

XML MUST DIE!

Make Seasar more Groovy.

Seasar のからさわぎ
2004 年 4 月 10 日 (土)
渋谷区初台シモモトビル B1

TAKAI Naoto <takai@commentout.com>

アジェンダ

- XML 死すべし！
- Seasar の宿命
- 救世主 groovy-seasar
- Groovy の正体
- XML との決戦
- groovy-seasar の秘められた能力
- さらなる飛躍を目指して……

XML 死すべし！

- 可読性
 - 計算機が読みやすいだけ
 - 計算機間で設定ファイルの交換なんてしない
- 拡張性
 - XML 関連の仕様っていくつおぼえてます？
 - 名前空間なんて難しすぎて使えない
- 構造
 - 樹木型のデータ構造を表現
 - if {...} else {...} すら表現できない

Seasar の宿命

- Dependency Injection
 - 依存関係は Seasar が解決する
- 依存関係を設定ファイルで定義
 - 依存関係の構造が設定ファイルにもちこまれる
 - それってプログラムで表現されていたんだよね？
- 設定ファイルこそがキモ！
 - 一目瞭然のシンプルな構造
 - 柔軟な拡張性
 - プログラムのスイッチングコスト

救世主 groovy-seasar

- Groovy を利用した S2Container の定義
 - 構造
 - 基本は XML による定義と同じ雰囲気
 - スクリプトとマークアップのマリアージュ
 - 拡張性
 - Groovy の機能をフルに活用できる
 - スイッチングコスト
 - 式言語 Sel をあたらしく覚える必要はない
 - 文法は Java とほとんどかわらない

Groovy の正体

- Java + {Python, Ruby, SmallTalk}
 - Closure だいすき！
 - 動的型付け言語
 - Java との親和性が高い
- JSR 241
 - Java 公認スクリプト言語
- GroovyMarkup
 - Closure でツリー構造を表現

XML との決戦

```
<components>
  <component name="foo" class="java.lang.Integer">
    <arg>0</arg>
  </component>
  <component class="java.util.Date">
    <property name="time">foo</property>
  </component>
</components>
```

```
builder.components(){
  component(name: "foo", class: Integer){
    arg("0")
  }
  component(Date){
    prop(name: "time", ref: "foo")
  }
}
```

groovy-seasar の秘められた能力

- 実は XML での設定よりも先に実装
 - むしろ標準
- Groovy の機能が利用できる
 - 制御構造
 - 条件分岐、繰り返し、例外処理
 - 外部リソースの参照
 - データベース、プロパティファイルを参照
- ライセンスは Seasar と同じ
 - 宣伝条項が増えない！

さらなる飛躍を目指して……

ご静聴ありがとうございました！
なにかご質問は？