

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Dokumentace k Projektu 5

Doplňky redakčního systému OpenCMS

Obsah

1 Zadání práce

- 1.1 Původní zadání
- 1.2 Zadání upřesněné na schůzkách se supervisorem
- 1.3 Shrnutí zadání

2 Popis Řešení

- 2.1 Stručný popis řešení
- 2.2 Průvodce po redakčním systému OpenCMS
- 2.3 Použité JSP knihovny značek a jejich funkce

3 Závěr

- 3.1 Shrnutí řešení
- 3.2 Možná rozšíření k bakalářské práci

4 Zdroje

- 4.1 Zdroje dostupné na internetu
- 4.2 Literatura

1 Zadání práce

1.1 Původní zadání

Doplňky redakčního systému OpenCMS

Upravit šablony, přidat drobné funkčnosti do OpenCMS nasazeného na KIV. Předpokládá se například:

- * často navštěvované
- * naposledy navštívené
- * aktuality

1.2 Zadání upřesněné na schůzkách se supervisorem

Funkčnosti nebudou prozatím přidány do *OpenCMS* nasazeného na KIV, stačí je zakomponovat do jedné ze základních šablon OpenCMS – *Template One* s použitím *Xml Content* – tzn. aktuality nemusí být ukládány do databáze, ale formát každé aktuality bude strukturovaným obsahem uloženým v *xml* souboru a validovaný pomocí validačního souboru *xsd*.

Aktualita bude mít následující parametry:

- Nadpis
- Obsah (hypertext)
- Datum a čas zveřejnění
- Oblast webu do které patří
- Kdo zveřejnil aktualitu

Aktuality budou řazeny reverzně chronologicky tzn. čerstvější aktuality budou nahoře.

1.3 Shrnutí zadání

OpenCMS je redakční systém založený na značkovacím jazyce XML a jazyce Java a JSP(Java server pages), které jsou součástí platformy Java Enterprise Edition. Projekt tedy bude vyvíjen za pomoci těchto technologií.

Jelikož jsem se s OpenCMS nikdy předtím nesetkal, bylo součástí zadání i nastudování tohoto redakčního systému. Chtěl bych toto zdůraznit, protože úplným nováčkům bude určitě chvíli trvat, než proniknou do jeho nitra. Součástí tohoto projektu tedy bude i průvodce po redakčním systému OpenCMS i proto, že jeho pochopení je možná důležitější, než rozsáhlá znalost JSP. Bude to jen pár rad, které je potřeba znát k vytvoření alespoň základního projektu, jako je tento.

2 Popis řešení

2.1 Stručný popis řešení

Jak již bylo řečeno v zadání, aktuality se nebudou muset ukládat do databáze, ale každá aktualita bude uložena v jednom xml souboru, které se pak podle potřeby budou vypisovat na stránce.

Jelikož byly zadány parametry aktualit, musel být nejprve vytvořen editor pro vlastní strukturovaný obsah – tento strukturovaný obsah má být xml soubor, jehož typ jsem nazval *kiv_aktualita*. Instalace OpenCMS sice obsahuje defaultní editor pro aktuality(news), ale ty mají samozřejmě jiné parametry, než které jsou potřeba pro tento projekt. Abych mohl vytvořit vlastní editor pro strukturovaný obsah, musel jsem nejprve vytvořit vlastní OpenCMS modul, který jsem nazval *com.projekt5aktuality.site*.

V adresáři modulu jsem vytvořil validační soubor *kiv_aktualita.xsd*, který definuje, co má editor strukturovaného obsahu definovat. V tomto validačním souboru jsou definovány parametry aktualit ze zadání a pro některé parametry jsou zde nastaveny i inicializační podmínky. Dále jsem v adresáři modulu vytvořil dvě jsp stránky:

- *detail.jsp*
- *side_list_aktuality.jsp*

První z nich je má vlastní šablona pro xml strukturovaný obsah, která zobrazí všechny údaje aktuality na jedné stránce.

