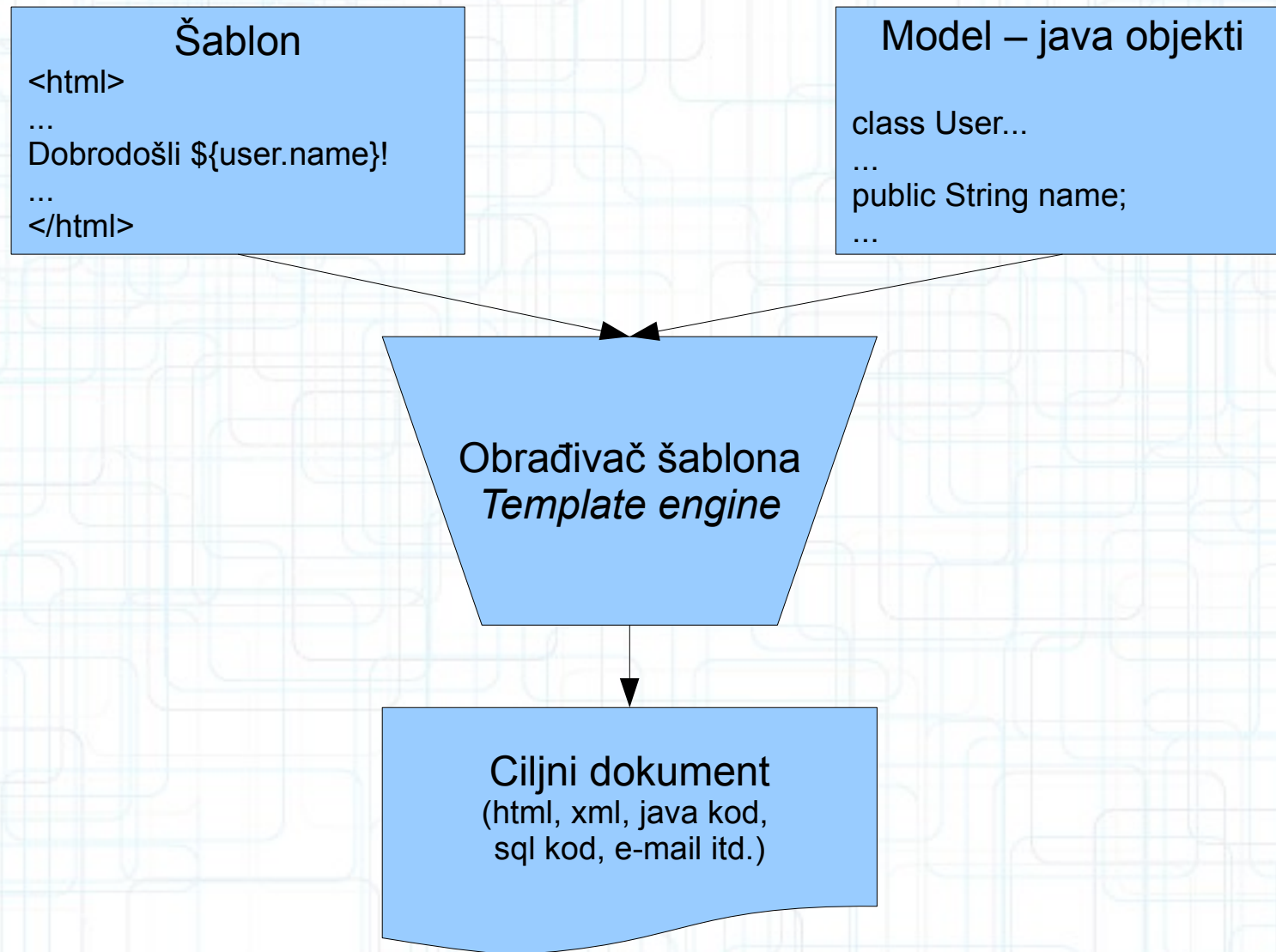


Rukovanje konfiguracijom softvera

Obradivači šablona
(*Template engines*)

Čemu služe?



Predstavnici

- Java – FreeMarker, Velocity, JSP, StringTemplate...
- Php
- Python – Cheetah, Mako, Jinja, Genshi, Django template system...
- ...

