

導入方法・使い方・特徴などを記述

- ・ Spring Framework

Spring FrameworkはWEBアプリ開発に用いるWEBフレームワークである。WEBシステムやWEBサービスを作るならこれだ！！

Frameworkは1つのフレームワークではなく、たくさんのフレームワークの集合体である。

特徴の1つに「拡張性が高い」ということが挙げられる。必要な機能だけを外部から使うことができるので、無駄がなく、また必要となる機能がある場合は、あとから簡単に追加できる。このようにオブジェクト指向の基本原則を徹底して、Javaを最大限生かすことができるように設計されている。

特徴の2つめは「保守性が高い」ことだ。「AOP (Aspect Oriented Programming)」というプログラミングを用いる事ができる。クラスに「本質的な処理」のみを書いて、「共通化できる処理」を別を書く。主に必要となる処理とそうではない処理を分けることで、コードをわかりやすくする効果がある。それのおかげで不具合の原因を特定しやすくなる。

特徴の3つめは「再利用性が高い」ことだ。

「AOP」の仕組みによって、共通のプログラムをまとめやすくなるため、再利用することが容易となる。再利用性の高いプログラムを、プログラマの腕ではなく仕組みで解決できる。

- ・ Play Framework

Play Frameworkは「Java」と「Scala」を使って作られたWEBフレームワークである。

バージョン2からは主に「Scala」言語に書き換えられているが、バージョン1は「Java」で主に作られていた。Rubyの「Ruby on Rails」やPythonの「Django」から大きな影響を受けているため、非常に似ている点が多い。他のフレームワークよりも軽量なため、利用しやすくJava初学者が基礎学習を覚えた後に使い方をすぐにでも覚えたほうがよいフレームワークと言われている。

特徴の1つとして「高速に開発できる」という事が挙げられる。MVCという設計に基づいて、WEBアプリ開発を高速で行える。Javaはコンパイルが必要であるため、「コンパイルの遅さ」が弱点となりがちであったが、Play Frameworkではコードの追加や修正が即座に反映され、なおかつJavaのメリットである「堅牢性」及び「高速な動作」を最大限活かせるようになっている。また、最新技術に幅広く対応しているため、どんな開発でも最も効率の良いものを選べる。また、「リソース消費が少ない」事も特徴であり、とにかく無駄がなくCPUやメモリーに負担が少ないので、少ないリソースでアプリを動作させることができる。

- ・ JSF (JavaServer Faces)

上の二つのフレームワークと同様にWEBフレームワークである。JSFの仕様を考える中心人物は「Struts」の開発者でもあるため、JSFとStrutsは重複する機能が多い。

その特徴は、JavaServer Facesに沿って開発したユーザインタフェースUIを開発環境で利用することで、GUIによるWebアプリケーション開発が可能となる点。

クライアントで発生したイベントを受けて、サーバ側で処理を実行する「イベントモデル」を提供できる点。

JavaServer Facesのフレームワークを利用して、独自のUIを開発する事ができる点である。