Druhý je šablonou, která zobrazí 5 nejčerstvějších aktualit na pravé straně stránky.

Samotný modul, tedy definuje vlastně jen šablony pro zobrazování nějakého obsahu. Tento samotný obsah se potom je potom uložen v adresáři OpenCMS virtuálního souborového systému(dále jen VFS) */sites/default/projekt5_aktuality/*. Do tohoto adresáře se ukládá veškerý potřebný xml obsah – jsou zde umístěny všechny vytvořené aktuality, jejichž soubory mají název *kiv_aktualita [číslo]*. Dále jsou zde umístěny dvě hlavní jsp stránky projektu:

- *index.jsp* – Úvodní stránka projektu
- *aktuality.jsp*

První z nich obsahuje pouze pár staticky vložených informací o projektu. Přes druhou stránku můžeme vytvářet, editovat i mazat nové aktuality, jejichž seznam tato stránka vypisuje. Obě stránky pak obsahují na pravé straně 5 nejčerstvějších aktualit, k čemuž používají výše zmíněnou šablonu *side_list_aktuality*.

2.2 Průvodce po redakčním systému OpenCMS

OpenCMS je opensource content management system(= systém pro správu obsahu, dokumentů nejčastěji webového obsahu) založený na Javě a XML. Pro svou funkčnost potřebuje webový kontejner – např Apache Tomcat a databázový systém – např. MySQL. Před samotnou instalací musí být nainstalovány následující komponenty:

- Java SDK 1.4, nebo novější
- webový kontejner – Tomcat
- MySQL

Instalace se provádí přes aplet na webové stránce, kterou spustíme na localhostu na Tomcatu. Tady zmíním hned první problém, se kterým jsem se setkal a trvalo mi dost dlouho, než jsem ho vyřešil. Při instalaci OpenCMS na webový kontejner Apache Tomcat se mi proces

instalace zastavil na kroku 8 – importování modulů do workplace na Tomcat. Byla to neočekávaná chyba na kterou jsem nenašel řešení ani v logu instalace, kde sice byly nějaké chyby zmíněny, ale ne už jejich řešení. Hledal jsem tedy na [ref_1], [ref_2] a [ref_3] co by mohlo chybu způsobit. Zvláště v mailing-listu a na fóru jsem našel mnoho příspěvků od uživatelů, kteří měli podobný problém a taky mnoho odpovědí na tyto příspěvky:

- vytvořit účet pro dva uživatele při instalaci systému – jednoho s právy Administrátora, druhého s právy běžného uživatele a přes toho spustit instalaci OpenCMS.
- nastavit správně práva uživatelů Tomcatu
- zkontrolovat, zda je správně spuštěn datábázový systém na počítači
- zapnout Virtual Machine na Tomcatu.
- nastavit proměnou v konfiguračním souboru datábázového systému MySQL
`max_allowed_packet=8M`

Všechny tyto možnosti jsem vyzkoušel, nemělo to však úspěšný výsledek. Tento problém jsem po delší době vyřešil až přeinstalováním webového kontejneru Apache Tomcat a datábázového systému MySQL. Při první instalaci OpenCMS, jsem už totiž měl obě zmíněné komponenty již přenastavované z dřívější doby a OpenCMS si zřejmě s těmito nastaveními moc nerozuměl. Na defaultní instalaci Tomcatu a MySQL se již OpenCMS nainstaloval bez problémů.

Po prvním seznámení s OpenCMS si uživatel musí zvykat na vlastní souborový systém OpenCMS VFS, který se značně odlišuje od typického souborového systému např. Microsoft Windows(Real file system dále jen RFS). Tyto souborové systémy jsou spolu propojené pouze přes moduly zabalené do zip souborů. Samotné jsp soubory vytvořené v OpenCMS v RFS nenajdem. K tomu abychom mohli jsp soubory editovat zvenku oblíbeným editorem jsp souborů, musíme nejdřív vytvořit modul s jeho základní strukturou(viz [rfc_4]), exportovat ho do RFS, rozbalit zip soubor V RFC, editovat jsp soubory(popř. soubory jiných typů př. XML, Java, textové soubory) a importovat zpět do VFS.

