

Play Framework

Play Framework

Play framework概要

- オールインワンフレームワーク
DBアクセス層、キャッシュ、DB、テストツール、本番用サーバまで全て含まれている。
- Scala/Java両対応。
- 動的言語と同じ開発スタイル。
ソースコードを修正して、ブラウザの更新ボタンを押すだけ（コンパイルやデプロイが不要）。

Play framework概要

- 非JavaEE
 - JavaEE:
標準としてのメリットがある反面、歴史的経緯による複雑化、参加企業の多さからくるフットワークの悪さ。
 - ステートレス
Http sessionのようなサーバに紐付いた情報を保持しないため、JavaEEサーバのようなセッションステッキ、セッションマイグレーションで悩むことなくサーバを増やせばスケールする。
 - 貧血症(Anemic)ドメインモデルからの脱却
カプセル化、防御的コピー、不変式を意識した「正しい」オブジェクト設計。

Play framework概要

- 名だたるメンバーの参画
 - Martin Oderskey: Scalaの父
 - Rod Johnson: SprintSourceの父
VMWareを経てTypesafe社へ。
 - James Gosling: Javaの父
Googleを経てTypesafe社へ。
 - Doug Lea: Java並行処理ライブラリ開発者

Play framework超入門

Play framework超入門

Play framework超入門

- 準備
 - Javaをインストール
 - Playは <http://www.playframework.com/download> から入手。今回は、2.1.1を使用。
 - zipを展開して、中のplay(Linux, Mac)あるいはplay.bat(Windows)にパスを通す。

Play framework超入門

- プロジェクトを作成する
 - `play new play-example` を実行(play-exampleは任意のプロジェクト名)。

play! 2.1.0 (using Java 1.6.0_41 and Scala 2.10.0), <http://www.playframework.org>

The new application will be created in `/home/shanai/scala/play-example`

What is the application name? [play-example]

プロジェクト名の決定

>

Which template do you want to use for this new application?

- 1 - Create a simple Scala application
- 2 - Create a simple Java application

JavaかScalaを決定

> 1

OK, application play-example is created.

Have fun!

Play framework超入門

- 自動生成されたファイル、ディレクトリ

```
-rw-rw-r-- 1 shanai shanai 151 2月 5 09:51 README
drwxrwxr-x 4 shanai shanai 4096 3月 21 12:22 app
drwxrwxr-x 2 shanai shanai 4096 3月 21 12:22 conf
drwxrwxr-x 2 shanai shanai 4096 3月 21 12:22 project
drwxrwxr-x 5 shanai shanai 4096 3月 21 12:22 public
drwxrwxr-x 2 shanai shanai 4096 3月 21 12:22 test
```

- app
アプリケーション ソースコード
- conf
設定ファイル
- project
プロジェクト設定ファイル
- public
静的コンテンツ
- test
テストコード

Play framework超入門

- 実行してみる

```
shanai@shanai-ThinkPad-X61:~/scala/play-example$ play run  
Getting org.scala-sbt sbt 0.12.2 ...
```

```
...
```

```
[info] Done updating.
```

