Play Framework (with Activator)

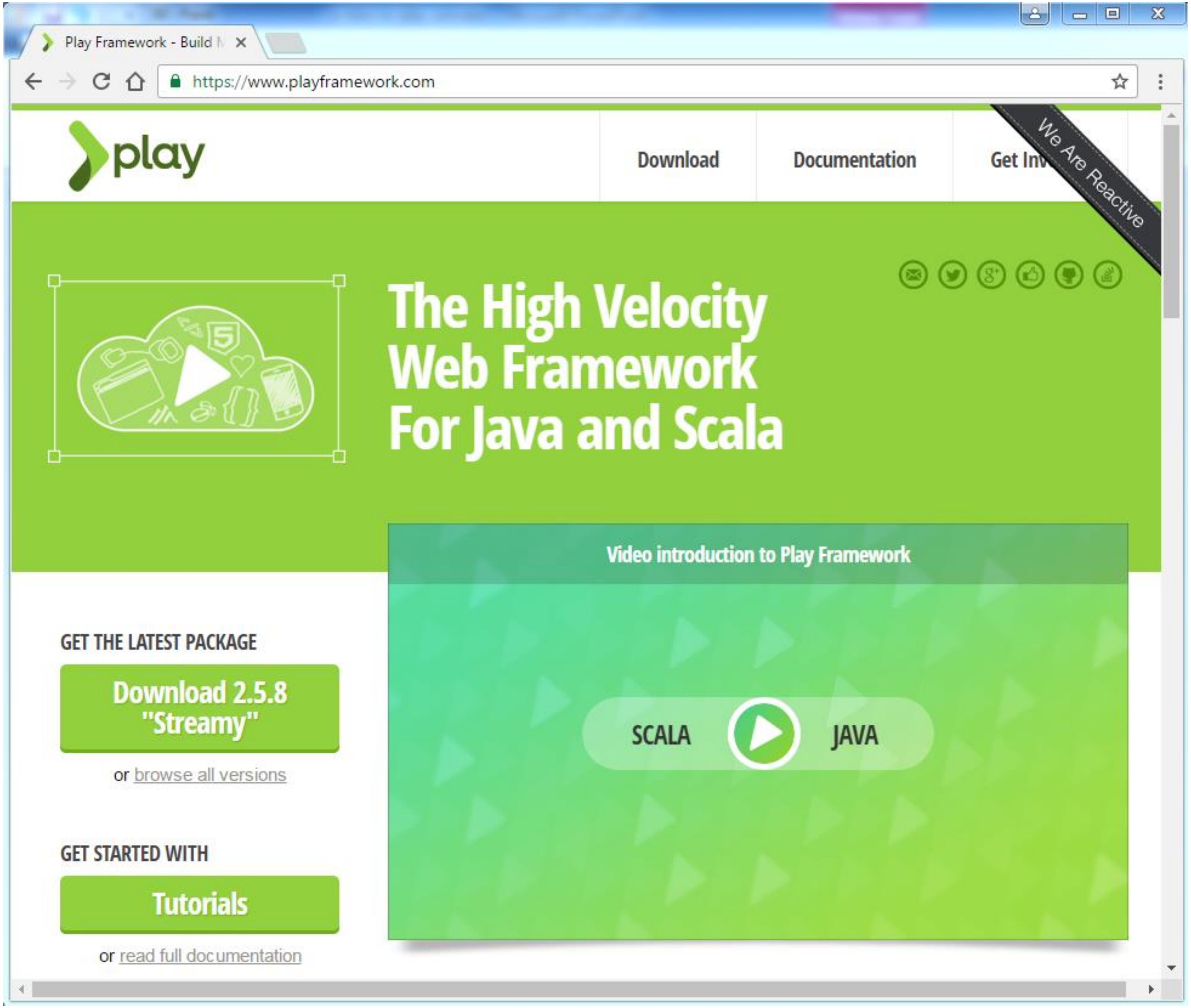
Produced by:

Dr. Siobhán Drohan (sdrohan@wit.ie)

Eamonn de Leastar (edeleastar@wit.ie)

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

Department of Computing and Mathematics
http://www.wit.ie/


Play Framework - Build

https://www.playframework.com

play

Download

Documentation

Get Inv

We Are Reactive

The High Velocity Web Framework For Java and Scala

Video introduction to Play Framework

SCALA

JAVA

GET THE LATEST PACKAGE

Download 2.5.8 "Streamy"

or browse all versions

GET STARTED WITH

Tutorials

or read full documentation

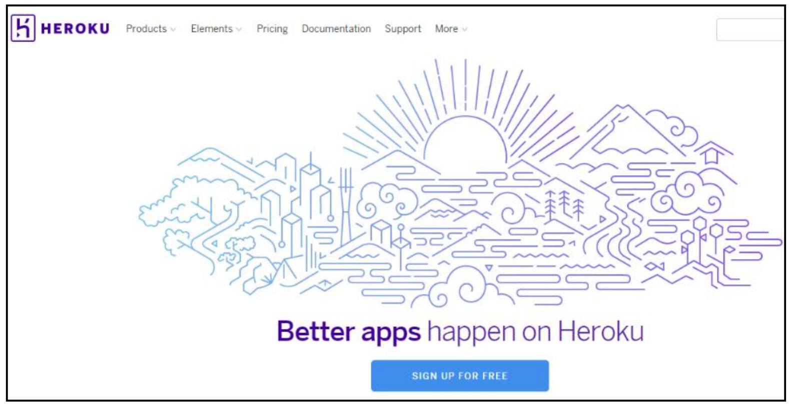
What is the Play Framework?

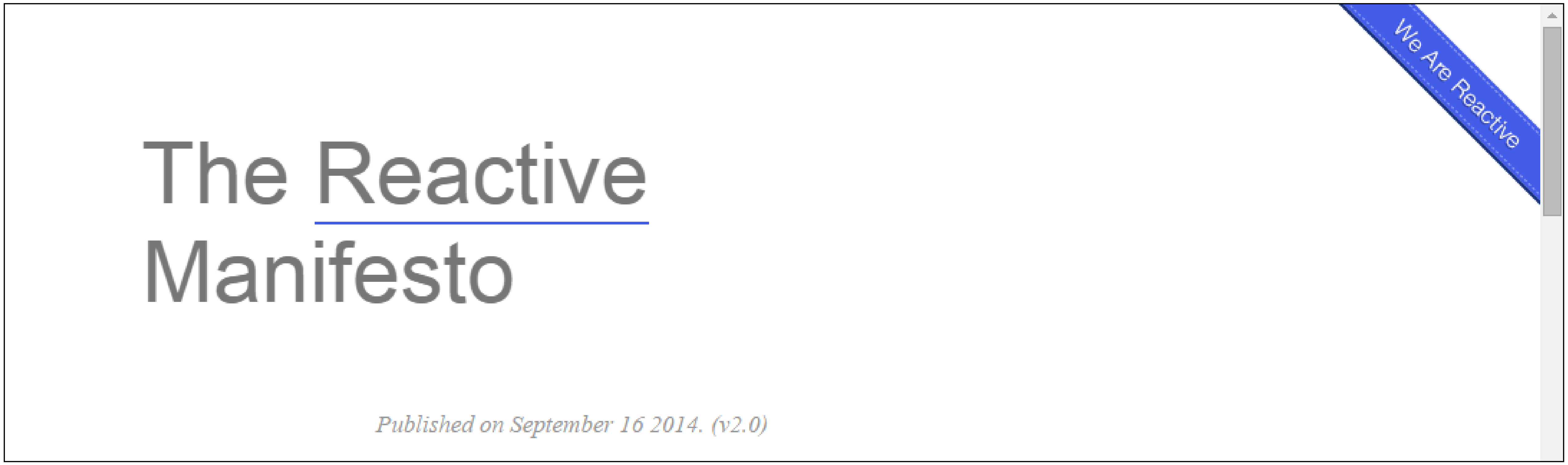
- Play is a based on a lightweight, stateless, web-friendly architecture.
- Web Framework for Java (and Scala).
- Play is a series of libraries available in Maven Repository.

…so you can use any Java build tool to build a Play project
e.g. maven, sbt, etc.

Play Framework and this module!

- Assignment 2 - you will refactor the pacemaker-console application as a cloud hosted service exposing a REST API.
  - Use the Play Framework to provide sufficient (but not too much) abstraction layers.
  - Use the Heroku cloud hosting service to deploy the application.
  - Attempt to keep as much of the model and service implementations from the console version intact.
  - Keep the app ‘Reactive’.

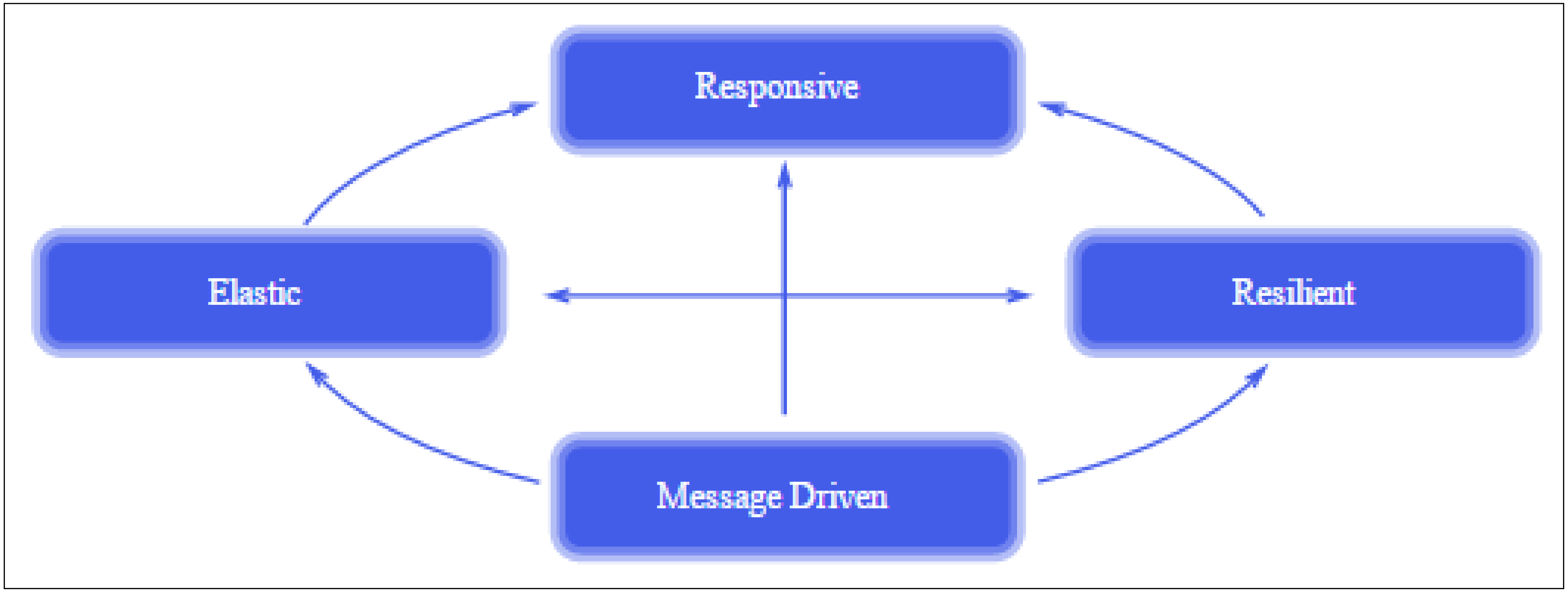




We Are Reactive

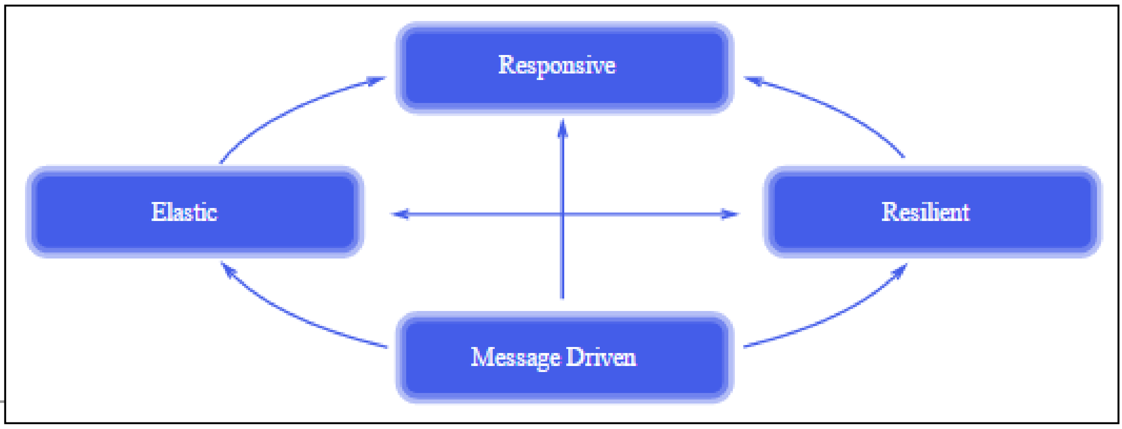
The Reactive Manifesto

Published on September 16 2014. (v2.0)



http://www.reactivemanifesto.org/

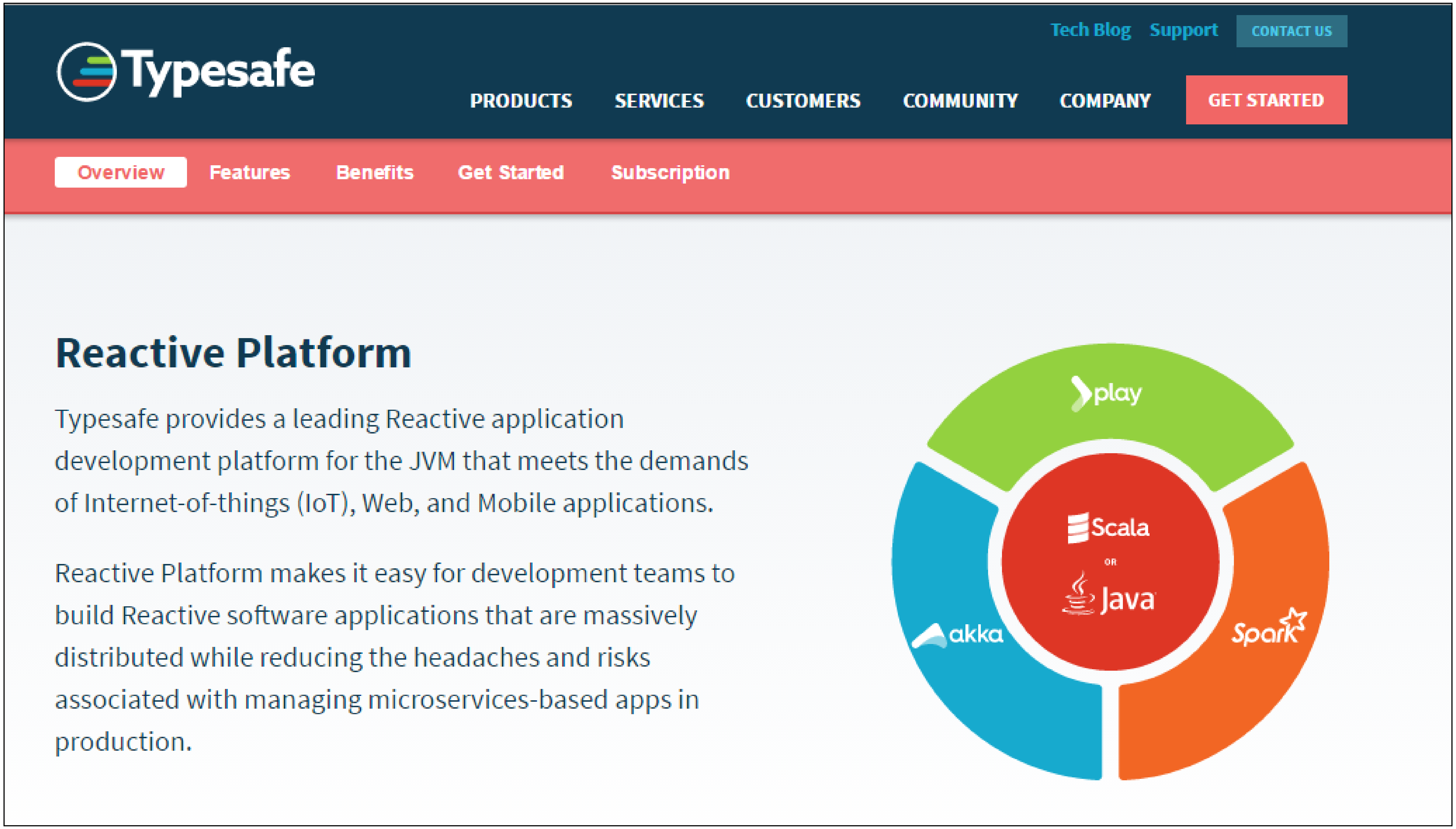
Reactive Manifesto



Responsive	• Responds in a timely manner. • Cornerstone of usability and utility; problems detected quickly and dealt with effectively. • Focus on rapid and consistent response times, delivering a consistent quality of service.
Resilient	• The system stays responsive in the face of failure; any system that is not resilient will be unresponsive after a failure. • Resilience is achieved by replication, containment, isolation and delegation.
Elastic	• The system stays responsive under varying workload. React to changes in the input rate by increasing or decreasing the resources allocated to service these inputs.
Message Driven	• Reactive Systems rely on asynchronous message-passing to establish a boundary between components that ensures loose coupling,

http://www.reactivemanifesto.org/

Typesafe – Reactive Manifesto and Play



Reactive Platform

Typesafe provides a leading Reactive application development platform for the JVM that meets the demands of Internet-of-things (IoT), Web, and Mobile applications.