Samotný VFS je však mocným nástrojem OpenCMS. Každý adresář i soubor je definován mimo svého jména ještě typem souboru (př. xmlcontent, jsp, můžem vytvářet i vlastní typy) a mnoha parametry, které pomáhají k samotnému vytváření jsp stránek.

Ještě zmíním něco málo o Template One šabloně, kterou tento projekt používá. Jedná se o základní vzorovou šablonu, která je obsažena v defaultní instalaci OpenCMS. Má demonstrovat možnosti OpenCMS co se týče šablon a slouží i jako příklad pro vytváření vlastních šablon. Pomocí této šablony můžeme pohodlně vytvořit pěkně vypadající stránku obsahující různá menu, css layout a spoustu dalších dodatečných funkcí, které nemusíme sami programovat, protože jsou již vytvořeny v modulu *templateone*.

Nyní se již po krátkém seznámení s OpenCMS, užitečným pro začínající tvůrce vrhnu na popis kroků, které jsem učinil při vytváření projektu.

Nejprve bylo třeba vytvořit vlastní modul(návod viz [ref_4]), abychom mohli vytvořit svůj vlastní editor strukturovaného obsahu. Po vytvoření modulu přes Administrátorské rozhraní musíme ještě pár parametrů přímo napsat do xml souboru s nastavením modulu. Tento soubor najdeme v RFS *TOMCAT_DIR/webapps/opencms/WEB-INF/config/opencms-modules.xml*. Nejprve přidám položku *exportpoints*, která udává, které adresáře a soubory budou při exportování modulu exportovány do RFS. Řádka vypadá takto:

```
<exportpoints><exportpoint
uri="/system/modules/com.projekt5aktuality.site/classes/"
destination="WEB-INF/classes/" />
</exportpoints>
```

V tomto adresáři je umístěn soubor *workplace.properties*. Tento adresář leží uvnitř servlet kontextu, tzn. že *workplace.properties* do něj musíme umístit tím způsobem jako balík javy. To je v tomto případě `/com/projekt5aktuality/site/`. Tuto strukturu bude mít adresář *classes*. Dále přidáme položku *resourcetypes*:

```
<resourcetypes>
  <type
    class=
      "org.opencms.file.types.CmsResourceTypeXmlContent "
    name="kiv_aktualita" id="301">
    <param
name="schema">/system/modules/com.projekt5aktuality.site/schemas/kiv_aktual
ita.xsd</param>
  </type>
</resourcetypes>
```

V tomto xsd bude definováno schéma pro náš vlastní strukturovaný obsah. Číslo id musí být mezi všemi moduly, které přidávají resource type, unikátní.

Další položka je *explorertypes*. Ta přidává parametry jako ikonu, jméno, uri na které budou soubory tohoto typu mapovány.

Nakonec přidáme zdroje, které se při exportování budou exportovat s tímto modulem:

```
<resources>
  <resource uri="/projekt5_aktuality/" />
  <resource
    uri="/system/modules/com.projekt5aktuality.site/" />
</resources>
```

První zdroj je adresář v `sites/default/` kde je umístěna hlavní stránka projektu `index.jsp`. Druhý zdroj je samotný adresář vytvořeného modulu.

Po editaci souboru *opencms-modules.xml* nesmíme zapomenout na restartování webového kontejneru, jinak naše změny nebudou v OpenCMS zaznamenány.

Nyní se vrhneme na vytvoření xsd souboru se schématem a potřebných doplňujících šablon pro zobrazení aktualit.