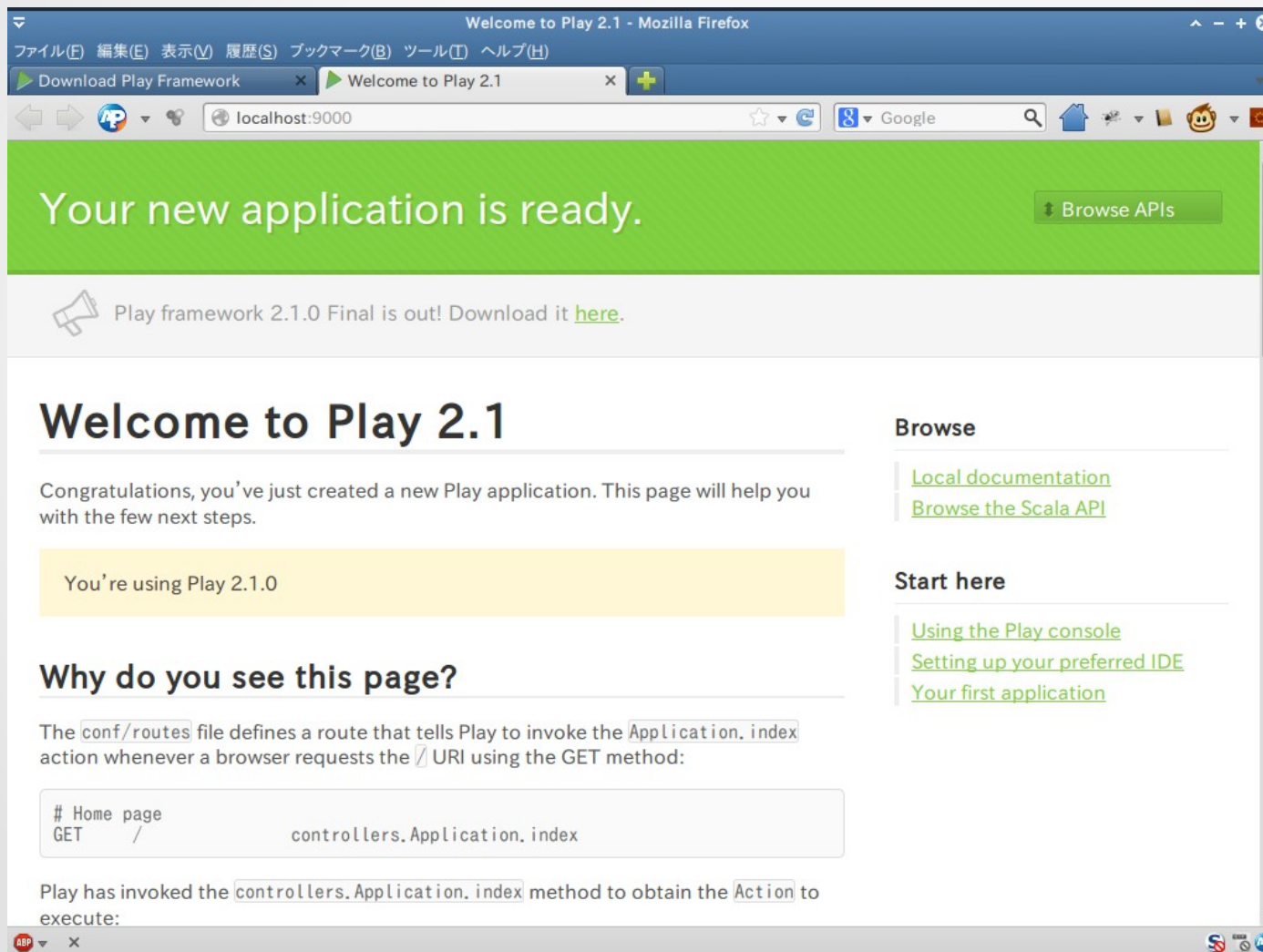
```
--- (Running the application from SBT, auto-reloading is enabled)
```

```
---
```

```
[info] play - Listening for HTTP on /0:0:0:0:0:0:0:0:9000
```

