

Semestrální Práce – Liferay

Vývoj webových aplikací 2

12. Května 2014

Stanislav Sazonov

Obsah

Obsah	2
Zadání semestrální práce	3
Motivace a úvod	4
Základní požadavky pro nasazení webové aplikace využívající Liferay	5
Vývojové prostředí	9
Struktura projektu	11
Osobní zkušenosti a závěr	13

Zadání semestrální práce

Jako téma svého semestrálního projektu jsem si vybral popis portálového řešení Liferay, jeho výhody/nevýhody, nasazování a osobní zkušenosti. Tuto práci jsem pojal jako teoretický přehled Liferay, ne jako manuál pro instalaci a použití. Přidal jsem i několik osobních zkušeností, názorů a příkladů z komerční praxe, které by mohly čtenáře také zajímat.

Motivace a úvod

V dnešní době téměř každý potřebuje mít vlastní web. Pro vytvoření jednoduchého webu, typu prezentace nebo osobní stránky bohatě postačí základní znalost HTML, JavaScript a například PHP.

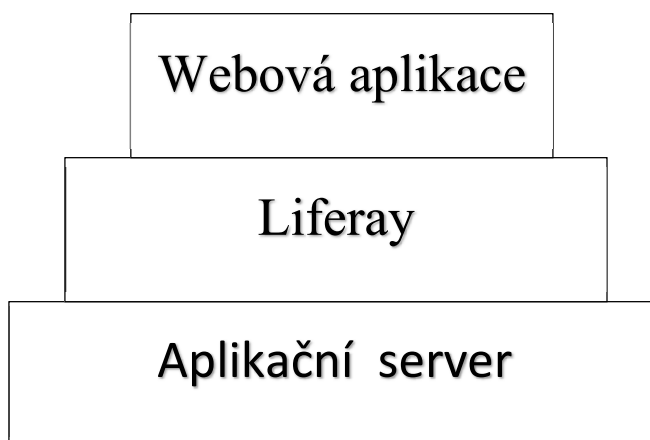
Korporátní weby a velké informační systémy však vyžadují použití mnohem sofistikovanějších technologie. Bez pochyb, nejrozšířenější je .NET nebo J2EE. I já jsem se před několika lety rozhodoval, zda se zaměřit na studium .NET či J2EE, nakonec jsem zvolil J2EE – spíše z osobní sympatie k Javě.

Pro Javu EE je napsána velká řada frameworků, které podstatně zjednodušují a urychlují vývoj webové aplikace, jako je například Spring, Hibernate a jiné. Avšak i přes to se developeri setkávají s tím, že musí ve svých projektech dokola řešit ty samé věci, jako jsou například: forum, agregátor obsahu webu, zabezpečení a jiné.

Jednou z možností, jak zmíněné problémy efektivně řešit je použití Liferay portálu, se kterým bych chtěl čtenáře v této práci seznámit.

Co je to Liferay?

Liferay je portál, postavený na J2EE. Portál je vlastně jaké si prostředí, které běží na aplikačním serveru a které spravuje a řídí webovou aplikaci, tuto závislost znázorňuje následující obrázek:



Ovšem Liferay sám o sobě není framework, ale portál, tedy jakési prostředí, které mnohé tyto feachury již v sobě obsahuje, na programátorovi pak akorát zbývá tyto komponenty vhodně nakonfigurovat a použít. Nikomu se nesnažím vnucovat používání Liferay, ale jsem přesvědčen, že každý J2EE developer by o této možnosti měl minimálně vědět. Za zkoušku to určitě stojí.

Základní požadavky pro nasazení webové aplikace využívající Liferay

K tomu, abychom publikovali webovou aplikaci využívající Liferay musíme splnit několik základních požadavků:

1. Webový server s operačním systémem (já osobně preferuju OS Linux) - v dnešní době se nejčastěji používá cloudové řešení.
2. Na server je potřeba nainstalovat Javu, tedy např. JDK 1.7. a vyšší
3. Dále je potřeba nainstalovat aplikační server. Osobně preferuju TomCat – je malý, spolehlivý, dobře odladěný, poskytuje široké nastavení a možná i nejrozšířenější
4. Dále budeme potřebovat Liferay, který nahrajeme do adresáře TomCat
5. Instalace databázového serveru

Nyní si o těchto čtyřech bodech povíme více

Výběr hostového serveru

Většina nově vznikajících aplikací se umísťuje do cloudu, je to moderní trend, který má mnoho výhod, například:

- Nemusíme se starat o fyzickou správu serveru
- Škálovatelnost – budeme-li potřebovat navýšit výkon, RAM atd., zařídíme to několika kliknutími myši
- Zjednodušená vzdálená podpora
- Lepší zabezpečení

Proto bych doporučil využít právě cloudové řešení s OS Linux. Po založení účtu na některém z cloudových serverů dostaneme přístupová data

Připojení se na server

Pro připojení na server se dá použít některý z SSH klients, já osobně preferuju PuTTY, který se dá stáhnout zde:

<http://www.putty.org/>

Po zadání potřebných přístupových údajů se vytvoří roura mezi serverem a PC a spustí se linuxová konzole na serveru. Tímhle způsobem se dá server pohodlně konfigurovat, vyžaduje to však hlubší znalost Linuxu.