Nejprve vytvoříme základní adresářovou strukturu v adresáři modulu:

1. `/classes`
2. `/elements`
3. `/jsptemplates`
4. `/schemas`

ad 1. bude obsahovat *workplace.properties* soubor

ad 2. bude obsahovat jsp, či html stránky, které budeme vkládat akcí *include* později do našich stránek v `/sites/default/`. Pro naši potřebu bude obsahovat pouze soubor *side_list_aktuality.jsp*, který zobrazí 5 nejčerstvějších aktualit v pravém menu.

ad 3. obsahuje dodatečné šablony. V našem případě pouze *detail.jsp*

V ad 4. bude umístěn náš xsd soubor *kiv_aktualita.xsd*, který popisuje co má editor strukturovaného obsahu obsahovat. V našem případě bude obsahovat parametry aktualit. Návod, jak by mělo takové xsd zhruba vypadat je v [ref_4].

První co musíme vytvořit je ale soubor *workplace.properties*, který musí být umístěn do adresáře *classes* ještě do podadresáře se strukturou, kterou jsem již výše popisoval. Aby se naše ikona strukturovaného obsahu správně zobrazila ve formuláři, kde budem později vytvářet nový soubor typu *kiv_aktualita*, musí *workplace* obsahovat minimálně tyto 2 řádky:

```
fileicon.kiv_aktualita = Aktualita redakcniho systemu kiv
title.kiv_aktualita = Vytvor novou aktualitu redakcniho
                    systemu kiv
```

Poté vytvoříme xsd soubor. Musí se dávat pozor, aby cesta k souboru i jeho jáho souhlasilo s definicí, kterou jsme vkládali do *opencms-modules.xml*.

Teď abychom mohli vytvářet jsp stránky musíme nejprve celý adresář s modulem publikovat (Publish directly) a poté opět restartovat Tomcat. Tím docílíme, že soubor workplace se zkopíruje na správné místo v adresářové struktuře Tomcatu.

Nyní již můžeme vytvořit první aktualitu typu kiv aktualita. Ve vytvořeném hlavním adresáři našeho projektu *projekt5_aktuality* vytvoříme nový soubor tak, že stačí vybrat typ souboru Structured content kde se již objeví námi nadefinovaný typ souboru kiv_aktualita. Po použití funkce *DirectEdit*, pomocí které budeme editovat soubory xmlcontent, již můžeme zadat všechny parametry aktuality, které jsme nadefinovali v xsd souboru. Aktualitu však ještě nemůžeme zobrazit v prohlížeči, protože ještě nemáme vytvořené šablony pro její zobrazení. Proto vytvoříme šablonu v adresáři našeho modulu v */jsptemplates*. Jak vytvořit takovou základní šablonu najdete v [ref_4].

K tomu aby všechny aktuality, které vytvoříme se zobrazovali podle této šablony, musíme tuto šablonu namapovat do property *template-elements* složky v */sites/default*:

```
/system/modules/com.projekt5aktuality.site/jsptemplates/default/
tail.jsp
```

Nyní když zobrazíme náš xml soubor s aktualitou v prohlížeči, uvidíme ji přesně podle šablony detail.jsp. Nyní již můžeme vytvořit všechny ostatní jsp stránky, ve kterých můžeme zobrazovat aktuality různými způsoby.

2.3 Použité JSP knihovny značek a jejich funkce

Ve vytvořených jsp souborech používám následující knihovny značek:

OpenCMS JSP Taglib

```
<%@ taglib prefix="cms" uri="http://www.opencms.org/taglib/cms" %>
```

- abychom ji mohli v jsp souboru použít, musíme do něj vložit výše napsanou direktivu
- dokumentace viz [ref_4]

Knihovny z Java Standart Tag Library (JSTL)

Core – obsahuje akce se základními funkcemi využívanými ve většině JSP aplikací

```
<%@ taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
```

Fmt – formátovací tagy, pro podporu lokálního prostředí

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jstl/fmt" %>
```

Použité funkce:

Vytvoření instance OpenCMS scriptletu:

```
<%
    CmsTemplateModules cms = new CmsTemplateModules(pageContext,
    request, response);
```

