

Apache Wicket

Prezentace web frameworku

Jiří Hradil, Kyberie s.r.o.
(jiri.hradil@kyberie.cz)

Hlavní vlastnosti

[WEB]

- Framework pro webové aplikace
- Nepoužívá XML pro konfiguraci
- (Téměř) čisté HTML šablony
- Komponentový model podobný Swingu/AWT
- Neobjevuje objevené, deleguje
- Stabilní verze 1.3.5
- Vývojová verze 1.4-rc2

Odstínění od HTTP protokolu

[WICKET IN ACTION]

- Poskytuje programový model, ve kterém se snaží odstínit od nevýhod http stateless protokolu
- Hlavní cíl Wicketu: transparentní state management, pomocí kterého tvoříme web aplikace, jako bychom je psali lokálně
- Výsledek: programujeme "lokálně" a šablony pro web píšeme v klasickém HTML

Ukázka – Java Server Pages

[WICKET IN ACTION]

```
<table>
  <tr>
    <c:forEach var="item" items="$sessionScope.list">
      <td>
        <c:out value="item.name" />
      </td>
    </c:forEach>
  </tr>
</table>
```

Ukázka – Apache Velocity

[WICKET IN ACTION]

```
<table>
  <tr>
    #foreach ($item in $sessionScope.list)
      <td>
        ${item.name}
      </td>
    #end
  </tr>
</table>
```

Ukázka – Java Server Faces

[WICKET IN ACTION]

```
<h:dataTable value="#{list}" var="item">
  <h:column>
    <h:outputText value="#{item.name}" />
  </h:column>
</h:dataTable>
```

Ukázka – Apache Wicket

[WICKET IN ACTION]

```
<table>
  <tr>
    <td wicket:id="list">
      <span wicket:id="name" />
    </td>
  </tr>
</table>
```

Wicket: HTML šablony

- prezentační kód píšeme v čistém HTML
- minimum speciálních atributů či tagů
- šablony může dělat web designer, který o Javě neví zhola nic
- snižuje se nebezpečí, že šablona bude obsahovat část aplikační logiky
- nepotřebujeme v šabloně žádnou procesní logiku!

Provázání HTML a Java třídy

- Konvence před konfigurací
- Šablona sdílí název se třídou:
 - `Priklad1.java`
 - `Priklad1.html`
 - `Uzivatele.java`
 - `Uzivatele.html`
- HTML se ukládají přímo do java balíků
- HTML jsou pak ve war pod WEB-INF

Web aplikace v objektech

- Celá aplikace je instancí třídy **Application**
 - Konfigurace aplikace
 - Globálně sdílené atributy či metody
- Stránka je instancí třídy **Page**
- Prvky na stránce (ty, které potřebujeme mít dynamické) jsou Wicket objekty
- (Opakování) hierarchie objektů je podobná Swingu/AWT
- Wicket hodně používá dědičnost

...a teď trochu praxe

- Nejjednodušší start pomocí Apache Maven2
- NetBeans IDE s Maven pluginem umí stáhnout archetyp přímo pro Wicket (pozor na verzi)

`pom.xml: ...`

`<dependency>`

`<groupId>org.apache.wicket</groupId>`

`<artifactId>wicket</artifactId>`

`<version>1.3.5</version>`

`</dependency>`

`...`

Příklad 1: Nahrazení textu

- Do HTML tagu vložíme atribut **wicket:id**
- Pomocí **wicket:id** provážeme tag s Java objektem v pozadí

HTML:

```
<h1 wicket:id="pozdrav">vypsát pozdrav</h1>
```

Objekt v Javě:

```
new Label("pozdrav", "Ahoj CZJUG!");
```

- Wicket nepovolí křížení HTML tagů!
- viz. [Příklad1.java](#)

Příklad 2: Odkaz a objekt stránky

- Vytvoření odkazu, aktualizujícího stránku
- Protože je stránka objekt, drží si hodnoty svých atributů
- Nemusíme řešit konverzaci – je automaticky stateful
- Wicket si ukládá vygenerované stránky do mapy a tato mapa je svázaná se session přihlášeného uživatele
- Každá vygenerovaná stránka se serializuje
- **Viz. Příklad2.java**

Příklad 3: Odkazy mezi stránkami

- Odkazovat lze rovnou vytvořením instance cílové stránky nebo na třídu stránky
- Nemusíme se vůbec starat o kódování/dekódování atributů v URL
- Protože je každá stránka serializovaná, lze bez problémů procházet historii
- Viz. [Příklad3_1.java](#)

Modely I.

[WICKET IN ACTION]

- Modely umožňují Wicket komponentám načíst data, která jsou třeba pro vykreslení obsahu komponenty
- Do modelu se také ukládají data, zadaná uživatelem
- Model ve Wicketu je Model dle MVC
- Model drží stav komponenty

Modely II.

[WICKET IN ACTION]

- **Model** – jednoduchý model pro uchování statického obsahu
- **PropertyModel** – dynamický přístup k atributům doménového objektu
- **CompoundPropertyModel** – automaticky sváže název komponenty se stejným atributem v doménovém objektu
- **LoadableDetachableModel** – neserializuje objekt v modelu, ideální pro načítání dat z DB
- ...a další implementace...