FreeMarker

- <http://www.freemarker.org>
- Osobine
 - generisanje bilo kakvog tekstualnog sadržaja (HTML, XML, java source, sql source...)
 - Lightweight – ne zavisi od servlet API-ja. Može se koristiti u bilo kom java projektu (jar biblioteka < 1MB)
 - Zamenjiv TemplateLoader. Šabloni se mogu učitavati sa lokalnog diska, iz jar fajla, iz baze podataka itd.
 - Moćan jezik za šablone.
 - Podrška za JSP taglib-ove (ukoliko su urađeni kao java klase)
 - Podrška za html escaping, lokalizaciju i dr.

Struktura šablona

- Tekst koji se ne obrađuje
- Varijable koje se interpoliraju
 - `${naziv_promenjive_modela}`
 - Prilikom obrade zamenjuje se sa string vrednošću promenjive.
- FTL tagovi ili direktive(*Freemarker Template Language Tags*)
 - `<#ime_taga parameri... > telo </#ime_taga>`
 - Logika: grananje, petlje i sl.
- Komentari
 - `<#-- Komentar -->`

Interpolacija

```
<html>
<head>
  <title>Dobrodošli!</title>
</head>
<body>
  <h1>
    Welcome ${user.name}. I have ${user.messages_count} unread
    messages!
  </h1>
</body>
</html>
```

- Vrednosti `${user.*}` će biti zamenjeni odgovarajućim stringovima prilikom renderovanja šablona.

Izrazi (Expressions) 1/7

- Koriste se pri interpolaciji ili u direktivama.
 - `${ 3+2}`
 - `<#if izraz >...`
- Vrednosti
 - Stringovi: `"Foo"`, `'Foo'`, `"It's \"quoted\""`, `r"C:\raw\string"`
 - Brojevi: `123.45`
 - Logičke vrednosti: `true`, `false`
 - Sekvence: `["foo", "bar", 123.45]`, `1..100`
 - Mape (hash): `{"name":"green mouse", "price":150}`



Izrazi (Expressions) 2/7

- Pristup modelu
 - top-level varijable: `user`
 - vrednost iz mape: `user.name`,
`user["name"]`
 - vrednost iz sekvence: `products[5]`
 - specijalne varijable: `.locale`

Izrazi (Expressions) 3/7

- Operacije sa stringovima
 - Interpolacija: "Hello \${user}!"
 - Konkatanacija: "Free" + "Marker"
 - Izdvajanje karaktera: name[0]
 - ugrađene operacije (built-ins)
 - \${'abc'?substring(0, 1)}
 - \${"green mouse"?cap_first}
 - "redhead"?ends_with("head")
 - \${user?html}
 - "abcabc"?index_of("bc")
 - ...

Izrazi (Expressions) 4/7

- Rad sa sekvencama
 - konkatanacija: `users + ["guest"]`
 - slice: `products[10..19]`, `products[5..]`

Izrazi (Expressions) 5/7

- Mape
 - konkatanacija: passwords + {"joe":"secret42"}
 - pristup: passwords["joe"], passwords.joe

Izrazi (Expressions) 6/7

- Aritmetičke operacije:
 - $(x * 1.5 + 10) / 2 - y \% 100$
- Poređenje:
 - $x == y, x != y, x < y, x > y, x >= y, x <= y, x \< y$, itd.
- Logičke operacije:
 - `!registered && (firstVisit || fromEurope)`
- Poziv metoda:
 - `repeat("What", 3)`

Izrazi (Expressions) 7/7

- Nedostajúće vrednosti:
 - default vrednost: `name!"unknown", (user.name)"unknown", name!, (user.name)!`
- Testiranje da li vrednost postoji:
 - `name??, (user.name)??`

Direktive (FTL tagovi) 1/5

- **if** direktiva

```
<#if logicki_izraz >  
    Tekst koji će biti emitovan ukoliko je izraz istinit.  
<#else>  
    Tekst koji će biti emitovan ukoliko izraz nije istinit.  
</#if>
```

```
<#if animals.python.price < animals.elephant.price >  
    Pythons are cheaper than elephants today.  
<#else>  
    Pythons are not cheaper than elephants today.  
</#if>
```


Direktive (FTL tagovi) 2/5

- **list** direktiva
 - iteracija kroz sekvence
 - specijalne varijable: `list_varijabla_index`, `list_varijabla_has_next`
 - Izlazak iz petlje: `<#break>`

```
<#list sekvenca as list_varijabla>  
  Tekst koji koristi list_varijabla promenjivu.  
</#list>
```

```
<p>We have these animals:  
<table border=1>  
  <tr><th>Name<th>Price  
  <#list animals as being>  
  <tr><td>${being_index}.${being.name}<td>${being.price} Euros  
  </#list>  
</table>
```

