Cubby

Webアプリケーションのためのシンプルなフレームワーク

BABA Yasuyuki
<baba@nulab.co.jp>

自己紹介

- 馬場保幸 (ばば やすゆき)
- 株式会社ヌーラボ (http://www.nulab.co.jp)

本日のアジェンダ

- Cubbyって何？
- Cubbyを使うと幸せになるポイント
- Maven2と連携した簡単なデモ

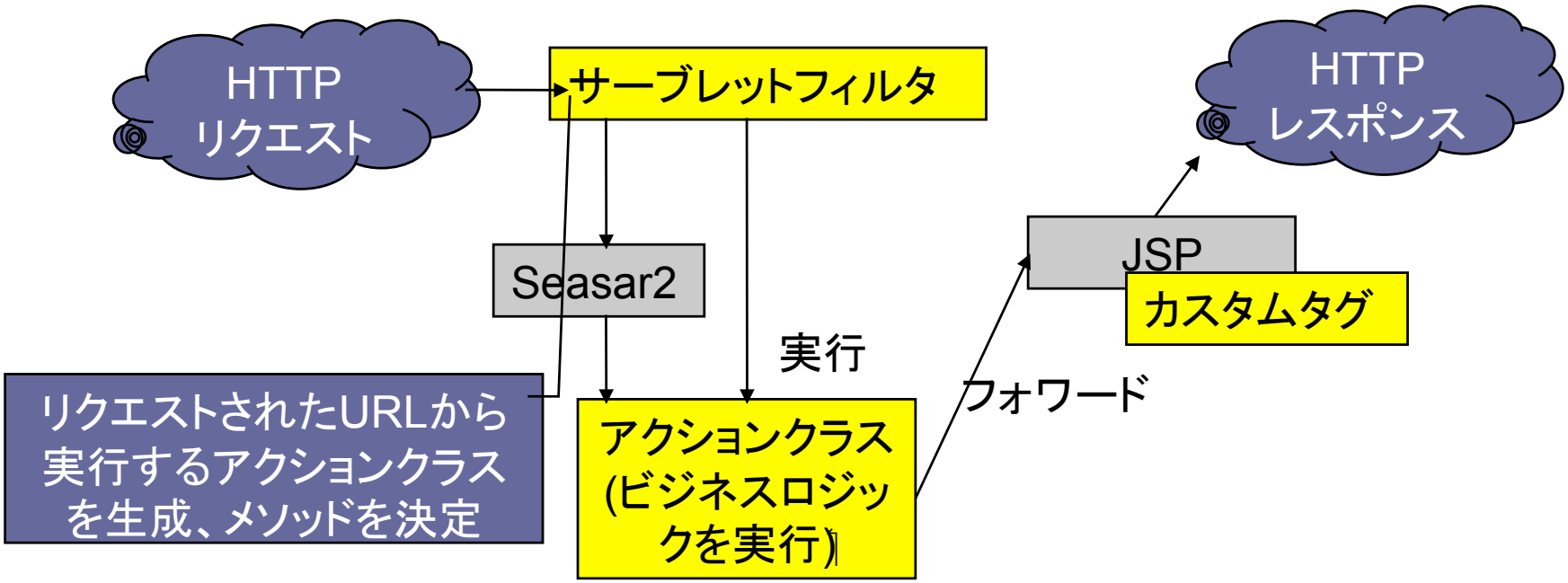
Cubbyって何？

- JavaでWebアプリケーションを開発するためのシンプルなフレームワーク
- 現在のバージョンは0.9.2
  - 近日中の1.0リリースを目指して、seasar.orgのsandboxで開発中
- Cubby = 小さな整理箱

Cubbyって何？

- http://cubby.sandbox.seasar.org/

アーキテクチャ


HTTP
リクエスト
サーブレットフィルタ
HTTP
レスポンス
Seasar2
JSP
カスタムタグ
実行
リクエストされたURLから
実行するアクションクラス
を生成、メソッドを決定
アクションクラス
(ビジネスロジッ
クを実行)
フォワード

Cubbyを作った理由

- Strutsの設定ファイルにうんざり
- きれいなURIのアプリケーションを作りたい
- 作りたかったから

習得が簡単

Cubbyを使うと幸せになれるポイント その1

HTMLに近いJSPカスタムタグ

- Strutsの場合

```
<html:text property="userName"/>
<html:select property="type">
  <html:optionsCollection
    property="choices"
    value="id"
    label="name"/>
</html:select>
```

HTMLに近いJSPカスタムタグ

- Cubbyの場合

```
<t:input type="text" name="userName" />
<t:select name="typeId"
  items="${action.todoTypes}"
  labelProperty="name"
  valueProperty="id"/>
```

- HTMLタグに近い構文なので、直感的

JSPカスタムタグの種類が少ない

- カスタムタグは5つです。
  - t:form
  - t:input
  - t:select
  - t:textarea
  - t:token
- 種類が少ないので、覚えることも少ない

