3s.
[info] Test run finished: 0 failed, 0 ignored, 1 total, 2.697s
[info] Passed: Total 5, Failed 0, Errors 0, Passed 5
[success] Total time: 10 s, completed Feb 15, 2018 12:19:19 PM
```

René Witte

Unit Tests

Example: Testing a Controller

```
@Test
public void testCount() {
  final CountController controller = new CountController(() -> 49);
  Result result = controller.count();
  assertThat(contentAsString(result)).isEqualTo("49");
}
```

René Witte
Concordia University

Functional Tests

René Witte
Concordia University


```
public class FunctionalTest extends WithApplication {

  @Test
  public void renderTemplate() {
    Content html =
      views.html.index.render("Your␣new␣application␣is␣ready.");
    assertThat("text/html").isEqualTo(html.contentType());
    assertThat(html.body()).contains("Your␣new␣application␣is␣ready.");
  }
}
```

Integration Tests

René Witte
Concordia University

```
public class BrowserTest extends WithBrowser {

  protected Application provideApplication() {
    return fakeApplication(inMemoryDatabase());
  }

  protected TestBrowser provideBrowser(int port) {
    return Helpers.testBrowser(port);
  }

  /**
  * add your integration test here
  * in this example we just check if the welcome page is being shown
  */
  @Test
  public void test() {
    browser.goTo("http://localhost:" + play.api.test.Helpers.testServerPort());
    assertTrue(browser.pageSource().contains("Your new application is ready."));
  }
}
```

Outline

René Witte
Concordia University

Reading Material

René Witte

Required

- [Ber16, Chapter 4] (Introduction to Play)
- [UFM14, Chapter 8.2.3] (Observer Design Pattern)
- Play Documentation: https://www.playframework.com/documentation/2.6.x/Home

Supplemental

- [GHJV95] (Observer Design Pattern)
- [Ber16, Chapter 2] (Reactive Web Applications)

References

[Ber16] Manuel Bernhardt.
Reactive Web Applications.
Manning Publications, 2016.
https://www.manning.com/books/reactive-web-applications.

[GHJV95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides.
Design Patterns: Elements of Reusable Object-Oriented Software.
Addison-Wesley, 1995.

[LdK14] Nicolas Leroux and Sietse de Kaper.
Play for Java: Covers Play 2.
Manning Publications, 2014.
https://www.manning.com/books/play-for-java.

[UFM14] Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft.
Java 8 in Action: Lambdas, streams, and functional-style programming.
Manning Publications, 2014.
https://www.manning.com/books/java-8-in-action.

René Witte