

Apache Wicket

Wicketってなに

- http://www.wicket-ja.org/about_wicket
- <http://www.ibm.com/developerworks/jp/web/library/wa-aj-wicket/>

オブジェクト指向

- Wicket は Java のオブジェクト指向言語としての機能を最大限に活用。
- Struts のような設定ファイルを持たない。唯一の設定ファイルは「web.xml」
- プログラマは自由にページを「継承」したり、機能を「委譲」したりすることができる

HTML をそのまま使う

- JSP のようなページ・テンプレートも否定
- HTML ファイルをそのままテンプレートとして使用
- HTML にもともと存在する属性を Wicket が使うこともほばない
- 利用するのは、HTML エディタが無視してくれる、独自の「wicket:id」属性だけ
- HTML ファイルを JSP に変換するための余計な労力は発生しない

コンポーネント指向

- ページをオブジェクトとして扱う。
- ページそのものが複数の「コンポーネント」の組み合わせでできていると考える
- 「テキストフィールド」や「ラベル」「ラジオボタン」という各要素もコンポーネントの一種
- コンポーネントはただのオブジェクトであって、入力補完をしてくれるフィールドや、日付選択用のボタンのついたフィールドなども「コンポーネント」
- 複数のコンポーネントを組み合わせた、例えば「ログインフォーム」「コメントフォーム」などもコンポーネント。複数の画面で再利用可能
- Ajax 対応コンポーネントをページに追加するだけで、Ajax 化されてウェブページを作ることができる。

ステートフル

- アプリケーションには状態があるが、HTTP という通信プロトコルには本来「状態」という概念がない。
- サーブレットはこの問題を上辺だけで解決してきた。管理はすべてプログラマまかせ。
- Wicket はページの状態をフレームワークが保持します。入力値やページの変更も「状態」として保持
- Wicket ではページの状態を Wicket が保持するから、ページに単純に値を保持できます。
- ページを「オブジェクト」として作ることができる
- サーブレットはサーブレット内に値を持つこと自体が禁止されていたのと対照的（状態はすべて「セッション」に保持して自分で管理しなければいけなかった）