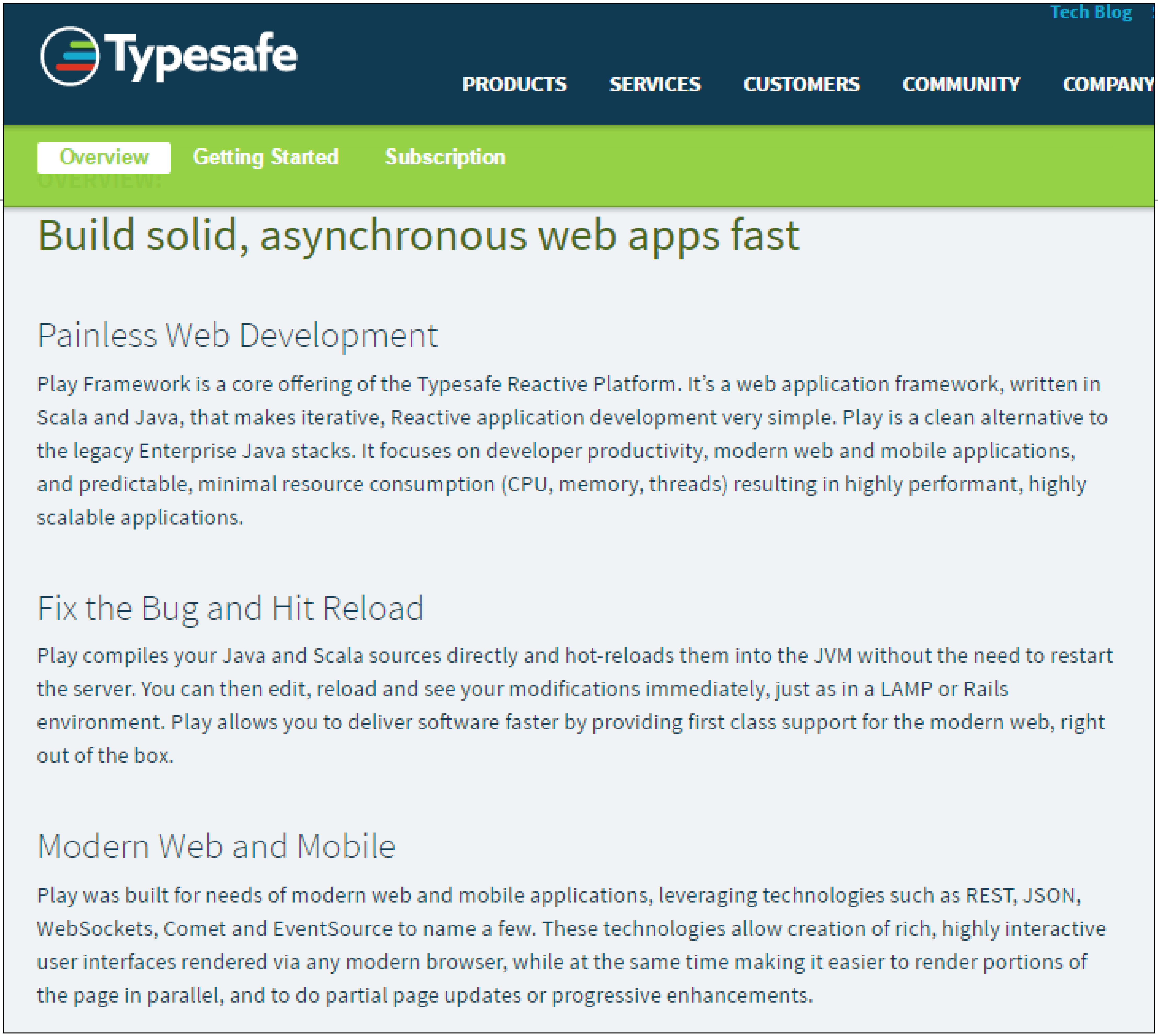
Reactive Platform makes it easy for development teams to build Reactive software applications that are massively distributed while reducing the headaches and risks associated with managing microservices-based apps in production.

https://www.typesafe.com/products/typesafe-reactive-platform#overview

Typesafe - Play

Play Framework makes it easy to build scalable, fast and real-time web applications with Java & Scala

https://www.typesafe.com/community/core-projects/play-framework



Build solid, asynchronous web apps fast

Painless Web Development

Play Framework is a core offering of the Typesafe Reactive Platform. It's a web application framework, written in Scala and Java, that makes iterative, Reactive application development very simple. Play is a clean alternative to the legacy Enterprise Java stacks. It focuses on developer productivity, modern web and mobile applications, and predictable, minimal resource consumption (CPU, memory, threads) resulting in highly performant, highly scalable applications.

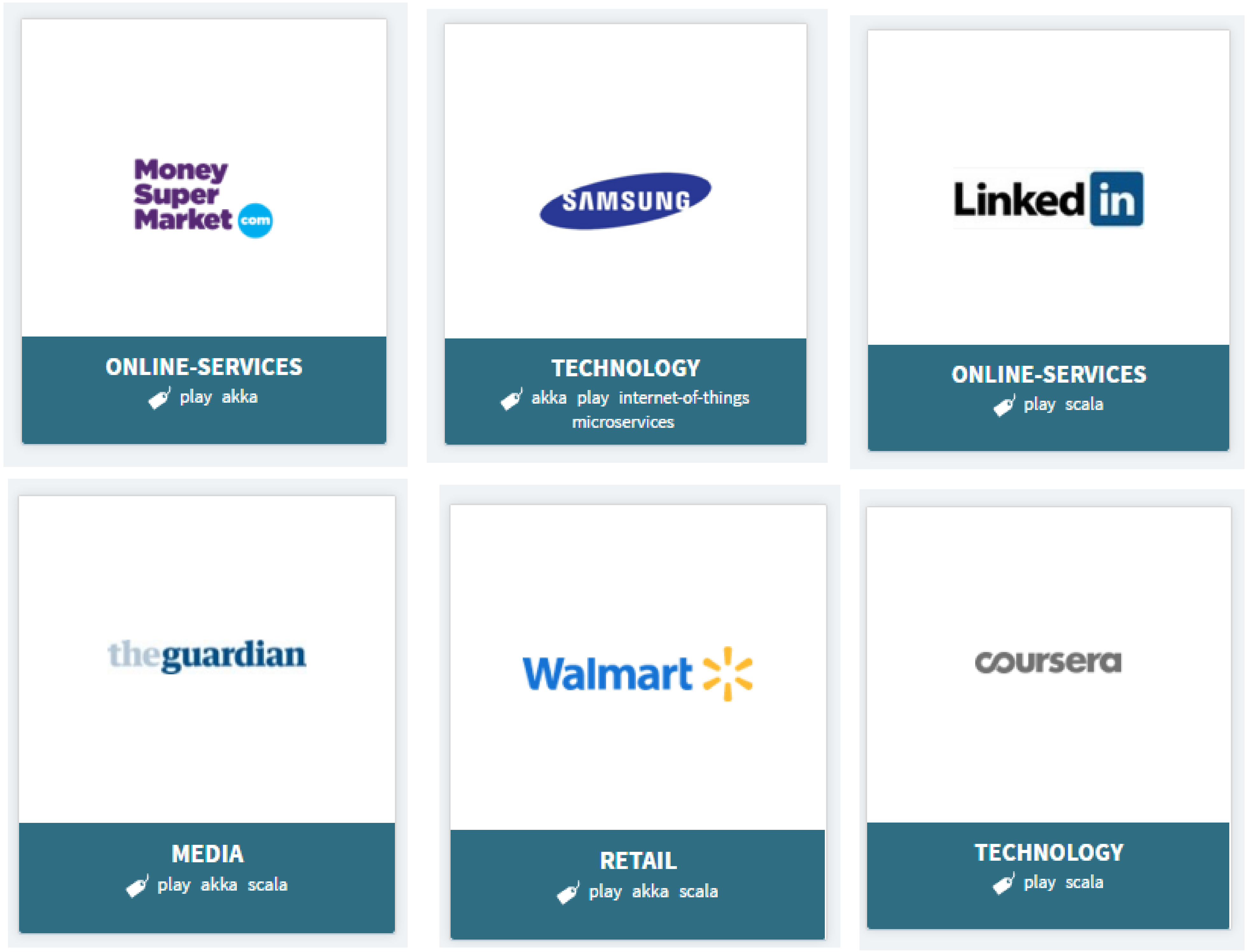
Fix the Bug and Hit Reload

Play compiles your Java and Scala sources directly and hot-reloads them into the JVM without the need to restart the server. You can then edit, reload and see your modifications immediately, just as in a LAMP or Rails environment. Play allows you to deliver software faster by providing first class support for the modern web, right out of the box.

Modern Web and Mobile

Play was built for needs of modern web and mobile applications, leveraging technologies such as REST, JSON, WebSockets, Comet and EventSource to name a few. These technologies allow creation of rich, highly interactive user interfaces rendered via any modern browser, while at the same time making it easier to render portions of the page in parallel, and to do partial page updates or progressive enhancements.

Some companies using Play



Money Super Market com

ONLINE-SERVICES

play akka

SAMSUNG

TECHNOLOGY

akka play internet-of-things microservices

Linked in

ONLINE-SERVICES

play scala

theguardian

MEDIA

play akka scala

Walmart

RETAIL

play akka scala

coursera

TECHNOLOGY

play scala

https://www.typesafe.com/resources/case-studies-and-stories

Installing Play (with Activator)

What is the Activator?

- Activator is the Lightbend Reactive Platform's build and tutorial tool. It also comes with a UI for learning Play.

- Activator can be described as “sbt plus templates” –

  - it combines sbt (a build tool) plus a means of downloading project templates (like Maven archetypes) and a web interface for managing those projects.

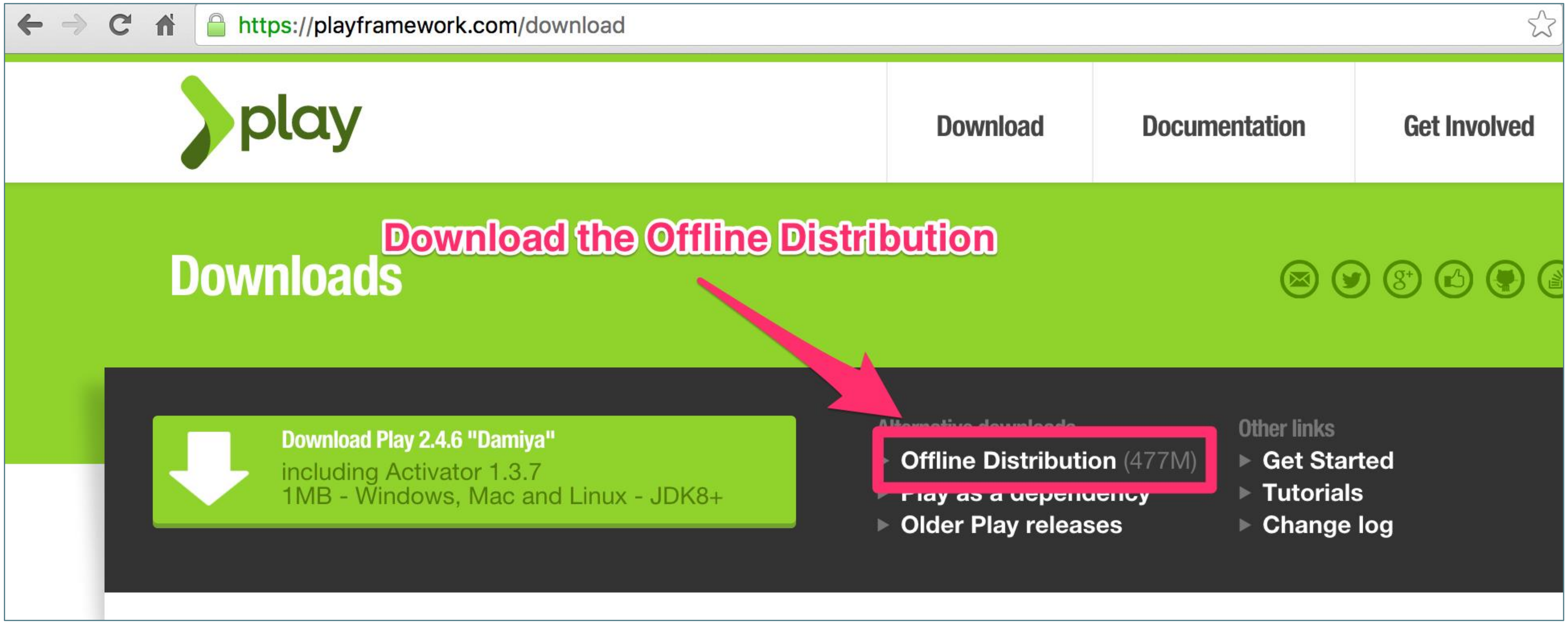
  - Templates can be examples (tutorials), or they can be “seed” templates that provide a starting point for your own projects.