```
%>
```

Po té můžem vyžít OpenCMS scriptlet funkce

Načtení XML obsahu

```
<cms:contentload collector="{property.collector}"
    param="/projekt5_aktuality/kiv_aktualita_{number}.html|kiv_aktualita"
    editable="true">
```

Tato funkce je použita v souboru *aktualit.jsp* a slouží k tomu, že si najde xml obsah ve VFS podle svého parametru *collector*. V tomto souboru se nastavuje *collector* v jeho *property* a hodnotou je: *allInFolderDateReleasedDesc*, což zaručí že všechny soubory typu *kiv_aktualita*, které se jmenují stejně jako *para(výše)* se zobrazí na stránce a budou seřazeny podle data zveřejnění. To zaručuje namapování parametru aktuality *Cas_zverejneni* do *property* šablony *Template One – datereleased*. Parametr *editable* zaručí, že budeme moci aktuality editovat pomocí funkce *DirectEdit*. V souborech *aktuality.jsp*, *side_list.jsp* je *collector* nastavený takto na reverzní chronologické řazení. V souboru *detail.jsp* je nastaven na *singleFile*, což zaručí zobrazení jedné aktuality na samostatnou stránku.

Zobrazení XML obsahu na stránku:

```
<c:set var="dateString">
    <cms:contentshow element="Cas_udalosti" />
</c:set>
<%
    cms.setDate("dateString");
%>
<fmt:formatDate value="{date}" type="both" dateStyle="short" />
```

c:set nastaví proměnnou *dateString*, do které se pomocí elementu *contentshow* (tetno element zobrazí xml obsah na stránce) načte parametr aktuality *Cas_udalosti*. Pomocí skriptletu *cms.setDate* se proměnná stane typem *Data* a *fmt:formatDate* ji správně sformátuje.

Zobrazení layoutu Template One na každé stránce

Abychom naše stránka použila hlavičku a patičku alá *Template One*, musíme do ni přidat:

```
<cms:include property="template" element="head" /> popř. foot
```

předtím však bychom řekli našemu projektu, že má používat šablonu *Template One*, musíme nastavit v adresáři stránek *property template* na:

```
/system/modules/org.opencms.frontend.templateone/templates/main
```

Nyní můžeme zkusit přidat *side_list_aktuality* na pravou stranu našich stránek, musíme do každé stránky, která ji bude používat přidat do *property*

```
xmlcontent-side-uri :
    kiv_aktuality
```

Na pravé straně se teď zobrazí sada článků *articles* z *demostránek xmlcontentdemo*. To je způsobeno tím, že šablona *Template One* je nastavena, tak že když *property xmlcontent-side-udri* obsahuje nějaký řetězec, tak šablona nezjišťuje jeho hodnotu a ihned na pravou stranu dosadí ony články z *xmlcontentdemo*. Aby se zobrazily naše aktuality, musíme mírně upravit šablonu v *Template One* následovně:

V adresáři modulu *Template One*:

```
/system/modules/org.opencms.frontend.templateone/elements/
```


je soubor *info_side.jsp*, což je šablona, která se stará o zobrazení postraních elementů. Obsahuje řádek:

```
String xmlDemo = cms.property("xmlcontent-side-uri", "search");
boolean showXmlDemo = CmsStringUtil.isEmpty(xmlDemo) &&
    !"none".equals(xmlDemo);
```

My ji doplníme o:

```
//pridano pro kiv_aktuality
boolean showAktuality = CmsStringUtil.isEmpty(xmlDemo) &&
    xmlDemo.equals("kiv_aktuality");
```

A rozšíříme podmínku zobrazení:

```
if (showAktuality) {

    cms.includeSilent("/system/modules/com.projekt5aktuality.site/elements/side_list_aktuality.jsp", null, true);
    if (showSideUri) {
        out.println(spacer);
    }
}
else if (showXmlDemo) {

    cms.includeSilent("/system/modules/org.opencms.frontend.templateone.xmlcontentdemo/elements/side_article_list.jsp", null, true);
    if (showSideUri) {
        out.println(spacer);
    }
}
```

Což zaručí, že pokud řetězec v *xmlcontent-side-uri* bude „kiv_aktuality“, zobrazí se naše aktuality, v ostatních případech *xmlcontent demo* články.

Podrobné informace o použitých knihovnách lze nalézt v [ref_4] a [ref_5].

3 Závěr

3.1 Shrnutí řešení

Vytvořené soubory modulu jsou:

```
com.projekt5aktuality.site_0.4.zip
```

a modifikovaný modul Template One:

```
org.opencms.frontend.templateone_2.0.1.zip
```

Je třeba oba tyto soubory importovat k správné funkci projektu.

Projekt samozřejmě splňuje jen základní funkčnost zadání a má být spíše demonstrací toho, co má být později vytvořeno jako modul aktualit do redakčního systému na kivu. Naučil jsem se na něm ale pracovat s moduly, s xml contentem, vytvořit si vlastní jednoduchý template, připojit vlastní modul do již vytvořeného templatu, a zjistil základní rozdíly implementace JSP aplikací samotných a JSP aplikací v rámci OpenCMS. Rozdíl je na první pohled velký, protože openCMS používá pro většinu implementace vlastní definované akce a skriptlety, které se v JSP Standardních knihovnách používají jiným způsobem přes prefix="jsp". Až na prefix mají OpenCMS základní akce i stejné názvy jako standardní. Standardní základní akce by mohli být použity v OpenCMS také, avšak není to doporučeno z důvodu, že není zaručena, jejich správná funkčnost pod OpenCMS.

Tyto základní znalosti se budou při pozdějším vývoji použitelných aktualit do redakčního systému velice hodit.

3.2 Možná rozšíření k bakalářské práci

Rozšíření jsou patrná z původního zadání. Vylepšena bude muset být editace aktualit, aby nějaké elementy byly povinné, nějaké needitovatelné, pouze pro zobrazení. Parsovat aktuality podle hodnot jejich parametrů, aby se např. jednotlivé aktuality zobrazovaly pouze v předdefinované oblasti webu, aby mohl aktuality editovat pouze, člověk, který je zadal atd. Vytvořený modul by neměl být už ani závislý na šabloně Template One, ale měl by být vytvořen jako plnohodnotná šablona, která se o svůj obsah stará sama, maximálně používá šablonu redakčního systému kivu, která bude muset být nastudována. Tento modul by měl v tomto případě jít importovat právě do modulu kivu.

Samozřejmě vhodnější pro větší projekty bude, když jednotlivé aktuality budou uloženy v databázi. Xml Content je pro větší projekty poměrně nevhodný.

Samozřejmě se najdou i další možná vylepšení, protože tento projekt byl opravdu pouze zkušební, a takto vytvořené aktuality by se z hlediska zbytečnosti parametrů jako je např. zmíněná oblast webu, v praxi neuplatnily.

4 Zdroje

4.1 Zdroje dostupné na internetu

- [ref_1] <http://www.opencms.org/opencms/en/> - oficiální stránka redakčního systému OpenCMS.
- [ref_2] <http://mail.opencms.org/pipermail/opencms-dev/> - archivovaný mailing-list OpenCMS, kde je možné najít řešení různých problémů týkajících se vývoje stránek nasazených na OpenCMS
- [ref_3] <http://www.opencms-forum.de/opencms-forum/index> - Fórum o OpenCMS
- [ref_4] Dokumentace OpenCMS. Lze ji stáhnout na:
<http://www.opencms.org/opencms/en/download/documentation.html>.
Dokumentace je většinou přítomna i v základní instalaci OpenCMS jako *alkacon-documentation*.
- [ref_5] <http://nb.vse.cz/~zelenyj/it380/eseje/xkrap18/JSTL.htm> - Popis Java Standart Tag Library

4.2 Literatura

- [lit_1] Gary Bollinger, Bharathi Natarajan: JSP Java Server Pages – podrobný průvodce začínajícího tvůrce JSP
- [lit_2] Marty Hall: Java servlety a stránky JSP – Kniha popisující technologie vytváření webových stránek s mnoha praktickými příklady