

Spring Framework

概要: JavaのWEBフレームワークで、WEBアプリケーションを作ることができる。特に、WEBシステム、WEBサービスの開発に向いている。

特徴

変更に強い

DI(Dependency Injection)という仕組みが導入されている。

Javaを使ったアプリ開発は大規模なものが多く、変更や修正が入ることが多いが、DIにより依存性のあるプログラムは外部から取り入れるようになっているため1つ1つのプログラムを独立させやすい。したがって、変更が必要な場合でも最低限の変更で済むので開発が楽。

テストが簡単

Spring MVC Test」という専用のテストプログラムがある。これは1つ1つの処理がうまくいっているかをテスト(単体テスト)するのではなく、無事に処理を終えるかをテスト()するもの。バグの混入を最小限に抑え、安全に開発を進めることができる。

拡張性が高い

フレームワークの基本的な機能の大部分が「インターフェイス」として提供されている。必要な機能だけを外部から使うことができるので、無駄がない。後から機能の追加も簡単。オブジェクト指向の基本原則を徹底して、Javaを最大限に生かすことができるように設計されている。

保守性が高い

AOP (Aspect Oriented Programming) というプログラミングを使うことができる。
これは、クラスには本質的な処理のみを書いて、共通化出来る処理を別を書くという風に使える。
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。
したがって、バグやエラーの原因を特定しやすい。

再利用性が高い

AOPの仕組みにより、共通のプログラムをまとめやすいため、再利用も簡単にできる。
したがってプログラムを書く量が減り、修正も最低限で済む。
特に長期アプリ開発においてもものすごく効率化できる。

Play Framework

概要: JavaとScalaを使って作られたWEBフレームワーク。WEBシステム、WEBサービスの開発に向いている。高速。

特徴

高速開発

軽量で動作が速い。

「MVC」に基づいて開発を行うことができるため、WEBアプリ開発を高速で行うことができる。

このフレームワークは、JavaのWEBアプリ開発の弱点だった「コンパイルの遅さ」も解消されているので、コードの追加や修正が即座に反映されるようになっている。

Javaのメリットである、堅牢性や高速な動作を最大限活かせるようになっている。

リソース消費が少ない

CPUやメモリに負担が少なく、少ないリソースでアプリを動作させることができる。つまり、それだけ無駄がなく速いということ。

大規模まで耐えられるアプリを作ることができるため、将来のことを心配せずに開発を行える。

高い拡張性

リアルタイム通信やテストに関するもの、また幅広い拡張言語など幅広く対応しているため、開発において最も効率の良いものを選ぶことができる。そのため、開発をスムーズに行うことができる。

Apache Struts

概要: Apacheソフトウェア財団によって開発されているMVCを採用したWebアプリケーションフレームワーク。2000年に初期バージョンが開発され、現在はStruts2もリリースされている。

特徴

開発効率

HTMLとコードを分けるため、見やすく綺麗なコードになる。したがってスムーズに開発、修正を行うことができる。また、ActionServletというものが用意されており、画面の遷移をコントロールする設定ファイルを変更するだけで遷移先の変更が可能。

ApacheStruts2ではアノテーションによる設定ファイルの削減や、DIコンテナ機能(アプリケーションにDI機能を加える機能)による外部ファイルでクラスの依存関係を設定できるなどの改良が加えられている。