More on SBT Build

- SBT is the build system underneath Play applications.
- It is responsible for resolving dependencies, compiling the project, running the tests, etc.
  - build.sbt → the sbt settings that describe building your app.
  - project/plugins.sbt → SBT plugins used by the project build including Play itself.
  - project/build.properties → marker file that declares the sbt version used.
- More detailed information on SBT in the Play Framework

Download Play (with Activator)

- https://www.playframework.com/download


https://playframework.com/download
play
Download
Documentation
Get Involved
Download the Offline Distribution
Downloads
Download Play 2.4.6 "Damiya"
including Activator 1.3.7
1MB - Windows, Mac and Linux - JDK8+
Alternative downloads
Offline Distribution (477M)
Play as a dependency
Older Play releases
Other links
Get Started
Tutorials
Change log

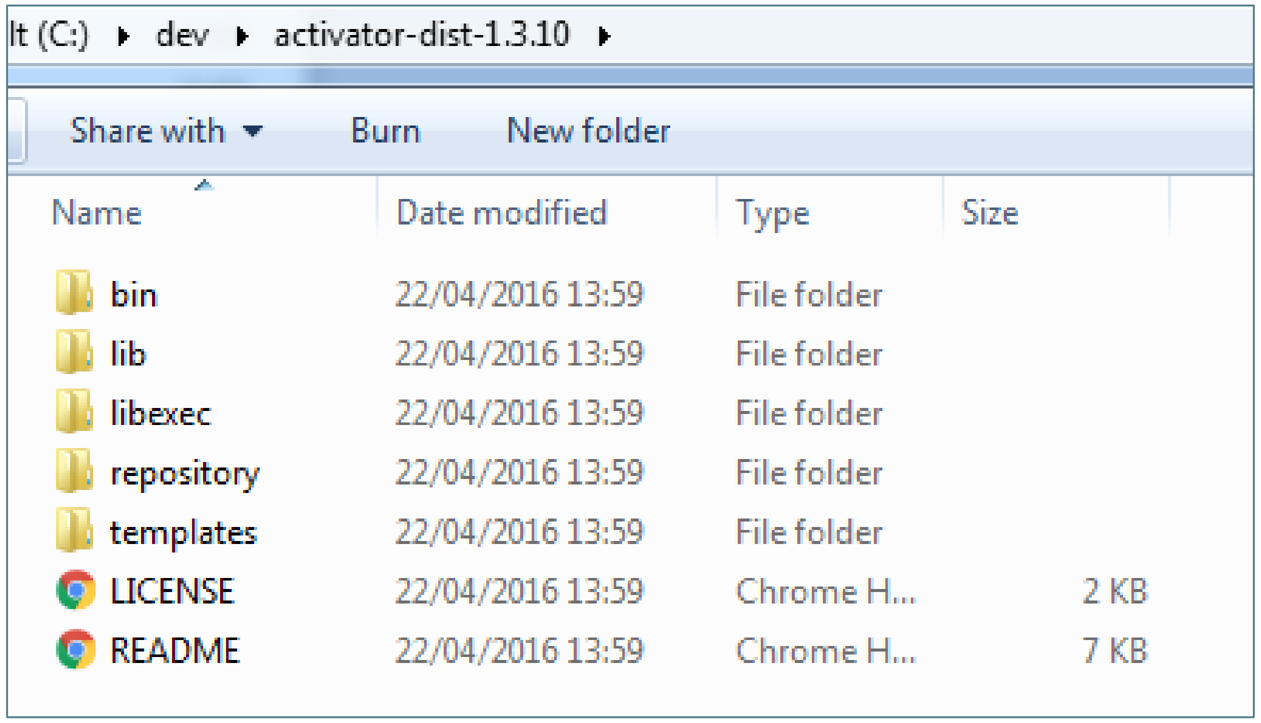
https://www.lightbend.com/activator/docs

Offline Distribution

- Activator is distributed as a single archive file that expands out to its own subdirectory.
- The “offline distribution” comes with all of Activator’s possible dependencies included.
  - much larger initial download,
  - but installing the offline distribution means starting up a new Play project is **much** faster, as all the dependencies are already resolved.

Installing Play (with Activator)

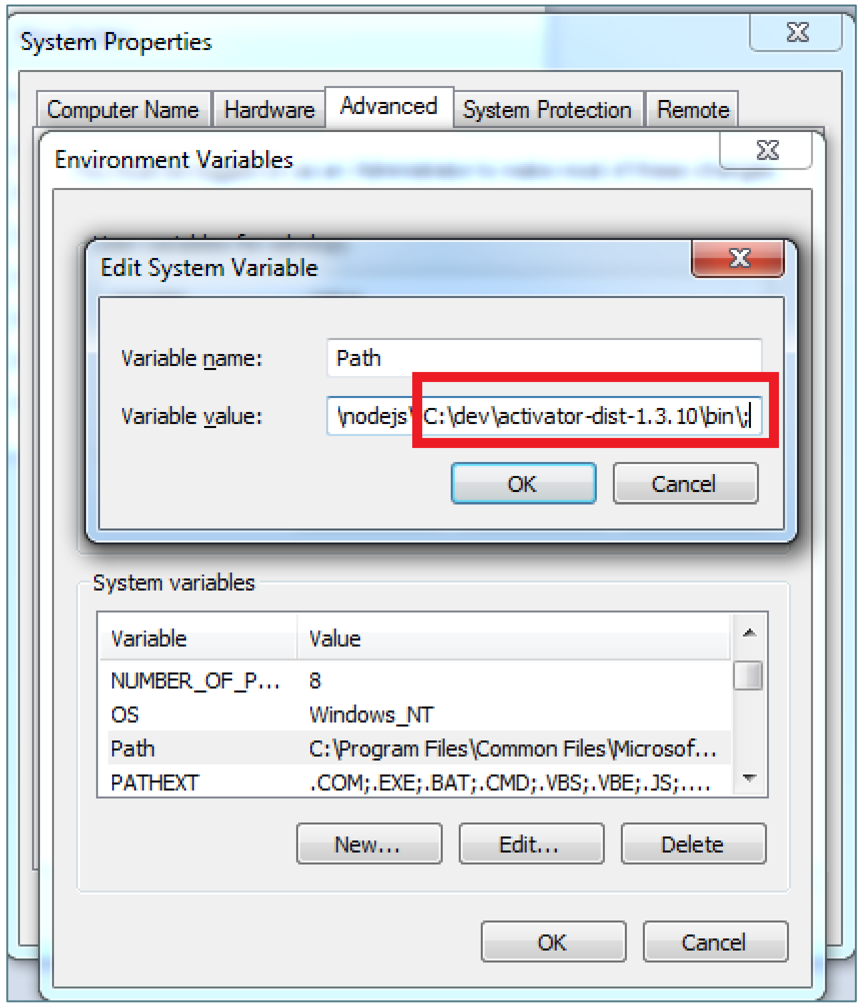
- Extract the **typesfe-activator** zip file to a folder where your development software resides e.g. a **dev** folder on your C drive.



lt (C:) ▸ dev ▸ activator-dist-1.3.10 ▸

Name	Date modified	Type	Size
bin	22/04/2016 13:59	File folder	
lib	22/04/2016 13:59	File folder	
libexec	22/04/2016 13:59	File folder	
repository	22/04/2016 13:59	File folder	
templates	22/04/2016 13:59	File folder	
LICENSE	22/04/2016 13:59	Chrome H...	2 KB
README	22/04/2016 13:59	Chrome H...	7 KB

- Add the Activator bin folder to your system path.



System Properties — Advanced — Environment Variables — Edit System Variable

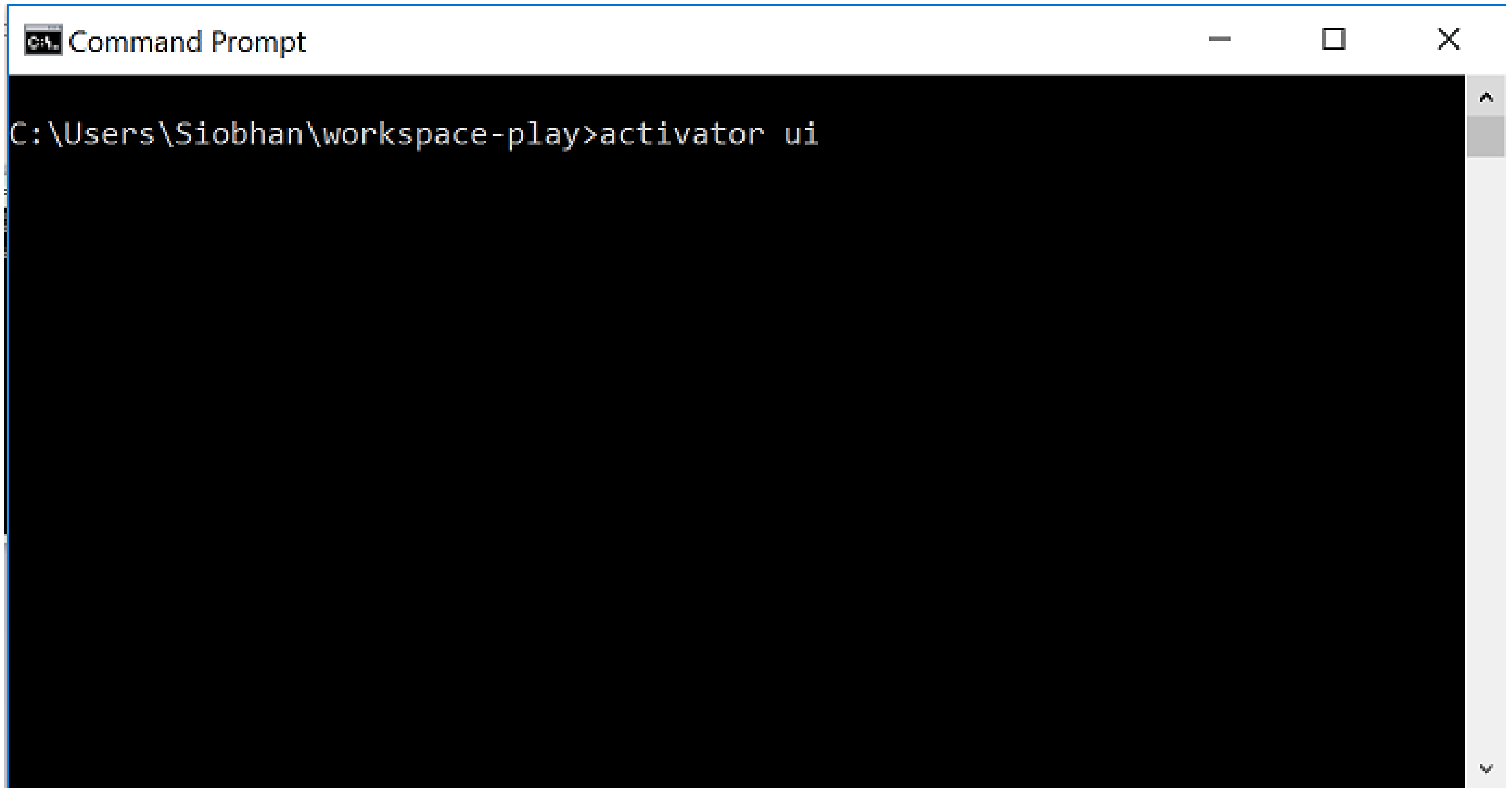
Variable name: Path

Variable value: \nodejs\ C:\dev\activator-dist-1.3.10\bin\;

System variables

Variable	Value
NUMBER_OF_P...	8
OS	Windows_NT
Path	C:\Program Files\Common Files\Microsof...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Starting Activator’s UI – Using Command Prompt

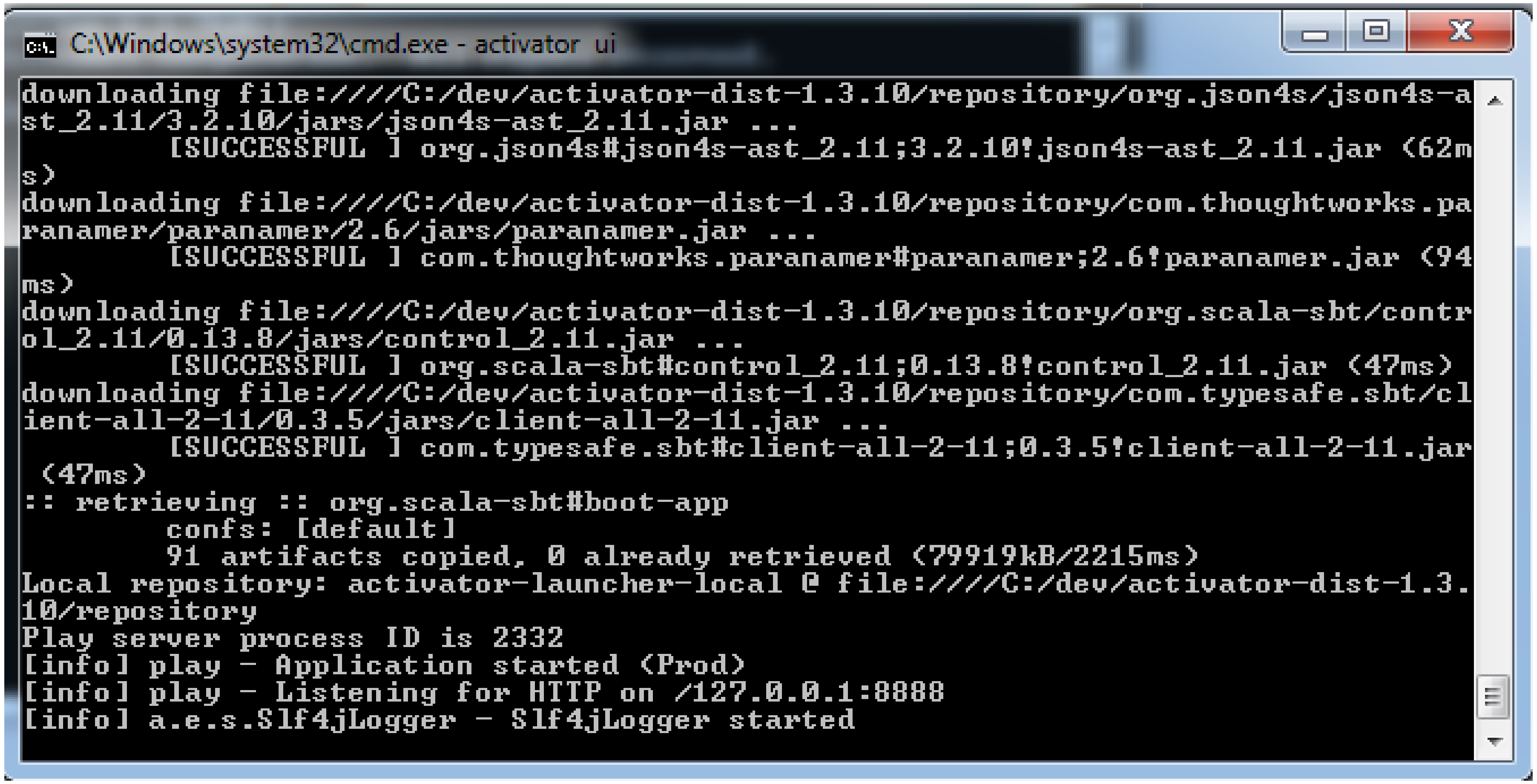


```
C:\Users\Siobhan\workspace-play>activator ui
```

Starting Activator's UI – Using Command Prompt

- The first time you run this command, a series of files will be downloaded.