Jednotlivé soubory se dají nahrávat pomocí utility WinSCP, která se dá stáhnout zde:

<http://winscp.net/>

WinSCP má interface prohlížeče se dvěma okny, kdy okno nalevo zobrazuje adresářovou strukturu na PC a okno napravo zobrazuje adresářovou strukturu na serveru. Pomocí této utility se dají soubory jednoduše kopírovat a nastavovat přístupová práva.

Instalace aplikačního serveru a Liferaye

Liferay je možné používat téměř s jakýmkoliv z běžných aplikačním serverů, podporuje Tomcat, GlassFish, JBoss, WebLogic a některé další. Ovšem podle názorů mých kolegů v práci je lepší používat některý z bundlů, tedy balíček, který v sobě obsahuje: aplikační server + Liferay + JRE. Takovou to sestavu (bundle) můžeme stáhnout na stránkách Liferaye:

<http://www.liferay.com/>

Používání bundlu má jednu výhodu – taková sestava je celkem solidně otestovaná a je mnohem větší jistota, že celý systém bude stabilnější a odolnější vůči chybám. Asi nejrozšířenější bundle je Tomcat + Liferay + JRE, proto bych ho také zvolil. Stažený bundle nainstalujeme tak, že ho jednoduše rozbalíme do adresáře. Jak už bylo řečeno, bundle se skládá z těchto tří částí:

- Aplikační server
- Liferay
- JRE – kompatibilní verze Javy v zip archivu

Rozbalený bundle může mít následující adresářovou strukturu (záleží na verzi budlu):

```
liferay-portal-6.1.1-ce-ga2
|- data
|- deploy
|- license
|- logs
|- tomcat-7.0.27
    |- bin
    |- conf
    |- jre1.6.0_20
    |- lib
    |- logs
    |- temp
    |- webapps
        |- ROOT
        |- welcome-theme
```

Na první pohled je patrné, že projekty se „deployují“ do složky `./liferay-portal-6.1.1-ce-ga2/deploy`, která je ve stejné úrovni jako adresář **tomcat-7.0.27**, je to kvůli tomu, že Liferay “přebírá” kontrolu nad projektem. Další zajímavostí je složka **jre1.6.0_20** v Tomcatím adresáři, je to kvůli tomu, že se Tomcat spouští na Javě dodávané v bundle aby se zajistila maximální kompatibilita verzí.

Nic samozřejmě nebrání použít nějaký jiný aplikační server, vývojář ale pak musí počítat s tím, že se mohou objevovat zcela nečekané chyby a musí se obrnit trpělivostí při řešení některých problémů.

Já osobně, po zkušenosti s celkem „choulostivým“ Liferayem, bych v žádném případě od bundlu neustupoval, i tak jsem zažil perné chvíle a určitě bych si nechtěl ještě více komplikovat život.

Pokud by se ale přeci jen někdo rozhodnul nepoužít bundle, pak bude potřeba, aby si zmíněné 3 komponenty (Aplikační server, Liferay, JRE) stáhnul sám, poskládal do adresářů a nakonfigurovala cesty. Učitě je dobré při tom ověřit kompatibilitu všech těchto částí.

Instalace databázového serveru

Liferay defaultně používá svou vnitřní databázi, která se dodává přímo s Liferazey. Je to vlastně lightweight databáze, která slouží spíše pro prezentaci, než pro nějaké komerční nasazení. Liferay dokáže pracovat pracovat prakticky s jakou-koliv databází, pro kterou je napsán odpovídající „javovský“ driver.