Direktive (FTL tagovi) 3/5

- **include** direktiva
 - uključivanje drugog šablona
 - putanja: izraz koji se evaluira u string
 - opcije: encoding, parse

```
<#include putanja_do_sablona opcije>
```

```
<#assign me = "Juila Smith">  
<h1>Some test</h1>  
<p>Yeah.  
<hr>  
<#include "/common/copyright.ftl">
```

Direktive (FTL tagovi) 4/5

- **noparse** direktiva
 - blok teksta koji se neće obrađivati od strane FreeMarker-a

```
<#noparse>  
  <#list animals as being>  
    <tr><td>${being.name}<td>${being.price} Euros  
  </#list>  
</#noparse>
```


Direktive (FTL tagovi) 5/5

- **compress** direktiva
 - brisanje suvišnih *white-space* karaktera.
 - ako se u bloku ws karaktera nalazi *new-line* ceo blok se menja sa *new-line* karakterom. U suprotnom se menja sa *space* karakterom.

```
<#compress>
  1 2 3 4 5
  ${moo}
  test only

  I said, test only

</#compress>
```

Definisanje korisničkih direktiva

- **macro direktiva**
 - definisanje novih direktiva.
 - moguće je definisati proizvoljne parametre

```
<#macro greet person color="black">  
  <font size="+2" color="{color}">Hello ${person}!</font>  
</#macro>
```

```
<@greet person="Fred" color="blue"/>
```

Kreiranje varijabli šablona

- **assign i local** direktive

```
<#assign x = "plain">
1. ${x} <!-- ovde se renderuje "plain" -->
<@test/>
6. ${x} <!-- Vrednost nije promenjena pozivom makroa -->
<#list ["loop"] as x>
    7. ${x} <!-- dolazi do skrivanja loop varijablom. -->
    <#assign x = "plain2"> <!-- zamena vrednosti x variijable -->
    8. ${x} <!-- x je i dalje loop variijabla. -->
</#list>
9. ${x} <!-- nova vrednost x variijable - "plain2". -->

<#macro test>
    2. ${x} <!-- ovde se vidi x variijabla -->
    <#local x = "local">
    3. ${x} <!-- lokalna x variijabla skriva globalnu -->
    <#list ["loop"] as x>
        4. ${x} <!-- loop variijabla skriva lokalnu -->
    </#list>
    5. ${x} <!-- ponovo se vidi lokalna variijabla -->
</#macro>
```


Konfigurisanje FM 2/1

- klasa *freemarker.template.Configuration*
- Koristi se za definisanje parametara obrađivača kao i varijabli koje će biti dostupne svim šablonima.
- Najčešće je jedan objekat konfiguracije na nivou cele aplikacije ali može ih biti i više.
- Svaki šablon ima pridružen samo jedan objekat konfiguracije.
- `Configuration.getTemplate` metoda za kreiranje Template-a.

Konfigurisanje FM 2/2

```
Configuration cfg = new Configuration();  
    cfg.setDirectoryForTemplateLoading(  
        new File("/putanja/do/sablona"));  
    cfg.setObjectWrapper(new DefaultObjectWrapper());
```

Kreiranje modela podataka

```
Map root = new HashMap();  
    root.put("user", "Big Joe");  
    Map latest = new HashMap();  
    root.put("latestProduct", latest);  
    latest.put("url", "products/greenmouse.html");  
    latest.put("name", "green mouse");
```


Učitavanje i procesiranje šablona

```
Template temp = cfg.getTemplate("test.ftl"); // cfg je instanca  
                                              //Configuration klase
```

```
Writer out = new OutputStreamWriter(System.out);  
temp.process(root, out); // root je model podataka  
out.flush();
```

Integracija sa Restlet-om

- `org.restlet.ext.freemarker_2.3.jar`

```
// U Application klasi
this.fmc = new freemarker.template.Configuration();
this.fmc.setTemplateLoader(new ClassTemplateLoader(getClass(),
    "templates"));

// U Resource klasama
protected Representation getTemplateRepresentation(String templateName,
    Map<String, Object> dataModel, MediaType mt) {

    return new TemplateRepresentation(
        templateName, getFmcConfiguration(), dataModel, mt);
}

public Representation represent(Variant variant)
    throws ResourceException
{
    final Map<String, Object> dataModel = new TreeMap<String, Object>();
    dataModel.put("user", this.user);
    return getTemplateRepresentation("user.html",
        dataModel, variant.getMediaType());
}
```