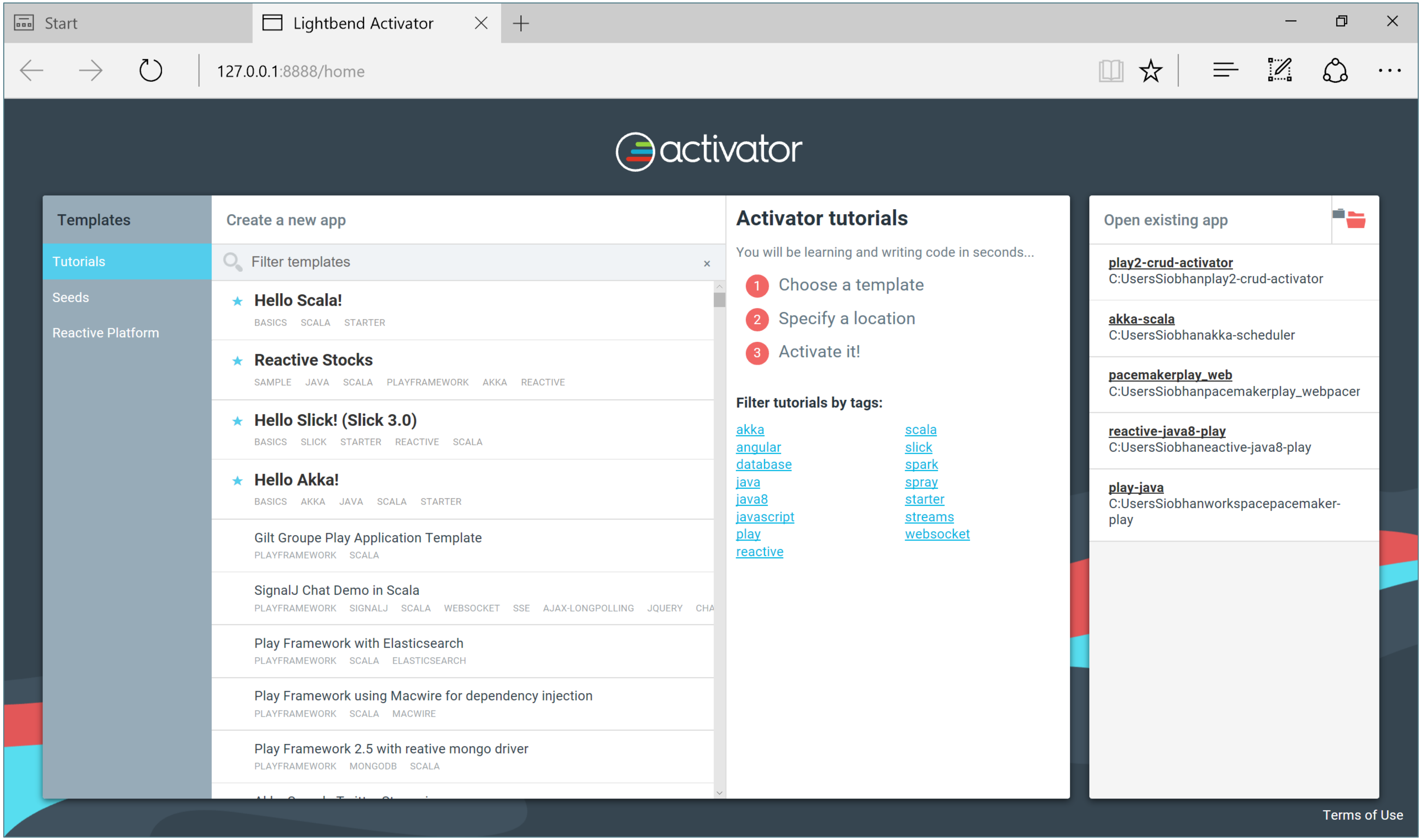
- The Activator UI will listen on http://localhost:8888 and the Web UI should automatically open for you.



C:\Windows\system32\cmd.exe - activator ui

```
downloading file:////C:/dev/activator-dist-1.3.10/repository/org.json4s/json4s-a
st_2.11/3.2.10/jars/json4s-ast_2.11.jar ...
        [SUCCESSFUL ] org.json4s#json4s-ast_2.11;3.2.10!json4s-ast_2.11.jar (62m
s)
downloading file:////C:/dev/activator-dist-1.3.10/repository/com.thoughtworks.pa
ranamer/paranamer/2.6/jars/paranamer.jar ...
        [SUCCESSFUL ] com.thoughtworks.paranamer#paranamer;2.6!paranamer.jar (94
ms)
downloading file:////C:/dev/activator-dist-1.3.10/repository/org.scala-sbt/contr
ol_2.11/0.13.8/jars/control_2.11.jar ...
        [SUCCESSFUL ] org.scala-sbt#control_2.11;0.13.8!control_2.11.jar (47ms)
downloading file:////C:/dev/activator-dist-1.3.10/repository/com.typesafe.sbt/cl
ient-all-2-11/0.3.5/jars/client-all-2-11.jar ...
        [SUCCESSFUL ] com.typesafe.sbt#client-all-2-11;0.3.5!client-all-2-11.jar
 (47ms)
:: retrieving :: org.scala-sbt#boot-app
        confs: [default]
        91 artifacts copied, 0 already retrieved (79919kB/2215ms)
Local repository: activator-launcher-local @ file:////C:/dev/activator-dist-1.3.
10/repository
Play server process ID is 2332
[info] play - Application started (Prod)
[info] play - Listening for HTTP on /127.0.0.1:8888
[info] a.e.s.Slf4jLogger - Slf4jLogger started
```

Activator UI



Start | Lightbend Activator

127.0.0.1:8888/home

activator

Templates

- Tutorials
- Seeds
- Reactive Platform

Create a new app

Filter templates

★ **Hello Scala!**
BASICS SCALA STARTER

★ **Reactive Stocks**
SAMPLE JAVA SCALA PLAYFRAMEWORK AKKA REACTIVE

★ **Hello Slick! (Slick 3.0)**
BASICS SLICK STARTER REACTIVE SCALA

★ **Hello Akka!**
BASICS AKKA JAVA SCALA STARTER

Gilt Groupe Play Application Template
PLAYFRAMEWORK SCALA

SignalJ Chat Demo in Scala
PLAYFRAMEWORK SIGNALJ SCALA WEBSOCKET SSE AJAX-LONGPOLLING JQUERY CHA

Play Framework with Elasticsearch
PLAYFRAMEWORK SCALA ELASTICSEARCH

Play Framework using Macwire for dependency injection
PLAYFRAMEWORK SCALA MACWIRE

Play Framework 2.5 with reative mongo driver
PLAYFRAMEWORK MONGODB SCALA

Activator tutorials

You will be learning and writing code in seconds...

1. Choose a template
2. Specify a location
3. Activate it!

Filter tutorials by tags:

akka	scala
angular	slick
database	spark
java	spray
java8	starter
javascript	streams
play	websocket
reactive	

Open existing app

play2-crud-activator
C:UsersSiobhanplay2-crud-activator

akka-scala
C:UsersSiobhanakka-scheduler

pacemakerplay_web
C:UsersSiobhanpacemakerplay_webpacer

reactive-java8-play
C:UsersSiobhaneactive-java8-play

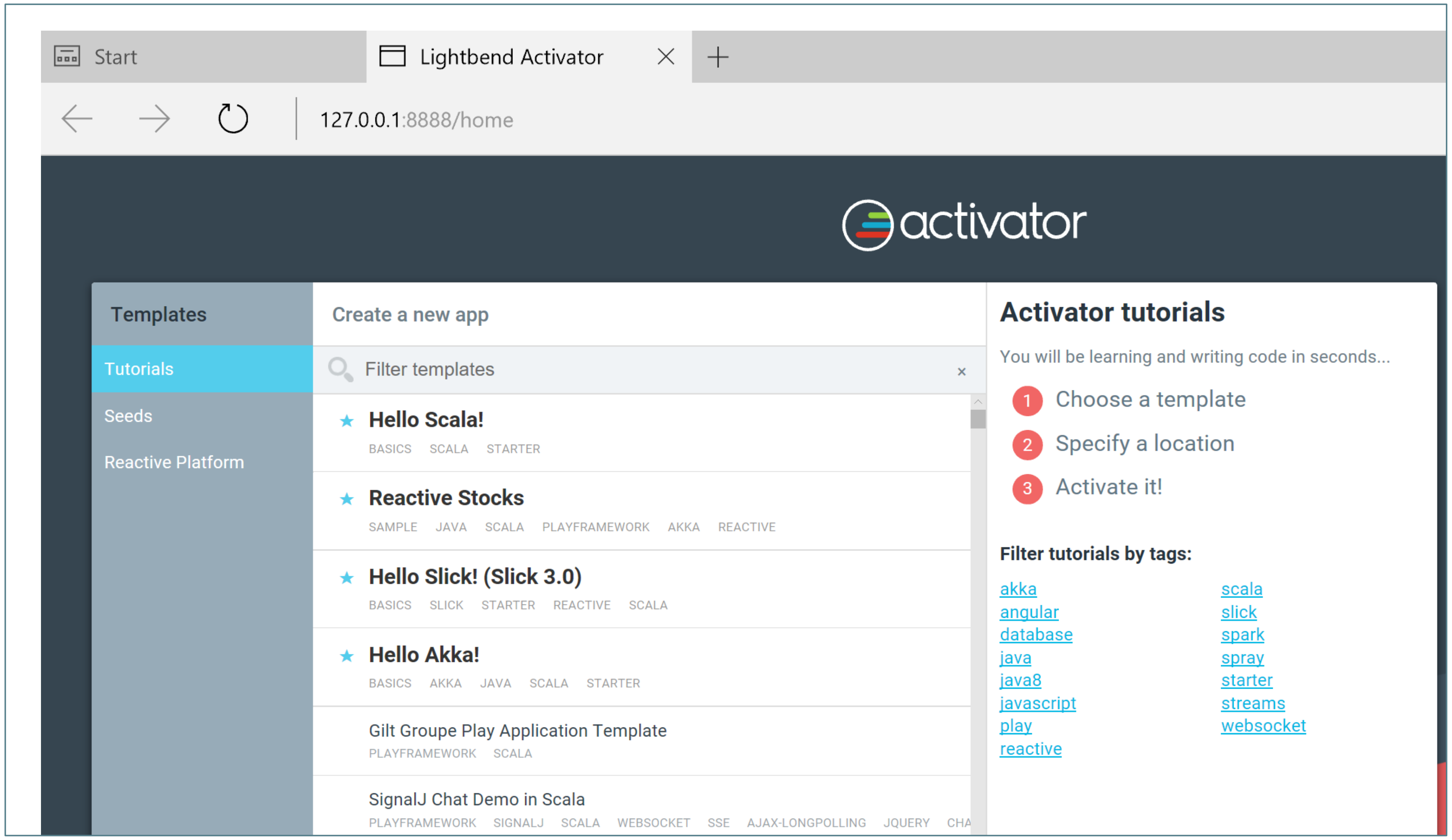
play-java
C:UsersSiobhanworkspacepacemaker-play

Terms of Use

Activator UI - Seeds and Tutorials

Activator comes with:

- *tutorials* → to walk you through various types of Play projects.
- *seeds* → that can be used to start off a Play project: e.g. play-java and play-scala.



Hello Scala!
BASICS SCALA STARTER

Reactive Stocks
SAMPLE JAVA SCALA PLAYFRAMEWORK AKKA REACTIVE

Hello Slick! (Slick 3.0)
BASICS SLICK STARTER REACTIVE SCALA

Hello Akka!
BASICS AKKA JAVA SCALA STARTER

Gilt Groupe Play Application Template
PLAYFRAMEWORK SCALA

SignalJ Chat Demo in Scala
PLAYFRAMEWORK SIGNALJ SCALA WEBSOCKET SSE AJAX-LONGPOLLING JQUERY CHA

Activator tutorials

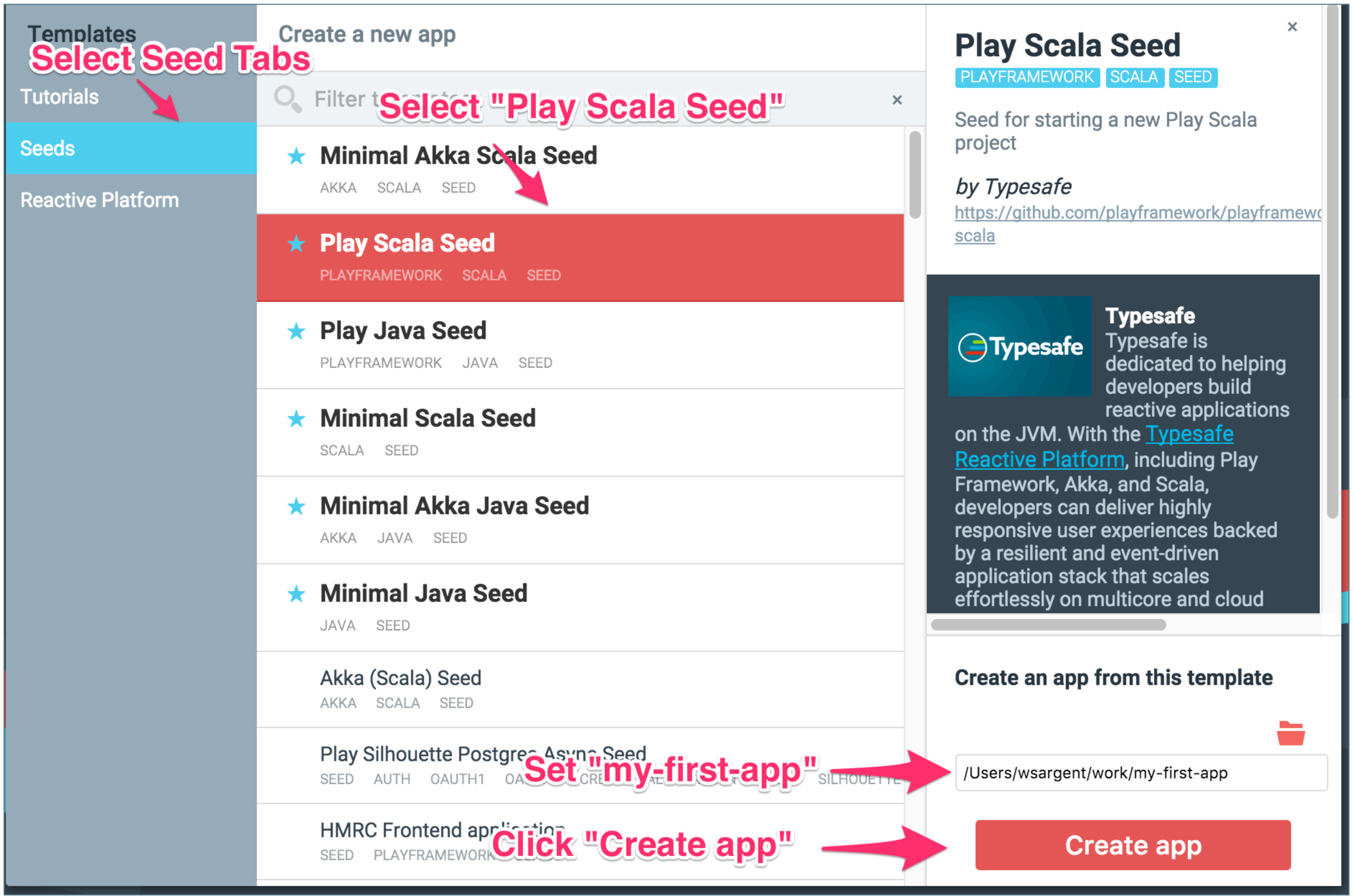
You will be learning and writing code in seconds...

