

Play で遊んでみる

Scala+Play 2.0 で Web アプリ開発入門を辿りながら .

まずは、インストール

play-2.2.3 をダウンロードして、/usr/local 以下に展開して

```
export PATH=/usr/local/play-2.2.3:$PATH
```

でパス通した .

実行してみる

```
play new sampleApp
```

で、プロジェクトを作ったら、そのプロジェクトディレクトリに移動して、play コンソールを起動 .

```
cd sampleApp  
play
```

プロンプトが表示されたら、

```
run
```

<http://localhost:9000> にアクセスするとデフォルトページがみえる、と .

初回は結構時間がかかった .

play コンソールの表示みていると

```
[info] 'compiler-interface' not yet compiled for Scala 2.10.3. Compiling...
```

とかでてるから、別途 Scala いらんのかな？

Eclipse プロジェクトにする

```
eclipse with-source=true
```

eclipsify じゃなかった . がりがりと、いろんなものがダウンロードされる .

Tepesafe Activator でも試す

ホームページには、Best Choise って書いてあるので、こっちも試してみる .

同じようにダウンロード、展開して /usr/local に移してパス通す .

```
activator new sampleApp2
```

とかすると、テンプレートを選ぶよう促されるので、hello-play を選ぶ。
で、

```
cd sampleApp2
activator
```

で、コンソールが起動してくる。

```
run
```

とすると同じように <http://localhost:9000> でアクセスできるようになる。

Eclipse で開発をする ... 前に

2.11 用の環境を構築していたのだけど、Play がひっばってきた 2.10 の Jar でエラーがでる。
とりあえず、2.10 の Eclipse 環境を用意。

のページの "For Scala 2.10.4" から Scala IDE for Eclipse をインストール。
<http://scala-ide.org/download/current.html>

Hello World 的な

- Application.scala の views.html.index の引数を変更してみる
- views/index.scala.html の中身を変更してみる

views/index.scala.html が、views.html.index として見えている、とのこと。

動く仕組みについて

- コントローラ : conf/routes で、HTTP リクエストのパスと Scala コードが関連づけられる

```
GET / controllers.Application.index
```

と書いてあることで、/ にアクセスすると controllers.Application.scala の index が実行される。

- モデル : Anorm というデータアクセスライブラルを使う (まだ使ってない)
- ビュー : Scala template を使うのがデフォルト。HTML に Scala を埋め込める、というものの。

プロジェクトの構成

Play 2.0 アプリの構造に説明があるとのこと。あとで読む。

play のコマンド

run とか test とか console とか . console で repl が立ち上がる .
プロジェクトの設定を変更したら reload ,依存ライブラリの更新チェック / ダウンロード / 設定は update .

Eclipse のデバッガとの連携

play コンソールをデバッグモードで起動 .

```
play debug
```

Eclipse の "Debug as" "Debug Configuration" で Remote Java Application なデバッグ設定を作る .
ホストは localhost ,ポートは 9999 . 9999 は play debug を実行したときにターミナルに表示されて
る .

Web ページ (<http://localhost:9000>) にアクセスすると ,
ブレークポイントにひっかかったところで Eclipse がデバッグモードになる .

Action とは ?

Action は , " リクエストを処理してクライアントへ送るレスポンスを生成するためのクラス " らしい .

で , たとえば , Ok 以外には のようなものがある .

ステータス	関数
200 OK	Ok()
404 NofFound	NotFound()
409 Conflict	Conflict()
500 Internal Server Error	InternalServerError()

また , Redirect(URL) でリダイレクトさせられる .

routes の定義でいろいろ指定する

- /:id とかすると id を変数で取り出せる
- 指定しないで , メソッドに引数 (たとえば id) がある場合 ?id=value なパラメタを渡せる
- ワイルドカードとか正規表現とかも使える
- メソッドの引数を定数にしたり , デフォルト値をセットしたりもできる