Já osobně rád používám Oracle Database Express Edition – je to výkonný databázový stroj, poskytuje velké množství nastavení, optimalizací atp., dá ses stáhnout zde:

<http://www.oracle.com/>

Po nainstalování databáze ji můžeme konfigurovat přes konzoli, což je výhodné v případě vzdáleného přístupu, a nebo můžeme využít utilitu Oracle SQL Developer, která se dá též stáhnout ze stránek Oraclu. Abychom “přinutili” Liferay používat jinou, než interní databázi, stačí upravit konfigurační properties file `portal-setup-wizard.properties`, umístěný v rootu složky s Liferayem. Přidáme do něj několik proměnných určujících přístup k DB. Příklad konfigurace pro Oracle DB:

```
jdbc.default.password=1234
jdbc.default.driverClassName=oracle.jdbc.driver.OracleDriver
jdbc.default.username=MY_LIFERAY
jdbc.default.url=jdbc:oracle:thin:@localhost:1521/xe
```

Dále je důležité nezapomenout přidat nezbytný driver do složky `liferay-portal-6.1.1-ce-ga2_oracle\tomcat-7.0.27\lib\ext`, v případě Oracle databáze jsou to 2 knihovny:

- `c3p0-0.9.2.1.jar`
- `c3p0-oracle-thin-extras-0.9.2.1.jar`

Po nakonfigurování a spuštění databáze, můžeme ze složky `liferay-portal-6.1.1-ce-ga2\tomcat-7.0.27\bin` spustit i aplikační server příkazem `catalina -run`

Jakmile se načte Liferay, zjistí, že přístupový bod k databázi byl změněn, vytvoří v nové DB několik desítek tabulek nezbytných pro svou funkčnost a do nich překopíruje data z databáze, kterou využíval před tím. Pokud by přístupový bod k databázi byl definován chybně, nebo databázový stroj byl vypnut, tak Liferay vyhodí exception a ani se nespustí.

Vývojové prostředí

Já osobně pro vývoj javovských aplikací nejradši používám NetBeans IDE. Měl jsem možnost 9 měsíců pracovat v Oracle v Praze na Chodově a musím říct, byl jsem moc překvapen, s jakou pečlivostí a jakým nasazením pracuje QA team, díky tomu se daří včas odhalovat chyby (zejména P1-P2) a ty fixovat ještě před releasem nové verze. A proto mám NetBeansy mám nejradši.

Bohužel jsem ale zjistil, že podpora Liferaye pro Eclipsy je mnohem silnější, než pro NetBeans. To se hlavně projevuje při debuggingu. Jelikož mám zkušenosti s Liferayem a vím, jak je to „choulostivý“ portál, doporučoval bych spíš použít Eclipsy. Já osobně pro vývoj Liferay aplikací používám Spring Tool Suite (STS), je to vlastně mutace Eclipse přizpůsobená pro Spring, dá se stáhnout zde:

<http://spring.io/tools/sts>

Po nainstalování a spuštění STS bude dobré doinstalovat Liferay plugin, který se instaluje přímo z vývojového prostředí a integruje se přímo do vývojového prostředí STS.

Jeden z nejmocnějších nástrojů při vývoji webových aplikací je debugger. Po nakonfigurování Liferay pluginu, tento plugin bude schopen sám spouštět server, deploydovat aplikace a pokud spustíme server v režimu debugging, tak budeme schopni využít jeden z nejmocnějších nástrojů ladění. Konkrétně STS podporuje tyto debugovací příkazy:



Key	Description
F5	Executes the currently selected line and goes to the next line in your program. If the selected line is a method call the debugger steps into the associated code.
F6	F6 steps over the call, i.e. it executes a method without stepping into it in the debugger.
F7	F7 steps out to the caller of the currently executed method. This finishes the execution of the current method and returns to the caller of this method.
F8	F8 tells the Eclipse debugger to resume the execution of the program code until it reaches the next breakpoint or watchpoint.

(převzato z: <http://www.vogella.com/tutorials/EclipseDebugging/article.html>)

Zároveň, kdykoliv breakneme aplikaci, můžeme nahlédnout do hodnot proměnných, objektů, parametrů, fieldů a podobných datových struktur. Díky tomu je Liferay plugin na tolik populární a rozšířený. Nedokážu si představit vývoj webové aplikace bez efektivního debugovacího nástroje, jakým Liferay plugin bezesporu je.

Po nainstalování pluginu již můžeme vytvořit Hello World aplikaci, stačí v menu STS zvolit File > New > Liferay Project, zadat název, umístění projektu a potvrdit.

Struktura projektu

Nově vytvořený projekt se obvykle skládá ze 4 základních modulů:

Portlet

V tomto modulu budou obsaženy všechny portlety, které v projektu vytvoříme a tedy i jejich podpůrné services, domény atp., pokud se například budeme držet MVC patternu.

Portlet je vlastně obdélníková grafická komponenta, která vykonává specifickou naprogramovanou funkci a která se umísťuje do HTML stránky. Příkladem portletu může být: vyhledávač zboží, aktuální kurzovní lístek a podobně.

Je zvykem používat MVC pattern při návrhu portletů, tedy:

Model - package model obsahuje domény, javovské beany, případně ORM, které portlet využívá

View - package view obvykle obsahuje JSP, případně JSF soubory, které nám zajišťují vykreslování obsahu do okna prohlížeče.