1. Choose a template
2. Specify a location
3. Activate it!

Filter tutorials by tags:

akka, angular, database, java, java8, javascript, play, reactive, scala, slick, spark, spray, starter, streams, websocket

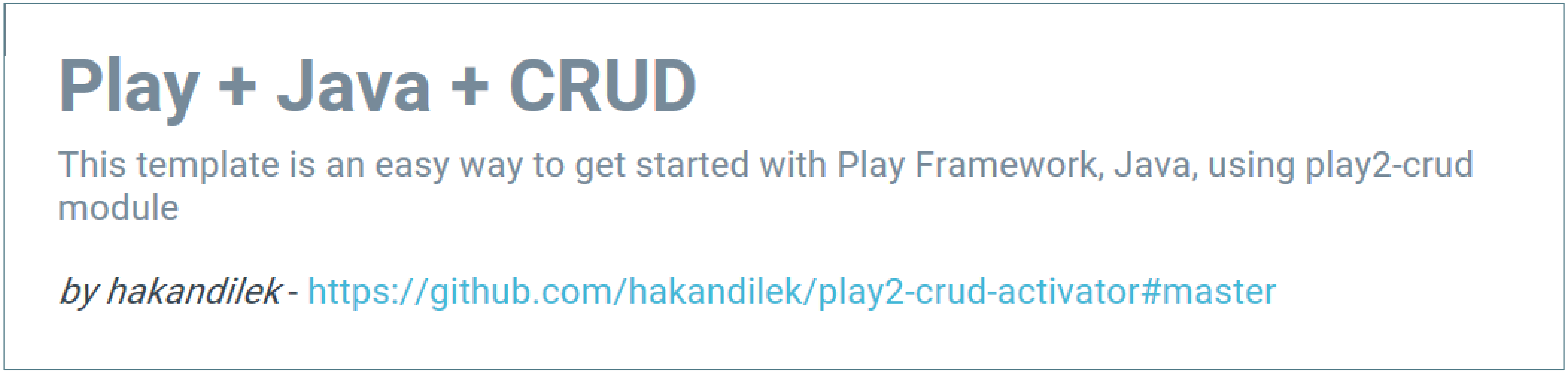
Activator UI - Seed Templates



Seed for starting a new Play Scala project

by Typesafe
https://github.com/playframework/playframew scala

Typesafe
Typesafe is dedicated to helping developers build reactive applications on the JVM. With the Typesafe Reactive Platform, including Play Framework, Akka, and Scala, developers can deliver highly responsive user experiences backed by a resilient and event-driven application stack that scales effortlessly on multicore and cloud

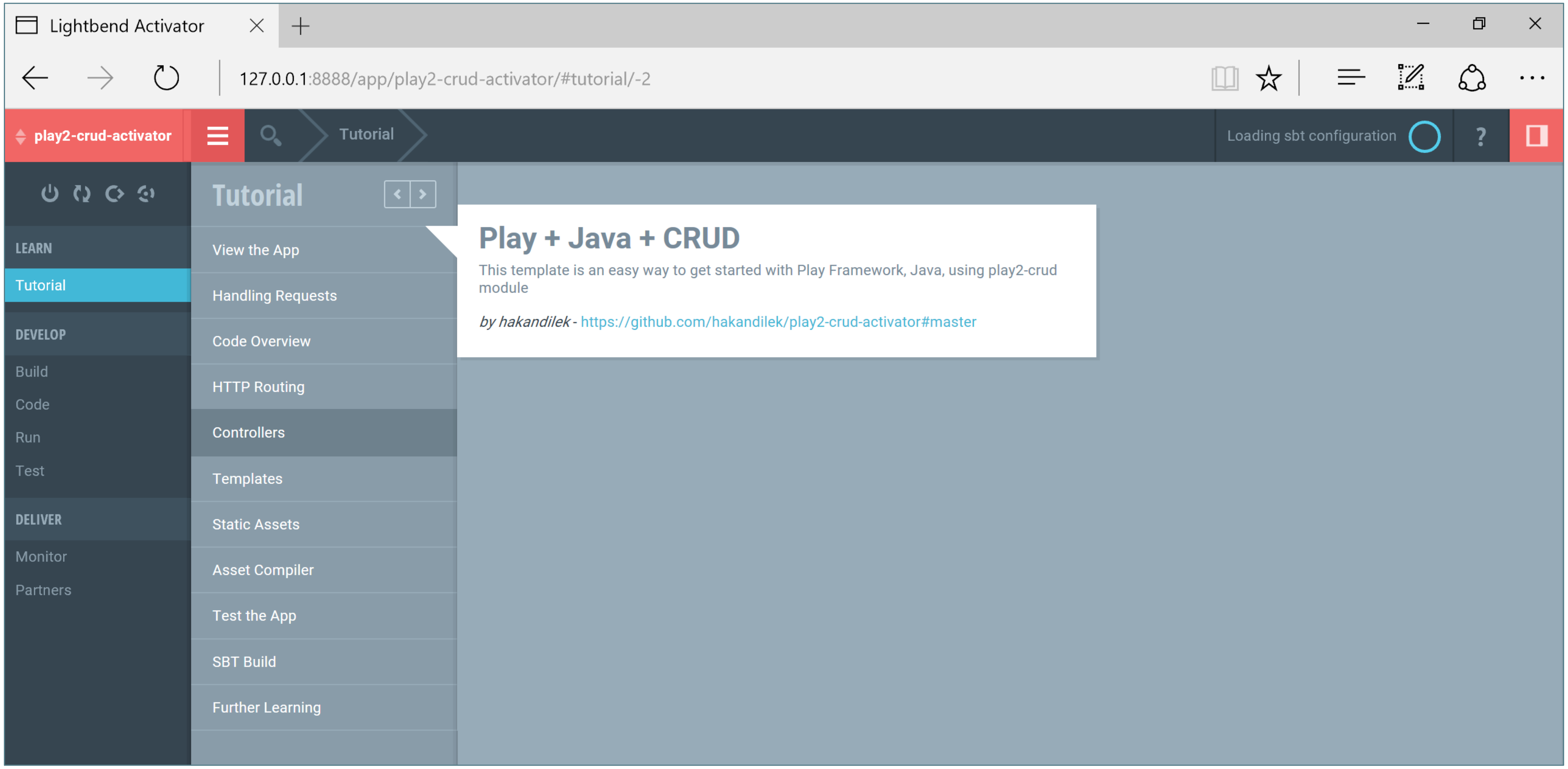

Play + Java + CRUD

This template is an easy way to get started with Play Framework, Java, using play2-crud module

by hakandilek - https://github.com/hakandilek/play2-crud-activator#master

You will step through this tutorial in labs this week.

Activator UI - Tutorials



Play + Java + CRUD

This template is an easy way to get started with Play Framework, Java, using play2-crud module

by hakandilek - https://github.com/hakandilek/play2-crud-activator#master

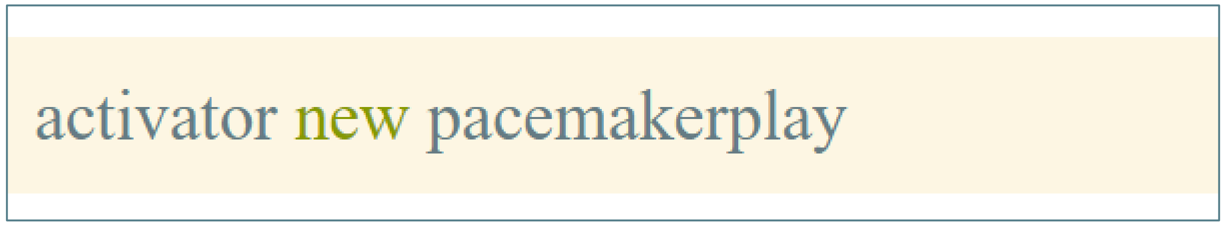
Hosting a new java-play seed project

- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

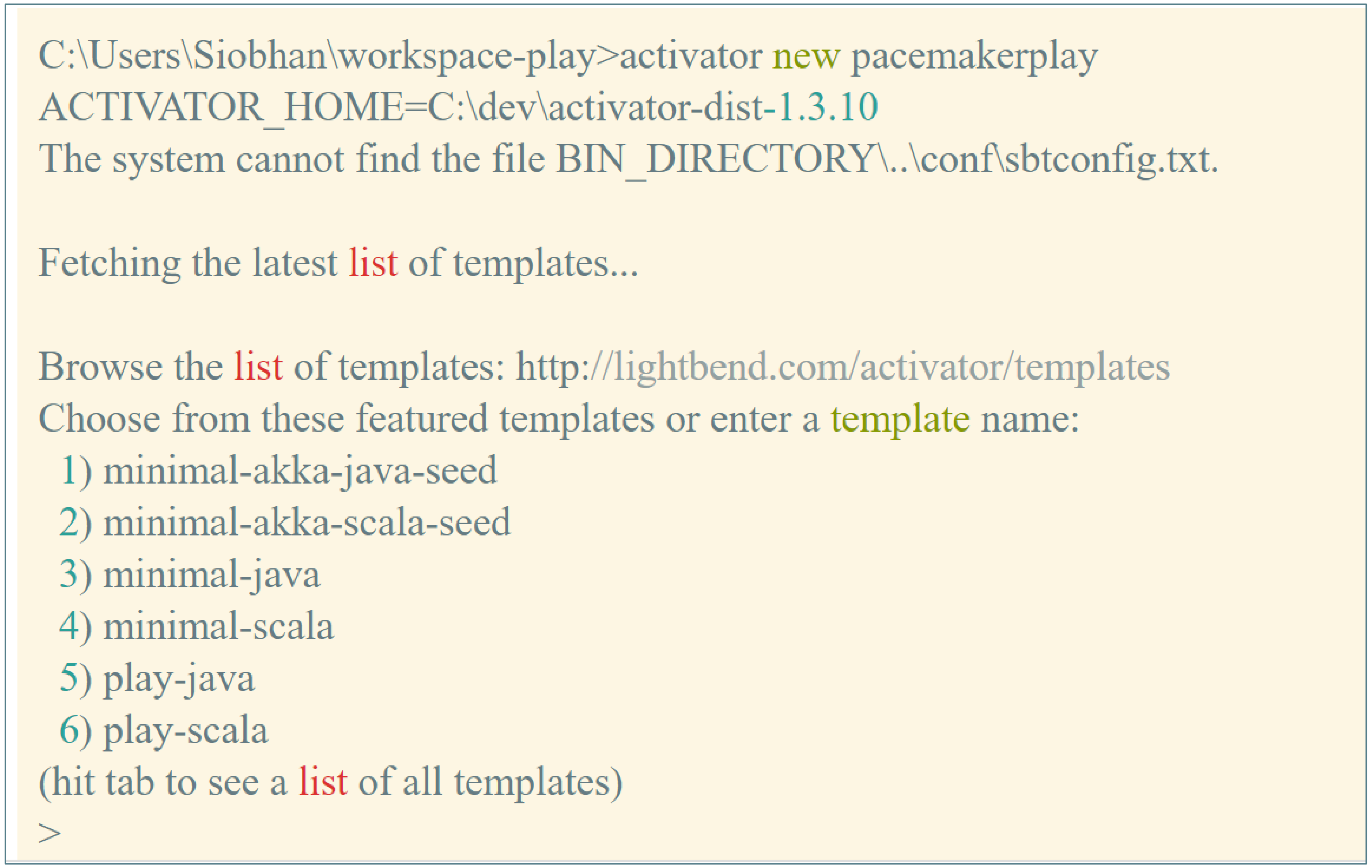
Create a new play-java project

With activator installed:

- Create a new project called pacemakerplay.
- When prompted, select seed option 5 to create a play-java project.



```
activator new pacemakerplay
```

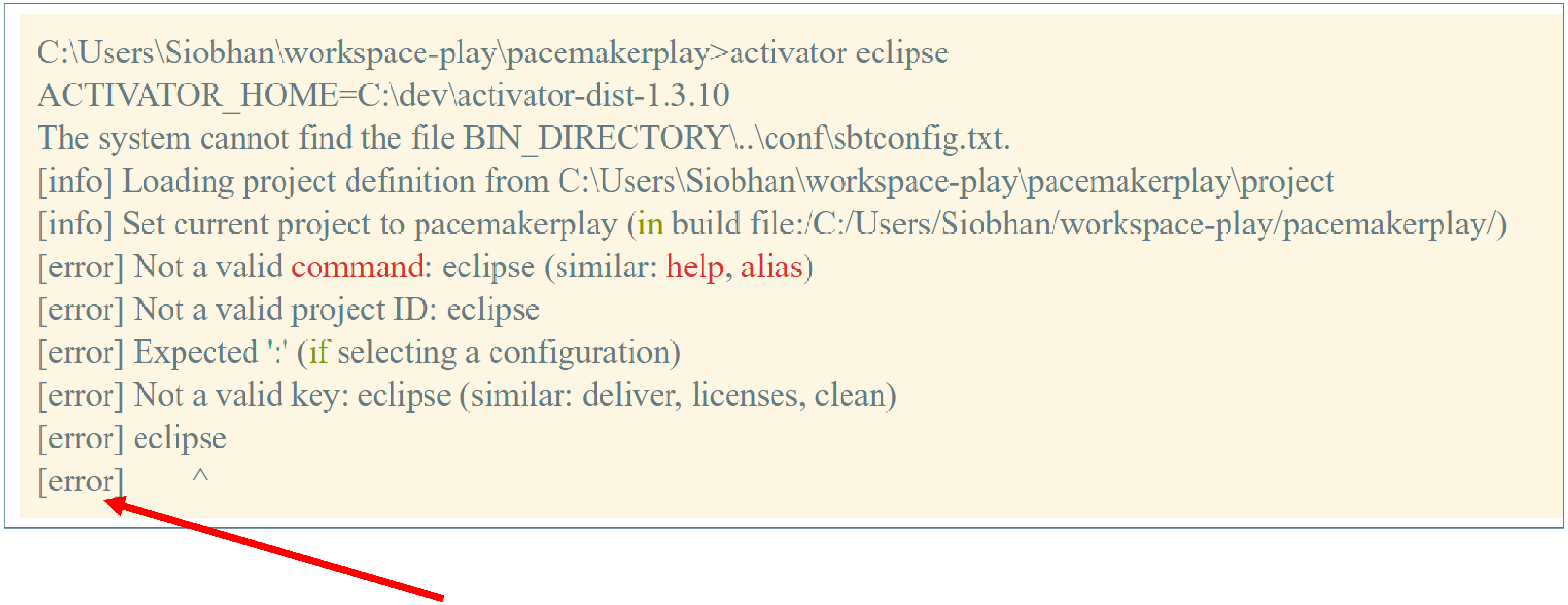


```
C:\Users\Siobhan\workspace-play>activator new pacemakerplay
ACTIVATOR_HOME=C:\dev\activator-dist-1.3.10
The system cannot find the file BIN_DIRECTORY\..\conf\sbtconfig.txt.

Fetching the latest list of templates...

Browse the list of templates: http://lightbend.com/activator/templates
Choose from these featured templates or enter a template name:
  1) minimal-akka-java-seed
  2) minimal-akka-scala-seed
  3) minimal-java
  4) minimal-scala
  5) play-java
  6) play-scala
(hit tab to see a list of all templates)
>
```

Create a new play-java project

Change directory into your new project (pacemakerplay) and attempt to “eclipsify” your project.

```
activator eclipse
```

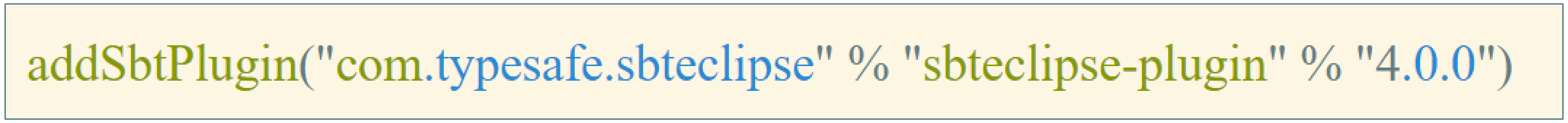


```
C:\Users\Siobhan\workspace-play\pacemakerplay>activator eclipse
ACTIVATOR_HOME=C:\dev\activator-dist-1.3.10
The system cannot find the file BIN_DIRECTORY\..\conf\sbtconfig.txt.
[info] Loading project definition from C:\Users\Siobhan\workspace-play\pacemakerplay\project
[info] Set current project to pacemakerplay (in build file:/C:/Users/Siobhan/workspace-play/pacemakerplay/)
[error] Not a valid command: eclipse (similar: help, alias)
[error] Not a valid project ID: eclipse
[error] Expected ':' (if selecting a configuration)
[error] Not a valid key: eclipse (similar: deliver, licenses, clean)
[error] eclipse
[error]        ^
```

Create a new play-java project

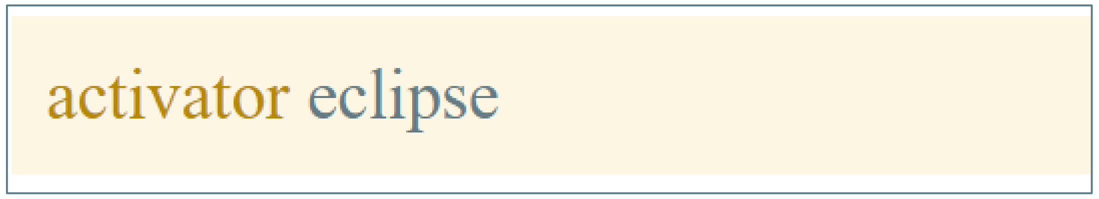
This is a problem with sbt (the build system) → it doesn't have the sbteclipse plugin installed.