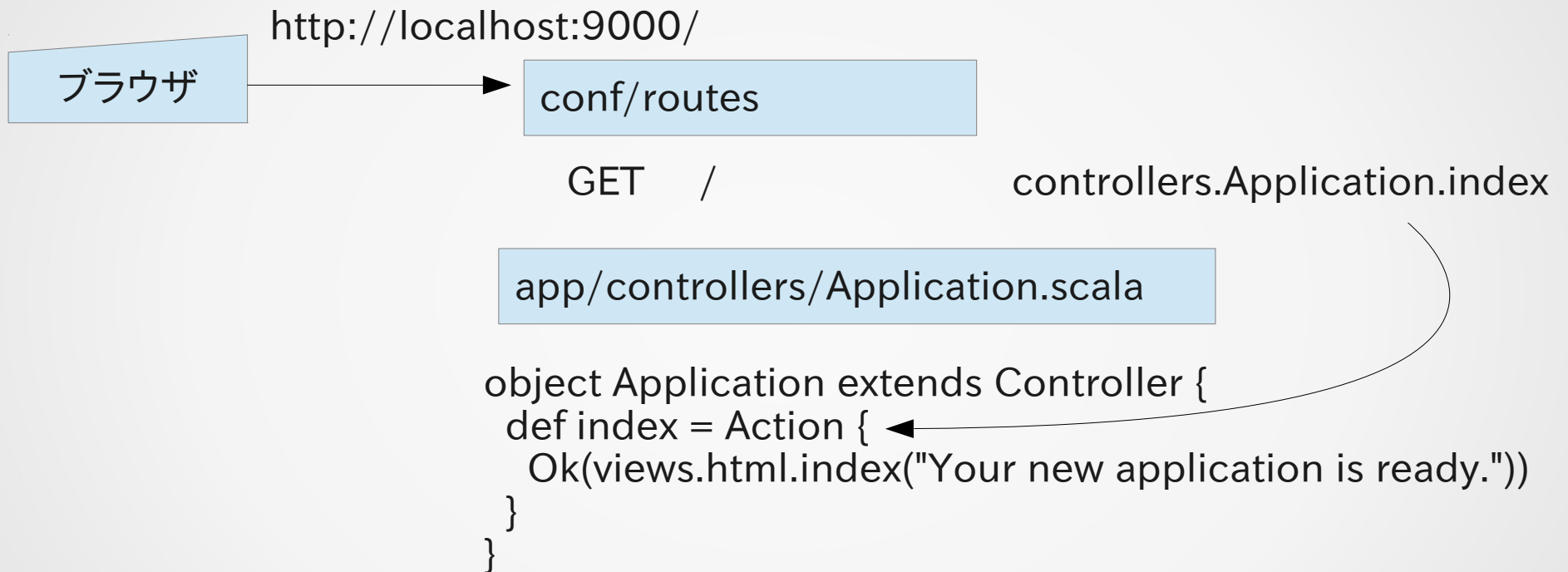
Play framework超入門

- `http://localhost:9000`をブラウザで開く



Play framework超入門

- リクエスト処理の流れ



Play framework超入門

- ビュー処理の流れ

app/controllers/Application.scala

```
object Application extends Controller {  
  def index = Action {  
    Ok(views.html.index("Your new application is ready."))  
  }  
}
```

app/views/index.scala.html

@(message: String) ←

```
@main("Welcome to Play 2.1") {  
  @play20.welcome(message)  
}
```

play_2.10.jarの中の関数

@を付けた部分はScalaの式

Play framework超入門

- mainの正体：

app/views/main.scala.html

@(title: String)(content: Html)

引数

<!DOCTYPE html>

<html>

<head>

<title>@title</title>

<link rel="stylesheet" media="screen"
href="@routes.Assets.at("stylesheets/main.css")">

<link rel="shortcut icon" type="image/png"
href="@routes.Assets.at("images/favicon.png")">

<script src="@routes.Assets.at("javascripts/jquery-1.9.0.min.js")"
type="text/javascript"></script>

</head>

<body>

@content

</body>

</html>

public/の静的コンテンツ
へのURLが返る。

関数呼び出しによってテンプレートを作成

Play framework超入門

- リクエストのRouting (conf/routes)

GET /clients/:id controllers.Clients.show(id: Long)

デフォルトパターンは[^/]+

関数の引数に渡せる。型も変換される

GET /assets/*file controllers.Assets.at(path="/public", file)

の場合のパターンは.

定数を渡すこともできる

GET /clients/\$id<[0-9]+> controllers.Clients.show(id: Long)

\$の場合は、パターンを指定

Play framework超入門

- リクエストのRouting (conf/routes)

GET /clients/ controllers.Clients.show(id: Long)

クエリストリングの”id”が引数に渡される。型も変換される

GET /clients controllers.Clients.list(page: Int ?= 1)

クエリストリングが無ければデフォルト値として1が使用される。

GET /api/list-all controllers.Api.list(Option[version])

Optionも使用できる。

Play framework超入門

Hello World
アプリケーション
(samples/scala/helloworld)

まとめ

- オールインワンのフレームワーク
- Java/Scala対応
- Httpのroutingには正規表現を使用
- 過度な抽象化が無い (Viewはhtmlが見える)