

***play!***



# O que é?

- Alternativa limpa e eficiente para aplicações web
- Seu principal objetivo é facilitar muito o desenvolvimento de aplicações Java / Scala
- Visa a produtividade dos desenvolvedores
- Solução muito boa para o desenvolvimento ágil para web

# *Quais as vantagens?*

- Limpo, Rápido e Dinâmico
- Produtivo, Poderoso e Divertido!
- Puro Java (Permite a importação e uso de qualquer biblioteca Java)
- Utiliza o sistema de template Groovy (Groovy trabalha de forma transparente com todos os objetos e bibliotecas Java existentes e compila diretamente para Java bytecode tanto no desenvolvimento de aplicações como no modo script.)
- Utiliza sistemas de tradução de Rotas
- Facilidade na resolução de erro

# Quais as vantagens?

- Assíncrono (I/O não bloqueante)
- Stateless
- Arquitetura RESTful ( REST (ou Representational State Transfer) é um estilo de desenvolvimento para sistemas distribuídos que privilegia a transferência de estado sem envolver camadas de mapeamento (como o SOAP faz). Assim, os verbos GET, POST, PUT e DELETE mapeiam quase que diretamente para ações READ, UPDATE, CREATE e DELETE (CRUD).
- Persistência com JPA (JPA é uma especificação de Persistência da plataforma Java, para persistir as classes java em bancos de dados relacionais. )
- Extensível (Através de Módulos / Bibliotecas Java)
- TDD - Test Development Drive (Test Driven Development (TDD) ou em português Desenvolvimento dirigido por testes )



# *Instalação - Requisitos*

- Ambiente Windows / Linux / Mac
- JDK e JRE 5 ou superior
- Editor de Texto (notepad++, Textmate, Vim)
- IDE Java - Eclipse ou Netbeans (Opcional)

# *Instalação – Ambiente Unix*

- Baixar Framework
- Extrair em uma pasta desejada (Ex:/opt)
- Adicionar o caminho no \$PATH



# *Instalação – Ambiente Windows*

- Baixar Framework
- Extrair em uma pasta desejada (Ex:/ C:\)
- Adicionar o caminho no \$PATH



# *Criando e rodando uma Aplicação*





# *Criando e rodando uma Aplicação*

Criando:

- `play new swxapp`

Rodando:

- `play run swxapp`

# Criando e rodando uma Aplicação

Play framework, 1.0.3 - [Documentation](#) - [Browse API](#)



## Your application is ready !

Congratulations, you've just created a new play application. This page will help you in the few next steps. [Click here to get more informations about your application environment.](#)

### Why do you see this page ?

The `conf/routes` file defines a route that tell play to invoke the `Application.index` action when a browser requests the `/` URI using the GET method:

```
# Application home page
GET      /           Application.index
```

So play has invoked the `controllers.Application.index()` method:

```
public static void index() {
    render();
}
```

Using the `render()` call, this action asks play to display a template. By convention play has displayed the `app/views/Application/index.html` template:

```
#{extends 'main.html' /}
#{set title:'Home' /}

#{welcome /}
```

This template extends the `app/views/main.html`, and uses the `#{welcome /}` tag to display this welcome page.

# *Configurando o projeto para IDE's*

Para o Eclipse

- `Play eclipsify swxapp`

Para o Netbeans

- `Play netbeansify swxapp`

# Definindo o Banco de Dados arquivo `conf/application.conf`

```
# Database configuration
# -----
# Enable a database engine if needed.
#
# To quickly set up a development database, use either:
# - mem : for a transient in memory database (HSQL in memory)
# - fs  : for a simple file written database (HSQL file stored)
# db=mem
#
# To connect to a local MySQL database, use:
# db=mysql:user:pwd@database_name
#
# If you need a full JDBC configuration use the following :
# db.url=jdbc:postgresql:database_name
# db.driver=org.postgresql.Driver
# db.user=root
# db.pass=secret
#
%
% db:base=SECRET
```