Edit the generated file **plugins.sbt** (found in pacemakerplay\project directory) to include the following plugin:



```
addSbtPlugin("com.typesafe.sbteclipse" % "sbteclipse-plugin" % "4.0.0")
```

Run the command again:

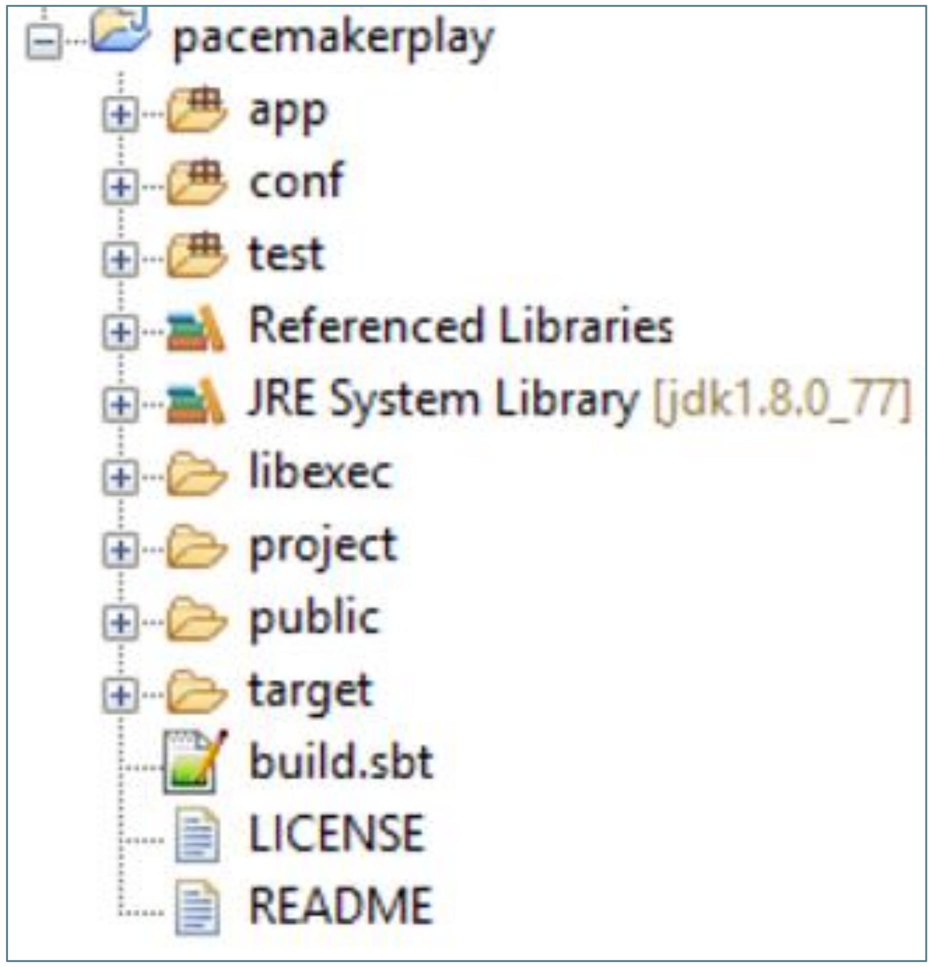


```
activator eclipse
```

Create a new play-java project

The command should work now.

Open Eclipse and import your **pacemakerplay** project.



pacemakerplay
app
conf
test
Referenced Libraries
JRE System Library [jdk1.8.0_77]
libexec
project
public
target
build.sbt
LICENSE
README

A Play Application has very few required files.

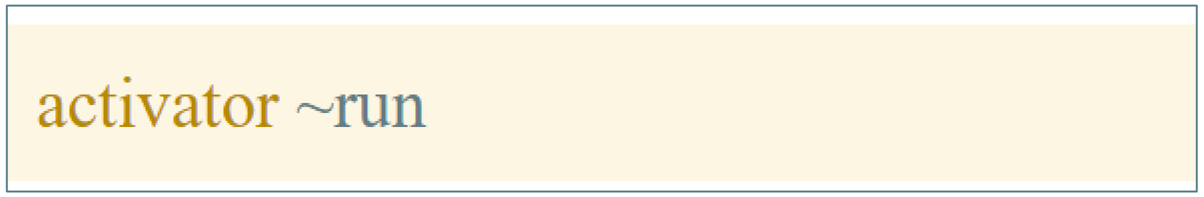
- **app** directory: contains source code.
- **conf** directory: contains *application.conf* and *routes* files.
- **project** directory: contains SBT information.
- **test** directory: contains unit, functional and integration tests.
- **public** directory: contains static assets e.g. CSS, images, etc.

Hosting a new java-play seed project

- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

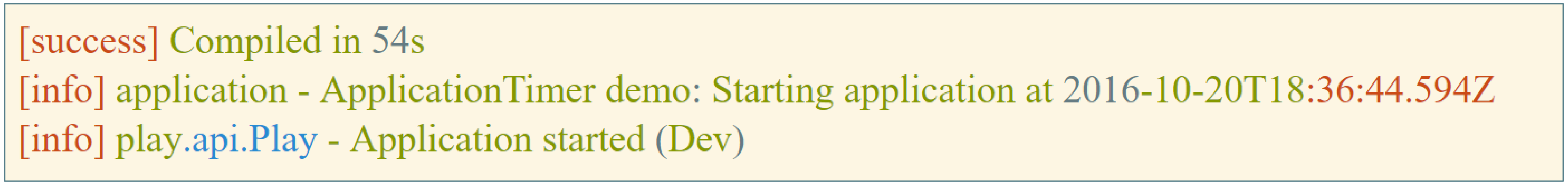
Running play on localhost:9000

- Within the **pacemakerplay** folder, enter the command to run play in continuous mode (i.e. triggered compilations will be enabled while the development server is running):



```
activator ~run
```

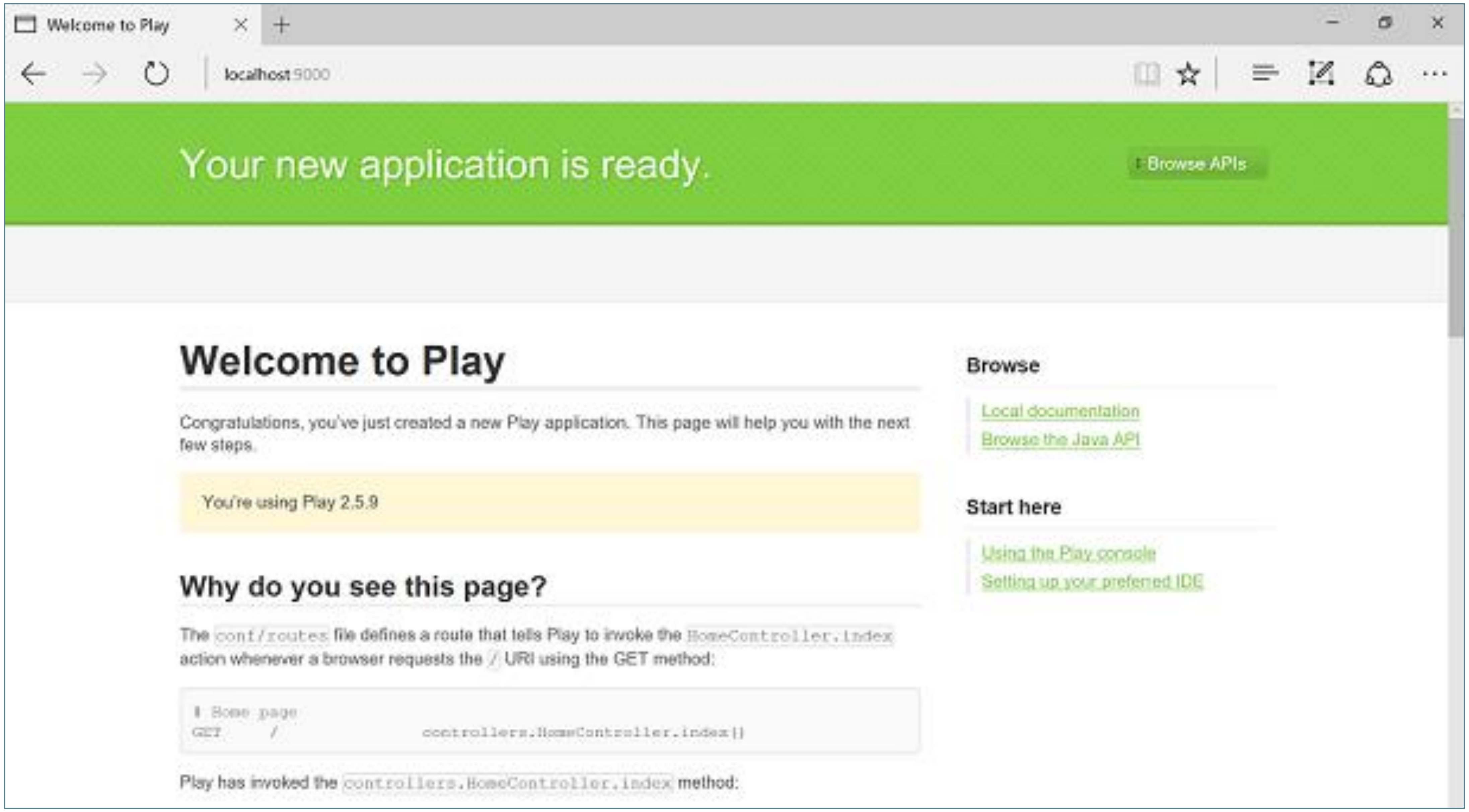
- When the command has finished executing, it should inform you of the following:



```
[success] Compiled in 54s
[info] application - ApplicationTimer demo: Starting application at 2016-10-20T18:36:44.594Z
[info] play.api.Play - Application started (Dev)
```

Running play on localhost:9000

- Navigate to http://localhost:9000 to view the default greeting page for your app:



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 2.5.9

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:

Browse

Local documentation
Browse the Java API

Start here

Using the Play console
Setting up your preferred IDE

Hosting a new java-play seed project

- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

REST

- REST stands for **Re**presentational **S**tate **T**ransfer.
- REST is an architecture style for designing networked applications; simple HTTP is used to make calls between machines.
- RESTful applications use HTTP requests to post data (create and/or update), read data (e.g., make queries), and delete data. Thus, REST uses HTTP for all four CRUD (Create/Read/Update/Delete) operations.
- Play is designed to support REST.

http://rest.elkstein.org/2008/02/what-is-rest.html

REST and Play!

- The Play framework makes it easy to build RESTful applications:
  - The Play router interprets both:
    URI (Uniform Resource Identifier) and
    HTTP (HyperText Transfer Protocol) methods
    to route a request to a Java call.
  - The protocol is stateless. This means you can’t save any state on the server between two successive requests.
  - Play considers HTTP as a key feature, thus the framework gives you full access to HTTP information.

Handling Requests

- When you make an HTTP request to a URL, the Play server figures out what code to execute to handle the request and return a response.
- In this application the request handler for requests to the root URL (e.g. "/") are handled by a Java Controller.
- You can use Java (and Scala) to create your controllers.
- Controllers asynchronously return HTTP responses of any content type (i.e. HTML, JSON, binary).

HTTP Routing, Controllers and Templates

- conf/routes → the configuration file used by the Play Router.
- Lists all the HTTP routes needed by the application.
- Each route consists of a mapping between a HTTP request verb and the controller method (Java code in our case) that handles the request.
- Any browser can access the application services through the defined routes.

Route matches HTTP method + URI → Java call.

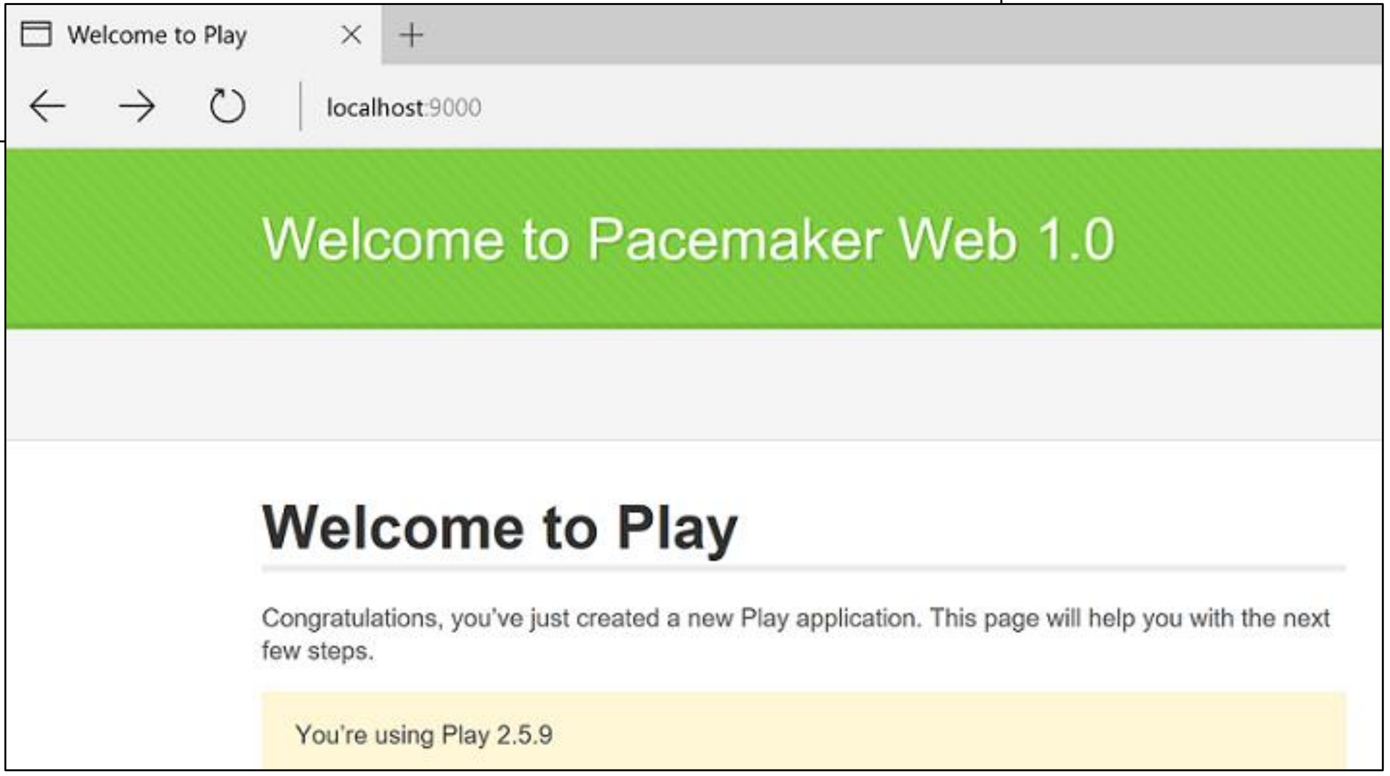
```
GET          /                 controllers.HomeController.index
```

HTTP Method

URI

Java Call

```
public class HomeController extends Controller {

public Result index() {
        return ok(index.render("Welcome to Pacemaker Web 1.0"));
    }

}
```


Welcome to Play

localhost:9000

Welcome to Pacemaker Web 1.0

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 2.5.9

Hosting a new java-play seed project

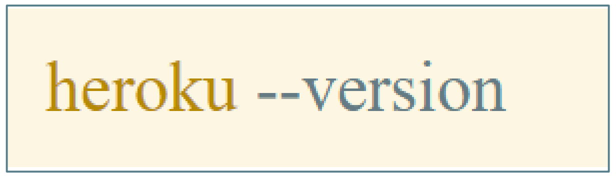
- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

Install Heroku (cloud application platform)