Příklad 4: Formuláře

- Formulář provážíme s modelem, reprezentující doménový objekt
- Formuláře opět nepotřebují speciální tagy, stačí známý **wicket:id**
- Validátory v HTML šablonách neřešíme, jsou to samostatné objekty na úrovni javovských Wicket komponent
- Formulář může hlásit chyby pomocí objektu **FeedbackPanel**
- Viz. **Příklad4_1.java**

Validátory

- Validátory kontrolují obsah zadaný uživatelem dle vynucené formy
- Mnoho validátorů je ve standardní distribuci:
 - `UrlValidator`
 - `EmailAddressValidator`
 - `PatternValidator`
 - ...a další...
- Jednoduše lze psát vlastní rozšířením třídy `AbstractValidator`
- Viz. upravený `Příklad4_1.java`

”redirect after post” a historie

- Wicket automaticky používá ”redirect after post”
- Díky serializovaným verzím stránek lze bez problémů procházet historii stránek
- Viz. ukázka na příkladu `Priklad4_1.java`

Příklad 5: Iterace

- Do tagu v HTML, který chceme procházet, přidáme **wicket:id**
- Tag napojíme na komponentu pro iteraci
- Několik možností
 - **RepeatingView** – opakuje zadaná data
 - **ListView** – prochází seznam
 - **PageableListView** – podporuje stránkování
 - **DataView** – pro rozsáhlé kolekce, načte pouze část pro zadanou stránku
- **Viz. Příklad5.java**

Příklad 6: AJAX

- Wicket umožňuje používat AJAX komponenty bez nutnosti jakkoli řešit JavaScript
- Mnoho vytvořených AJAX komponent je již ve standardní distribuci
- Viz. `Příklad6.java`

Příklad 7: Lokalizace

- Klasické properties nebo xml properties soubory
- Pro lokalizaci se použije properties soubor, názvem shodný s názvem nejbližší komponenty
- `Priklad8.java -> Priklad8_cs.properties -> Aplikace_cs.properties`
- Alternativně lze použít pro každou HTML šablonu samostatný soubor:
 - `Find_cs.html`
 - `Find_en.html`
- `<wicket:message key="klic">text</wicket:message>`

Znovupoužitelné komponenty

- Wicket Panel

ColorPicker.java:

```
public class ColorPicker extends Panel{}
```

ColorPicker.html

```
<html>
```

```
<wicket:panel>
```

```
...nějaké HTML s wicket:id, pokud je třeba ...
```

```
</wicket:panel>
```

```
</html>
```

Konzistentní layout

- Normální dědičnost
- Vytvoříme si základní stránku s layoutem aplikace, každá další wicket stránka tuto stránku dědí
- Tagy:
 - `<wicket:extend>`
 - `<wicket:child>`

Urychlení vývoje: sledování změn

- Sledování změn v HTML šablonách
- Sledování změn v class souborech
- Cíl: efektivita a rychlost vývoje, podobná scriptovacím jazykům ;)

Bezpečnost Wicketu

- Verze stránky je provázána se session uživatele a po určité době expiruje
- HTML šablony jsou pod WEB-INF
- Uživatel se ke stránce dostane:
 - Proklikáním, kde startovním bodem je titulní stránka v instanci **WebApplication**
 - Odkazem přes "hezká URL"

Hezká URL

- URL, generovaná Wicketem jsou "session relative":
- `http://localhost:8080/ApacheWicketCzJug/?wicket:interface=:2:::`
- Různé strategie pro "hezká URL"
 - **QueryStringUrlEncodingStrategy**
`/najdi?uzivatel=jirka`
 - **IndexedParamUrlCodingStrategy**
`/uzivatel/jirka`
 - ...a další...

Vývojový vs produkční mód

- Vývojový mód nastaví aplikaci pro jednodušší debugování a rychlejší vývoj (výchozí)
- Produkční mód je výkonnější a bezpečnější
- Přepínání:
 - Systémovou proměnnou
 - `Dwicket.configuration=development`
nebo
 - `Dwicket.configuration=deployment`
 - Případně filter parameter nebo context parameter

Produkční (deployment) mód

[WICKET IN ACTION]

- Ignorování komponent, které jsou ve Wicket třídě, ale nejsou v HTML šablonách
- Z HTML stránek jsou odstraněny veškeré wicket tagy a atributy
- Není vyhazován stacktrace výjimek, pouze stránka s obecnou chybou
- Je vypnutý AJAX debugger
- Z vložených JavaScript souborů jsou odstraněny bílé mezery a komentáře
- Statický textový obsah (CSS a JavaScript) je komprimován gzipem

Wicket a Spring

- Vytvoření aplikačního kontextu v hlavní třídě aplikace
- Anotace **@SpringBean**

Zkušenosti v Kyberii I.

- **Java Server Faces**

- Nové tagy :(
- Vlastní komponenty-neuvěřitelně složité :(
- Hodně XML :(
- Není kontrola nad výsledným HTML :(
- Standard ;)

- **Stripes**

- Jednoduché :)
- Nedotažené :(

Zkušenosti v Kyberii II.

- **Apache Wicket**
 - Odstínění od HTTP :)
 - Stabilní, jednoduchý :)
 - Aktivně používáme 1 rok – bez problémů :)
 - Trvalo to, než jsme ho našli :(

Kam dál?

- **Knihy**

- Wicket in Action
- Pro Wicket

- **Weby**

- <http://wicket.apache.org/>
- <http://wicket.apache.org/community.html>
- <http://wicketstuff.org>
- <http://www.wicket-library.com/>

- **IRC, diskuzní fóra...**

Použité zdroje

- [WICKET IN ACTION] - Wicket in Action, Martijn Dashorst, Eelco Hillenius, PDF verze, Manning Publications Co.: 2009, ISBN 1-932394-98-2
- [WEB] - Apache Wicket [online], dostupné z WWW: <http://wicket.apache.org>, 17.3.2009

Dotazy, diskuze

Otázky(1) a odpovědi (0..*)

Děkuji za pozornost!

jiri.hradil@kyberie.cz

Kyberie s.r.o.

<http://www.kyberie.cz>