設定ファイルいらず

Cubbyを使うと幸せになれるポイント その2

メタ情報の記述

- Strutsの場合
  - 設定ファイル struts-config.xml にまとめて記述

```
<form-beans>
  <form-bean
    name="todoForm”
    type="example.form.TodoForm" />
  </form-beans>
  <action-mappings>
    <action
      path="/todoEdit.do”
      type="ex.action.TodoEditAction“
      name="todo">
      <forward name="success" path="/todo/edit.jsp" />
    </action>
  </action-mappings>
```

メタ情報の記述

- Cubbyの場合
  - アノテーションを用いてアクションクラスのメソッドに記述

```
public class TodoAction extends Action {
  public ValidationRules editValidation =
    new DefaultValidationRules(“todo.") {
      @Override
      public void initialize() {
        add("userId", new RequiredValidator());
        ...
      }
    };
  @Validation(rules="editValidation",
    errorPage = "/todo/edit.jsp")
  public ActionResult edit() {
    ...
    return new Forward("/todo/confirm.jsp");
  }
}
```

アノテーションの種類

- @Path
  - アクションにアクセスするためのパス(URL)
- @Accept
  - アクションが受け付けるHTTPメソッド(GETやPOSTなど)
- @Form
  - リクエストパラメータを指定するためのオブジェクト(デフォルトはアクション自身)
- @Validation
  - バリデーションを行うオブジェクト

- ソースコード上に記述するため、ソースコードを見るだけでどのような動作をするかが明確になる

きれいなURI

Cubbyを使うと幸せになれるポイント その3

http://d.hatena.ne.jp/m-hashimoto/20071231/

http://ja.wikipedia.org/wiki/REST

https://backlog.backlog.jp/view/BLG-35

Cool URI

- Cool URI
  - http://www.w3.org/Provider/Style/URI.html
  - http://www.kanzaki.com/docs/Style/URI.html
- 一意のリソースを表すURI
- ウェブサイトを構築する際には、きれいなURIを設計することが重要
- なにせ気持ちがいい

mod_rewriteの場合

- httpd.conf

```
RewriteRule ^\/todo\/([0-9]+)? /todo.do?id=$1
```

Ruby on Railsの場合

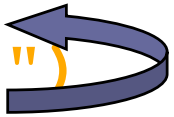
- config/routes.rb

```
ActionController::Routing::Routes.draw do |map|
map.connect '/todo/:id',
 :controller => "todo", :action=> "show"
end
```

Cubbyの場合

- アノテーションを用いてアクションクラスのメソッドに記述していく

```
// http://example.com/todo/1
public class TodoAction extends Action {
   public String id;
   @Path("/todo/{id}")
   public ActionResult show() {
      System.out.println(id);
   }
   // 正規表現も使えます
   // @Path("{id,[0-9]+}")
}
```



そのほか便利な機能

Cubbyを使うと幸せになれるポイント その4

Mayaaとの連携

- HTMLテンプレートエンジンMayaaと連携することもできる
  - login.html

```
<form id="form">
ユーザID:<input id="userId" type="text" name="userId" />
パスワード:<input id="password" type="password"
  name="password"/>
</form>
```

  - login.mayaa

```
<t:form m:id="form" action="${contextPath}/todo/login/
  process" method="post" value="${action}"/>
<t:input m:id="userName" type="text" name="userName" />
<t:input m:id="password" type="text" name="password" />
```

Maven2との連携

- Maven2 Archetype
  - プロジェクトのひな形を作成する仕組み
  - StrutsのblankやRailsのrailsコマンドのようなもの
  - Cubbyによる**実際に動作するプロジェクト**が生成される
    - Cubbyが依存するライブラリが設定されたpom.xml
    - 基本的なパッケージ構成のディレクトリ
    - Hello World

Maven2との連携

- ここで、実際にプロジェクトを作成してみます。

```
$> mvn archetype:create
 -DgroupId=(作成するプロジェクトのグループID 例:com.foo.bar)
 -DartifactId=(作成するプロジェクトのアーティファクトID
  例:barapp)
 -Dversion=(作成するプロジェクトのバージョン 例:1.0-SNAPSHOT)
 -DarchetypeGroupId=org.seasar.cubby
 -DarchetypeArtifactId=cubby-archetype
 -DremoteRepositories=http://maven.seasar.org/maven2/
$> mvn eclipse:eclipse
$> mvn tomcat:run
```

Cubbyは・・・

まとめ

Cubbyは

- シンプルで習得が容易です。
- きれいなURIが得意です。
- まずは、1度触ってみてください。

今後の予定

- Maven2との連携
  - URI一覧などのドキュメント出力
  - Scaffold
- サンプルアプリケーションの充実
- **シンプルであり続けること**

ありがとうございました

- ご意見、ご要望をお待ちしております。
  - https://ml.seasar.org/mailman/listinfo/cubby-user