- Create a free Heroku account: https://signup.heroku.com/

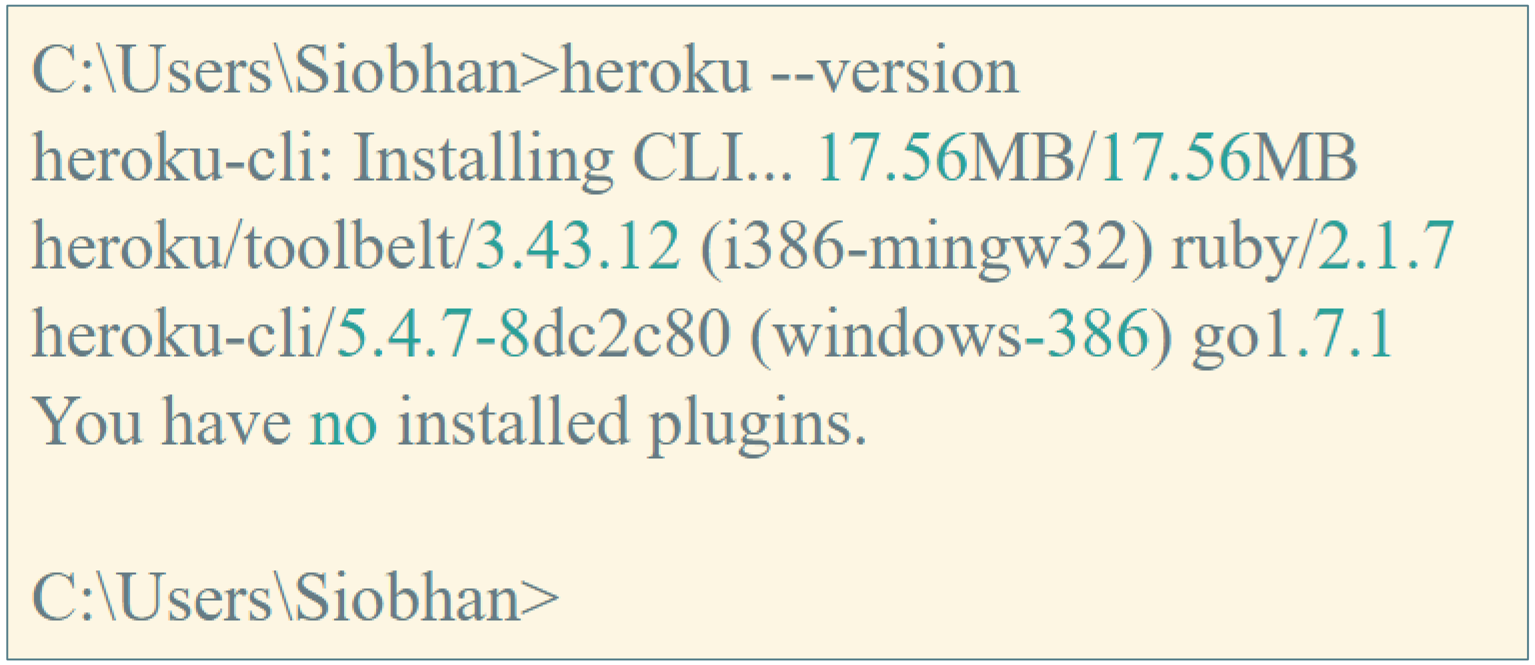
- Install Heroku CLI, the command shell which contains git (previously called Heroku Toolbelt): https://devcenter.heroku.com/articles/heroku-command-line

- Verify that Heroku installed successfully:



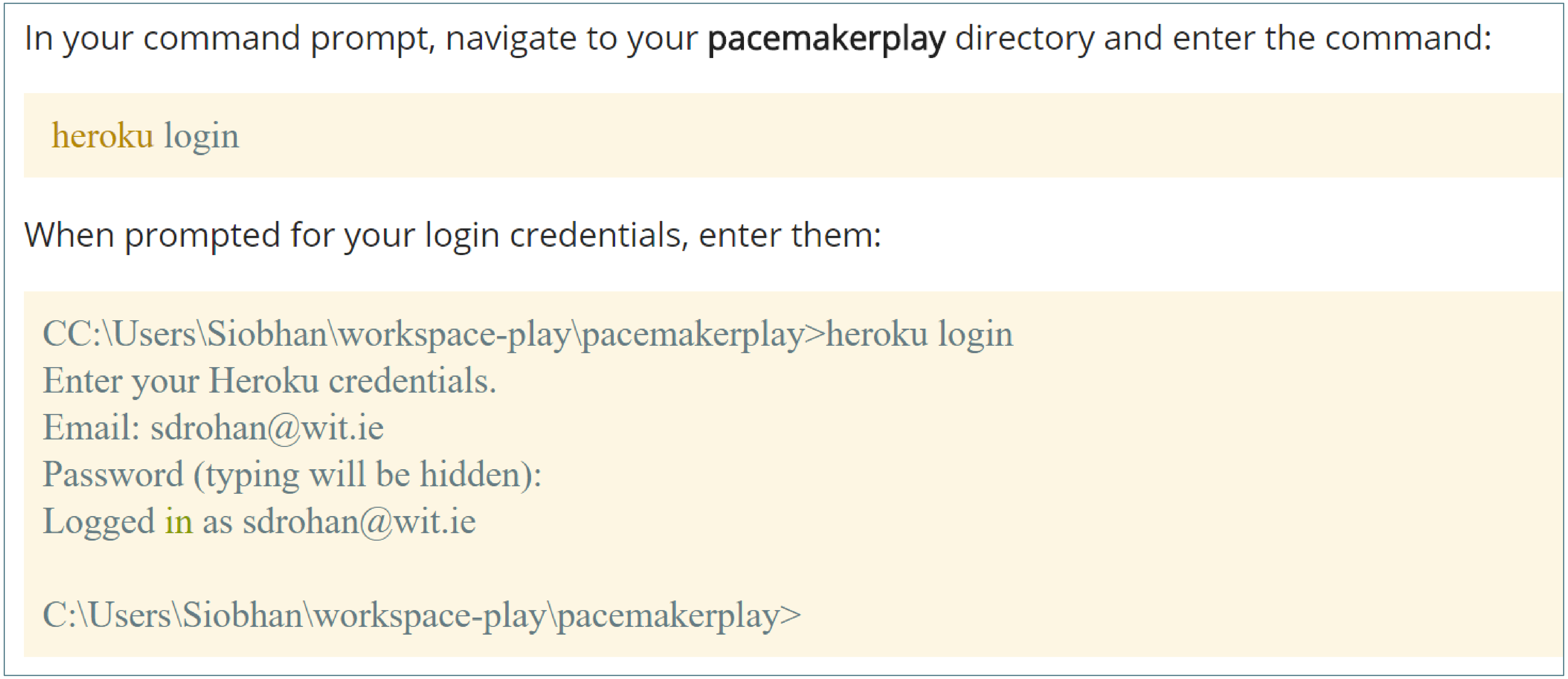
```
heroku --version
```

- This command will perform an initial install:



```
C:\Users\Siobhan>heroku --version
heroku-cli: Installing CLI... 17.56MB/17.56MB
heroku/toolbelt/3.43.12 (i386-mingw32) ruby/2.1.7
heroku-cli/5.4.7-8dc2c80 (windows-386) go1.7.1
You have no installed plugins.

C:\Users\Siobhan>
```

Log into Heroku



In your command prompt, navigate to your **pacemakerplay** directory and enter the command:

```
heroku login
```

When prompted for your login credentials, enter them:

```
CC:\Users\Siobhan\workspace-play\pacemakerplay>heroku login
Enter your Heroku credentials.
Email: sdrohan@wit.ie
Password (typing will be hidden):
Logged in as sdrohan@wit.ie

C:\Users\Siobhan\workspace-play\pacemakerplay>
```

Setting up a shared SSH Key

- To use SSH Git transport on Heroku, you'll need to create a public/private key pair to deploy code.

- This keypair is used for the strong cryptography and that uniquely identifies you as a developer when pushing code changes.

- Instructions to set up your key in your **git-bash shell**: https://devcenter.heroku.com/articles/keys

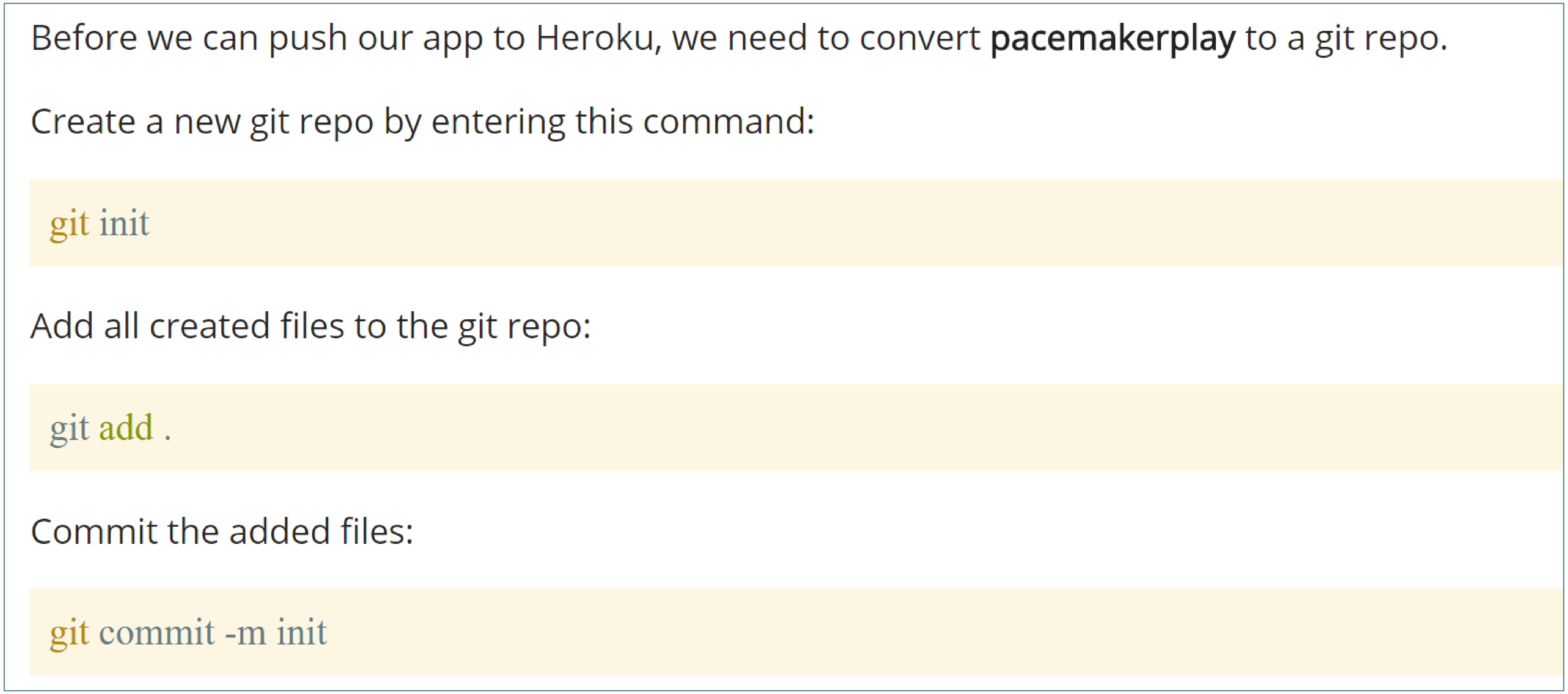
- Once you have your key generated, return to your **command prompt** and enter:

```
heroku keys:add
```

Hosting a new java-play seed project

- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

Git-enabling your app



Before we can push our app to Heroku, we need to convert **pacemakerplay** to a git repo.

Create a new git repo by entering this command:

```
git init
```

Add all created files to the git repo:

```
git add .
```

Commit the added files:

```
git commit -m init
```

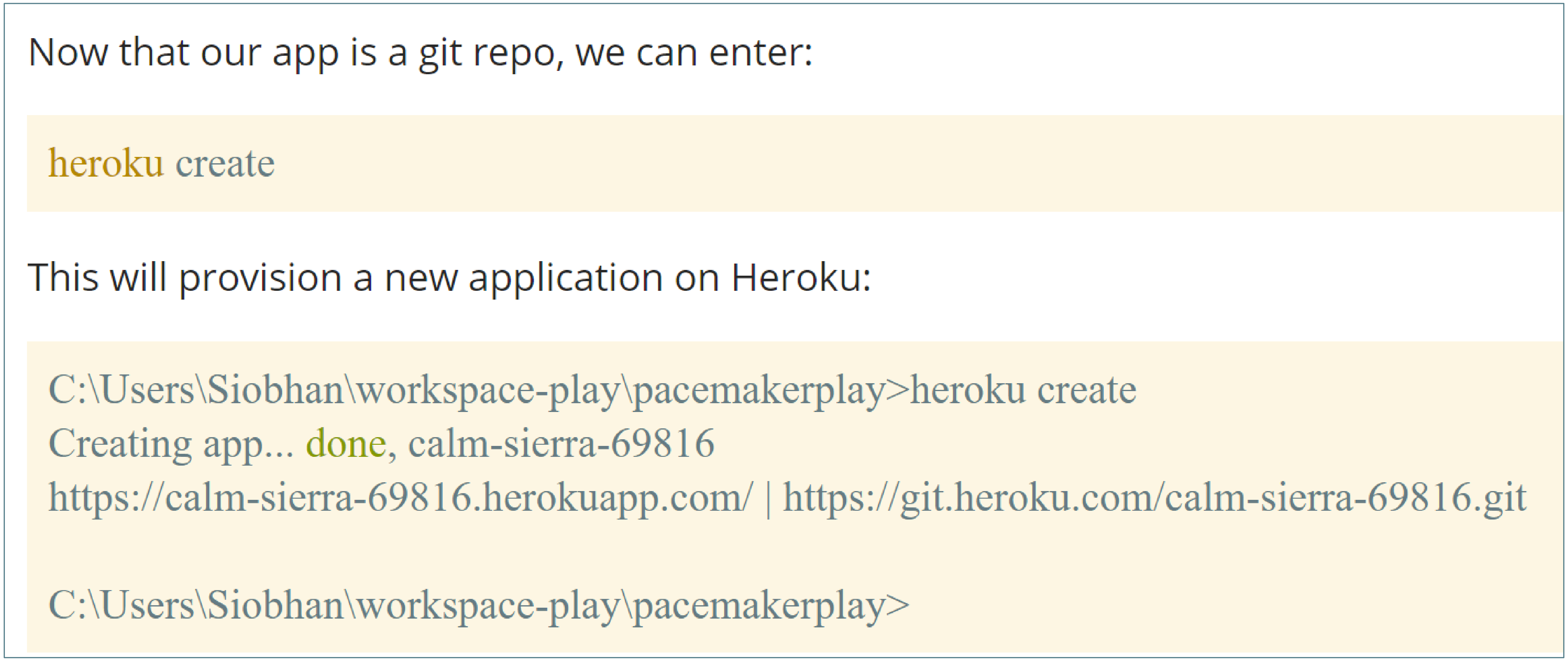
Some Git Shell Commands (heroku cli)

git init	Makes your current directory a Git repository.
git add .	Adds all modified and new files found in the current directory (and subdirectories) to the staging area (i.e. the index). They are then ready for inclusion in the next commit.
git commit –m “init”	To store all the files in your staging area into your Git repository, you need to commit them. The message we attached to this commit is “init”. You can use any message.