Controller - „řídí“ daný portlet. Často se nejdříve vytvoří services (v Javě je to interface) a následně samotná implementace. Jako příklad uvedu situaci, kdy potřebujeme z nějaké vzdálené databáze, naimportovat data do naší vnitřní databáze. Tento velký importní proces rozdělíme na několik menších: import kurzovních lístků, import zboží, import prodejem a podobně. Vytvoříme si interface importer a následně pro každou kategorii importů vytvoříme vlastní importér, implementující tento interface. Takový programovací přístup nám díky polymorfizmu poskytuje velkou řadu výhod. Můžeme například vytvořit metodu, která bude přijímat jednotlivé importéry a nad nimi volat importovací funkci bez nutnosti instanceof.

Hook

Tady budou obsaženy tak zvané „přípojky“, tedy některá řešení pro celý portál. Můžeme sem umístit například obsah stránky 404, případně kód pro její dynamické generování.

Layout

Liferay podporuje několik „vestavených“ layoutů stránek:



(převzato z liferay.com)

V modulu Layout však můžeme definovat i vlastní layouts (rozložení) stránek.

Template

V tomto modulu je popsán základní vzhled celé aplikace, tedy například: barva pozadí, položky hlavního menu, logo společnosti a podobně.

Osobní zkušenosti a závěr

Liferay používám ve své práci zejména na vývoj velkých informačních systémů, proto bych chtěl popsat jeho výhody/nevýhody z pohledu jak vývojáře, tak samotného uživatele systému.

Výhody

1. Správa webu

Největší výhodou Liferay je integrované uživatelského rozhraní pro správu webu, tedy uživatel (zadavatel) bude schopen provádět některé úpravy webu pouze s minimálními znalostmi o webových aplikacích. Uživatel bude schopen například:

- Vytvářet nové webové stránky, které se budou automaticky přidávat do menu
- Nastavovat layouty jednotlivým stránkám
- Přidávat hotové portlety na stránky
- Přidávat vlastní webové obsahy na stránky
- Nastavovat SEO jednotlivým stránkám

2. Hotová řešení

Liferay již poskytuje naprogramované komponenty, jako jsou například:

- Forum
- Agregátor obsahu webu
- Vyhledávání

3. Bezpečnost

Liferay poskytuje široké možnosti nastavení pro bezpečnost, jako jsou například:

- Vytváření a správa uživatelských rolí
- Autorizace a autentifikace
- Většina dat je před uložením do databáze zakódována

4. Kompatibilita

Kompatibilita téměř se všemi aplikačními servery (TomCat, GlassFish, JBoss...) a databázemi (Oracle, MySql,...)

Nevýhody

Nevýhody jsem se pokusil seřadit podle priorit, co mě osobně vadí nejvíce:

1. Chyby v systému

V takhle robustního systému jako je Liferay, se občas vyskytnou chyby, které programátorovi značně komplikují vývoj aplikace. Jedná se o čistě chyby Liferaye, navíc, pokud uvažíme použití některých frameworků, jako jsou například Spring a Hibernate, pak aplikační server, databáze a verze Javy, může se lehce stát, že některé komponenty zkrátka nebudou úplně kompatibilní. Důsledkem jsou pak nečekané chyby. Osobně mám zkušenosti s jednou nepříjemnou chybou: Liferay po deploydu aplikace generuje web.xml soubor a z nějakého důvodu prohodil spring listener s liferay listenerem, ve výsledku aplikace přestala fungovat. Chyby takového typu značně komplikují život.

2. Absence větší podpory

Dalším nepříjemným zjištěním, byla slabá podpora Liferay. Zkrátka, tento portál není tak rozšířen, jako například Spring a hledání odpovědí na některé otázky bývá občas dost komplikované.

3. Kontrola systému

Občas je potřeba provést nějaké úpravy v obsahu, generovaném Liferayem. Potřeboval jsem v agregátoru obsahu změnit zobrazení datumu z YY.MM.DD na YYYY.MM.DD. Bohužel se mi dodnes nepodařilo vyřešit tento „jednoduchý“ problém.

4. Performance

Liferay spotřebuje poměrně velké množství paměti (řádově několik GB v případě menší aplikace) a hodně CPU. S tím je třeba počítat již od začátku. Pokud bychom ale vyvíjeli korporátní weby, tak náklady na větší nároky u HW jsou téměř zanedbatelné ve srovnání s náklady na vývoj SW a náklady na její programátorskou údržbu.

Slovo závěrem

Liferay je určitě zajímavé řešení a myslím, že každý J2EE developer by měl o této možnosti měl vědět. O Liferay má smysl uvažovat zejména, pokud vyvíjíme větší webovou aplikaci a vyžadujeme možnost údržby ze strany uživatele. V potaz bychom měly brát i popsané záporné vlastnosti.