Hosting a new java-play seed project

- Create a new java-play project
- Running play on localhost
- REST and Routes in Play
- Setting up Heroku
- Git-enabling your app
- Deploying to Heroku

Provisioning a new app on Heroku



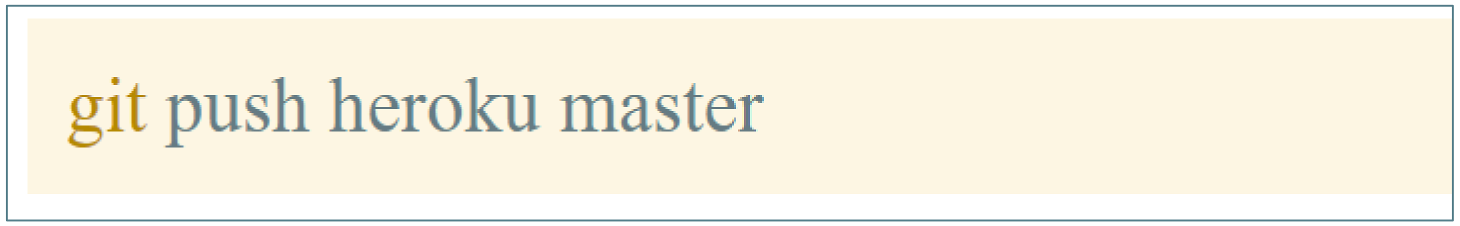
Now that our app is a git repo, we can enter:

```
heroku create
```

This will provision a new application on Heroku:

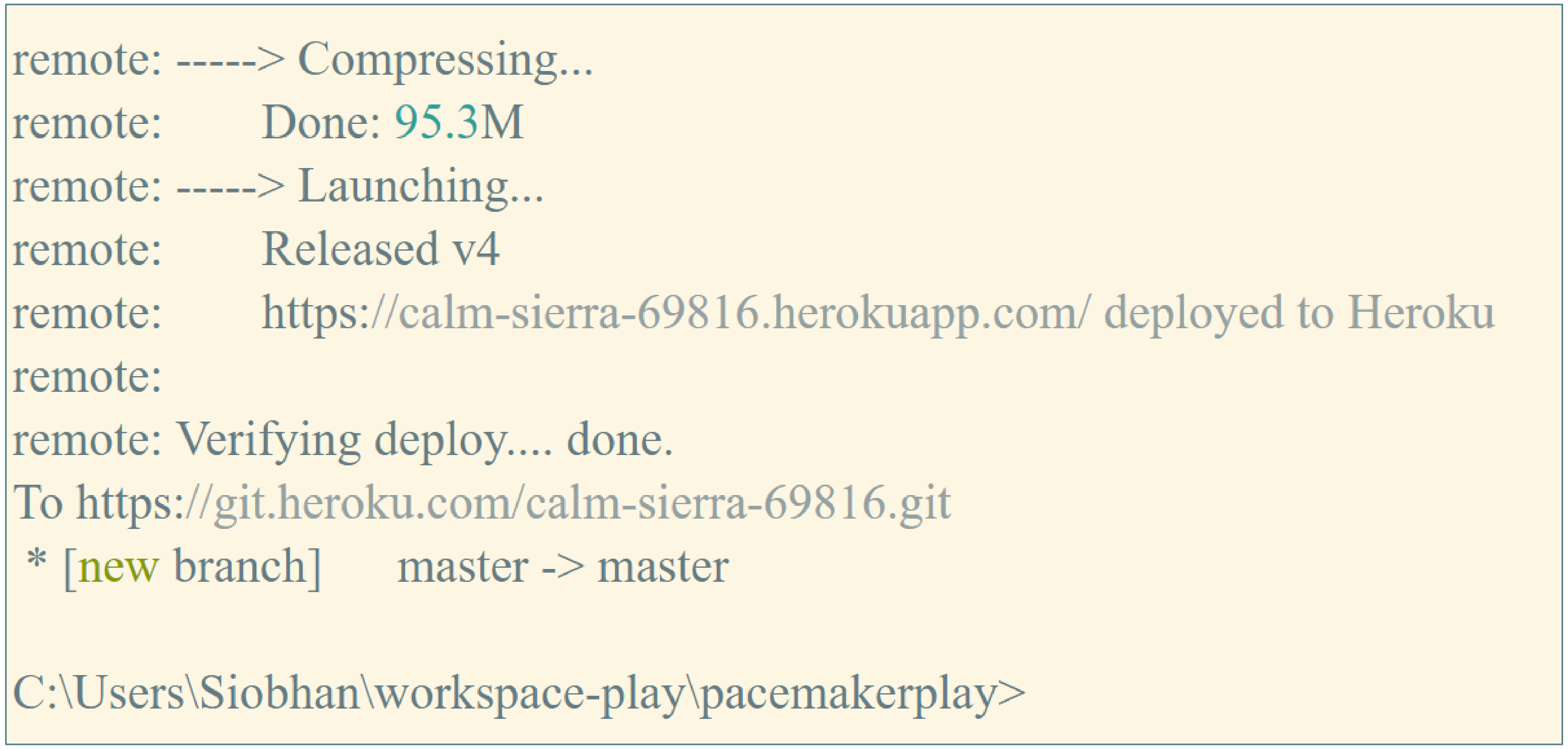
```
C:\Users\Siobhan\workspace-play\pacemakerplay>heroku create
Creating app... done, calm-sierra-69816
https://calm-sierra-69816.herokuapp.com/ | https://git.heroku.com/calm-sierra-69816.git

C:\Users\Siobhan\workspace-play\pacemakerplay>
```

Pushing the app to Heroku



```
git push heroku master
```

On the first push, there will be a LOT of console output from this command! But at the end, it should say something like this:

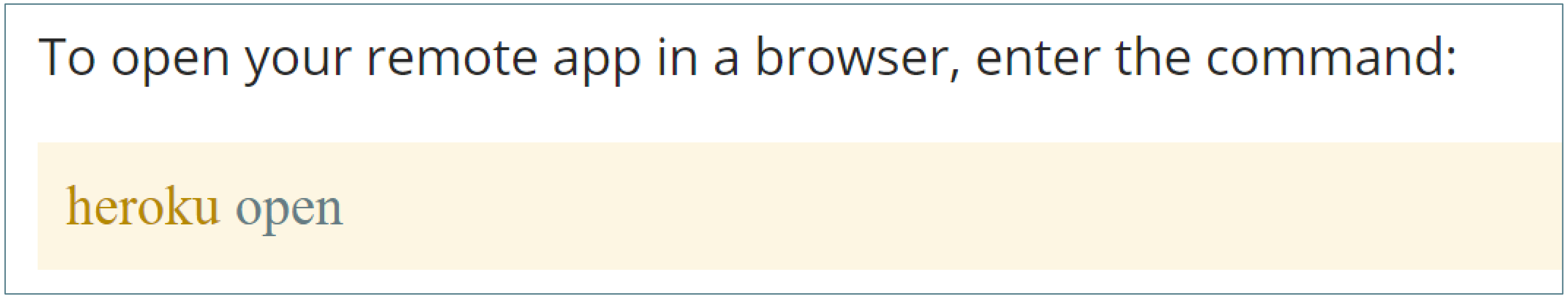


```
remote: -----> Compressing...
remote:        Done: 95.3M
remote: -----> Launching...
remote:        Released v4
remote:        https://calm-sierra-69816.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy.... done.
To https://git.heroku.com/calm-sierra-69816.git
 * [new branch]      master -> master

C:\Users\Siobhan\workspace-play\pacemakerplay>
```

More Git Shell Commands (heroku cli)

heroku create	Creates a new application on Heroku, along with a Git remote that must be used to receive your application source.
git push heroku master	All the committed changes that you made in your Git repository are local. You need to push them to the server.

Opening your Heroku app


To open your remote app in a browser, enter the command:

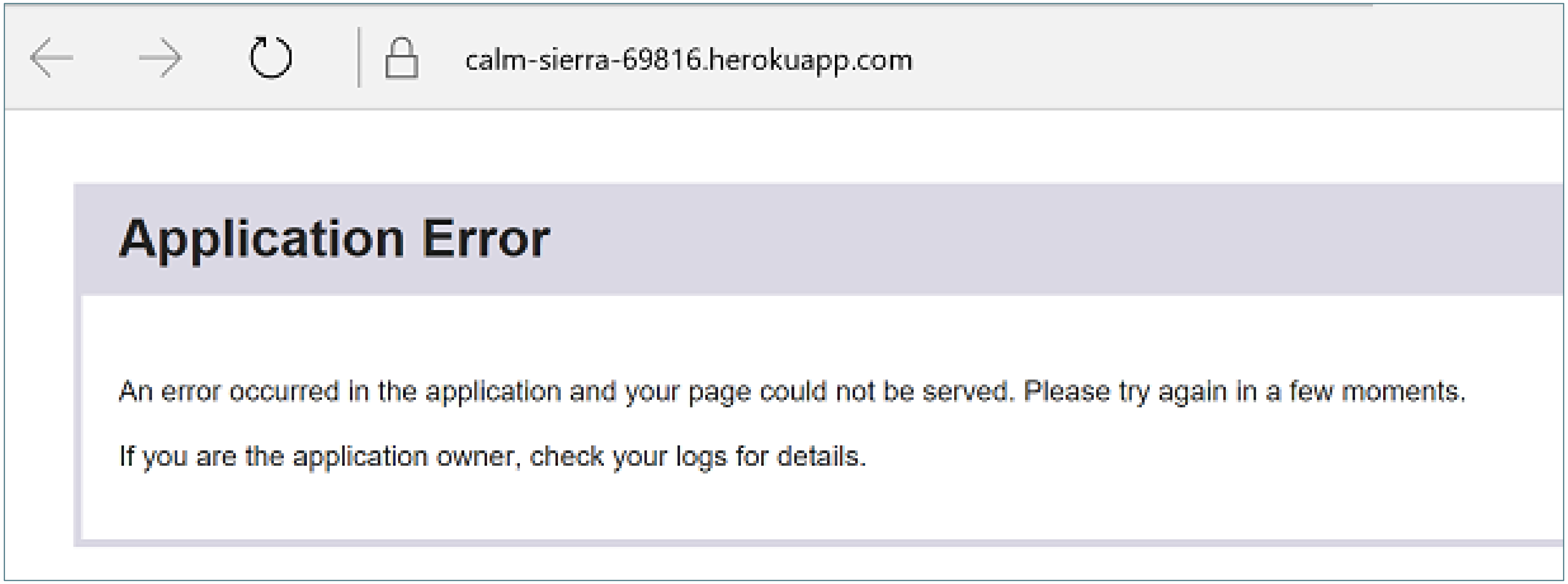
```
heroku open
```

However…

not all pushes to Heroku
will result in a
functioning remote app!!!!!

(even if your local version works perfectly)

Something went wrong!!!


calm-sierra-69816.herokuapp.com

Application Error

An error occurred in the application and your page could not be served. Please try again in a few moments.

If you are the application owner, check your logs for details.

On your command prompt for the project, this command will display the logs which can help you to isolate what went wrong:

```
heroku logs
```

Something went wrong!!!

- Play applications on Heroku are automatically provisioned with a Postgres database; the default local is h2.

- We are missing two things in our remote build:

  1. the PostgreSQL JDBC driver to your application dependencies (build.sbt).

  2. the Procfile which declares what commands are run by your application's dynos on the Heroku platform.

Dyno: Linux container that runs a single user-specified command; web dynos receive HTTP traffic from the routers.

1. Adding Postgres driver to build.sbt

```
name := """pacemakerplay"""

version := "1.0-SNAPSHOT"

lazy val root = (project in file(".")).enablePlugins(PlayJava)

scalaVersion := "2.11.7"

libraryDependencies ++= Seq(
   javaJdbc,
   cache,
   javaWs,
   "org.postgresql" % "postgresql" % "9.4-1201-jdbc41")
```

2. Procfile missing

- Create a **Procfile** in your project root directory (note upper case P for Procfile and no file extension) with the following:

```
web: target/universal/stage/bin/pacemakerplay
-Dhttp.port=${PORT}
-Ddb.default.driver=org.postgresql.Driver
-Ddb.default.url=${DATABASE_URL}
-Dplay.crypto.secret="thisisthesecretpleasechangeit"
```

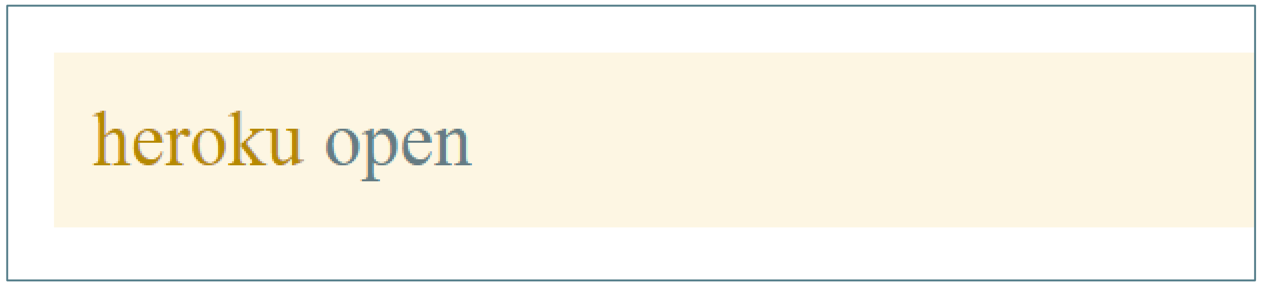
Push these changes to Heroku

On your command prompt for the project, enter these commands:

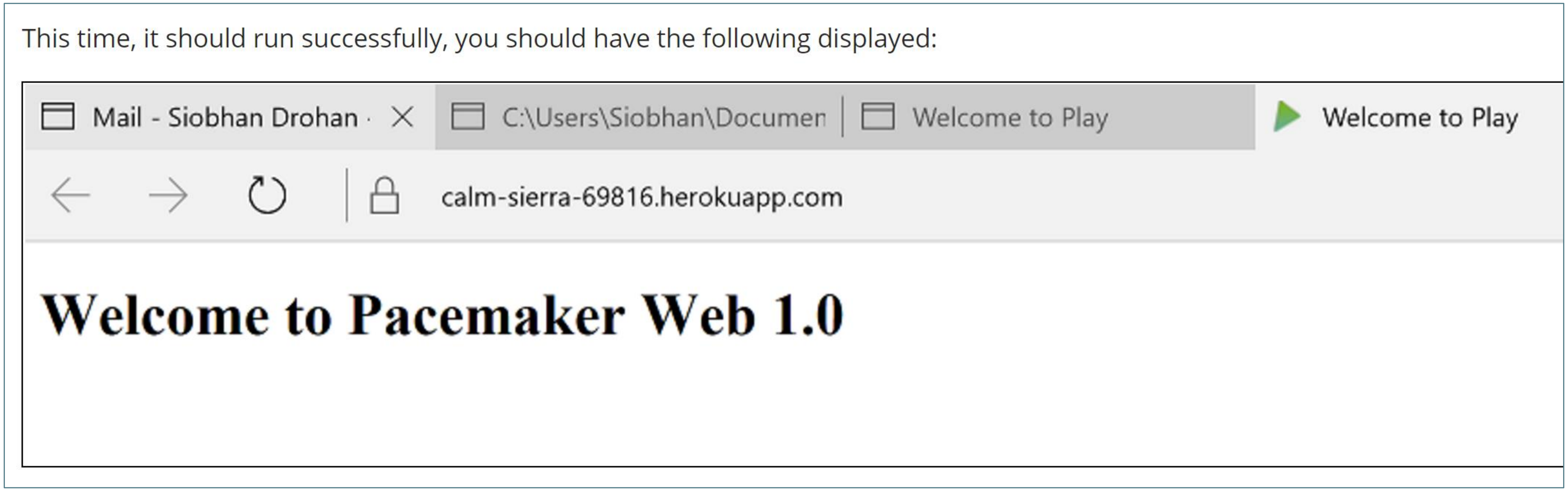
```
git add .
git commit -m “Adding Procfile.  Adding postgres dependency to build.sbt"
git push heroku master
```

Run your app again

On your command prompt for the project, enter this command:



```
heroku open
```



This time, it should run successfully, you should have the following displayed:

Welcome to Pacemaker Web 1.0

Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/