

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Vytvoření REST služby pro konverzi LaTeX souborů do PDF

TADEÁŠ KYRAL

Vedoucí: Ing. LUKÁŠ ZOUBEK

Obor: Softwarové inženýrství a technologie

Zaměření: žádné

Listopad 2018

Poděkování

V první řadě bych chtěl poděkovat svému vedoucímu JMENO za příjemnou spolupráci a cenné rady při našich konzultacích. Dále bych chtěl poděkovat mé rodině a kamarádům, kteří mě při psaní této práce podporovali.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 2. listopadu 2018

Abstrakt

Práce pojednává o návrhu a vývoji webové služby pro konverzi LaTeXu do PDF, pro uživatele v Moodle a CourseWare. Doplněk má uživatelům usnadnit práci s LaTeXovými soubory přímo skrze portál Moodle, bez potřeby stahování a kompilace na svém stroji. Byla naimplementována služba na server a následně byla zapojena do provozu. Uživatelům portálů se usnadnila práce s LaTeX soubory.

Klíčová slova: PDF, LaTeX, PDF, Moodle, CourseWare, Web service, REST, Java EE

Vedoucí: Ing. LUKÁŠ ZOUBEK
Katedra ekonomiky, manažerství a humanitních věd

Abstract

Text follows...

Keywords: PDF, LaTeX, PDF, Moodle, CourseWare, Web service, REST, Java EE

Title translation: Create REST service for conversion from LaTeX to PDF

Obsah

1 Úvod	1
1.1 Cíle práce	1
2 Teorie a technologie	2
2.1 REST	2
2.2 LaTeX	3
2.3 Java EE	3
2.4 PDF	3
2.5 FURPS+	4
3 Analýza	6
3.1 Požadavky	6
3.1.1 Funkční požadavky	6
3.1.2 Nefunkční požadavky	7
3.2 Existující řešení	7
3.2.1 OverLeaf	8
3.2.2 BlueLaTeX	8
3.2.3 ScienceSoft	8
3.2.4 ShareLaTeX	8
3.2.5 Porovnání	8
3.3 Kompilátory	9
3.3.1 TeXLive	9
4 Návrh	10
4.1	10
5 Implementace	11
6 Závěr	12
Literatura	13
A Seznam zdrojů	14
B Seznam zkratk	15
C Testování	16

Obrázky

2.1 Struktura objektů	4
4.1 Vizualizace průchodu aplikací ..	10

Tabulky

3.1 Funkční požadavky	6
3.2 Nefunkční požadavky	7

Kapitola 1

Úvod

LaTeX patří v akademickém prostředí mezi velmi oblíbené typografické systémy a je hojně využíván ke psaní skript a odborných prací. Mnoho akademiků využívá LaTeX i k vytváření materiálů pro studenty, primárně v oblasti matematiky, jelikož nabízí jednoduché nástroje ke psaní vzorců. Hlavním zdrojem materiálů pro studenty jsou portály Moodle a CourseWare, kam vyučující dokumenty nahrávají a můžou je tam i upravovat. Nynější editor podporuje jenom řádkové příkazy a není schopen zpracovat celý LaTeX dokument natož s více zdrojovými soubory. Tento stav mnoha učitelům nevyhovuje, protože by chtěli svoje dokumenty upravovat a přímo kompilovat v prohlížeči bez potřeby je stahovat a následně zase nahrávat.

Práce se dělí na čtyři hlavní části. Nejdříve proběhne seznámení s potřebnou teorií a technologiemi, dále budou specifikovány požadavky na službu z čehož se najdou již podobné existující řešení a porovnají se s požadavky. Následně se přistoupí k návrhu samotné aplikace.

1.1 Cíle práce

Hlavním cílem bakalářské práce je implementovat webovou službu, která bude schopna komunikovat pomocí REST Api. Služba bude dělat konverzi LaTeX souborů do PDF. K tomu vedoucí jednotlivé dílčí cíle jako analýza existujících řešení, návrh, implementace a v neposlední řadě testování.

Kapitola 2

Teorie a technologie

V této části si přiblížíme technologie potřebné k návrhu a vývoji webové služby specifikované v zadání práce. Postupně budou vysvětleny všechny zásadní pojmy, které pomůžou čtenáři doplnit znalosti.

2.1 REST

Representational state transfer (dále jen REST) je architektura pro komunikaci mezi distribuovanými systémy. Termín zavedl R. T. Fielding ve své disertační práci[1], kde toto rozhraní bylo taktéž popsáno a vymezeno. Mimo jiné stanovil i těchto 5 základních pravidel, které by měly být dodrženy, aby se aplikace mohla nazývat RESTful[2]:

- Klient-Server(Client-Server) - Toto omezení staví na principu oddělení zodpovědností (Separation of Concerns), nebo-li je klientská část starající se o uživatelské rozhraní a serverová část přistupující k databázi. Zlepšuje škálovatelnost systému a zjednodušuje použitelnost na různých platformách.
- Bezstavovost(Stateless) - Každý požadavek musí přenášet všechna související data, server totiž neuchovává žádné informace o nynějším spojení a každý požadavek bere jako nový.
- Keš(Cache) - Data přenášená v odpovědi mohou být označena jako kešovatelná, tudíž si je klient může uložit a kdykoliv použít znova.
- Jednotné rozhraní(Unified interface) - Základem tohoto omezení je princip HATEOAS (Hypermedia As The Engine Of Application State), které říká, že klient nepotřebuje znát pravidla komunikace dopředu a data musí obsahovat odkazy na další data v aplikaci. Jasně definovaná adresa zdroje (např. URI), reprezentace přenášených dat (např. HTML), typ média (např. JSON)
- Vrstvený systém(Layered system) - Přidáním vrstev se aplikace, kde každá vrstva je izolovaná a může komunikovat jenom se soudějícími vrstvami, zpřehlední a zlepší se její škálovatelnost.

Nejčastějším typem protokolu využívající tuto architekturu je Hypertext Transfer Protocol(HTTP). Pomocí třech hlavních metod GET, POST, DELETE v požadavku poslaného z klienta server buď odpovídající vrátí požadovaná data, přijme data poslané v těle a například uloží do databáze nebo data vymaže.

2.2 LaTeX

Vychází z typografického sázecího systému TeX, který takto popisuje Pavel Satrapa ve své knize [3]: „Patří do rodiny tak zvaných značkovacích jazyků (markup languages) a dal by se zjednodušeně charakterizovat jako programovací jazyk pro sazbu textů. Jeho základním vstupem je textový soubor, který obsahuje jak sázený dokument, tak příkazy ovlivňující sazbu. Určité znaky mají přiřazen speciální význam a jejich prostřednictvím jsou v textu odlišeny řídicí konstrukce. Typickým příkladem je zpětné lomítko, jímž začínají příkazy.“

LaTeX je rozšíření TeXu o balíček přednastavených řídicích konstrukcí. Hlavní cílem těchto systémů je jednoduchost pro psaní matematických a jiných vzorců. Ovšem je také velmi oblíben kvůli silné možnosti upravovat dokumenty ke svému zalíbení.

Příkazy a text se píše do souborů s příponou tex tyto soubory musí dodržovat jistou stromovou strukturou. V kořenu stromu je jeden hlavní soubor, do kterého jsou vnořovány další. To napomáhá přehlednosti velkých dokumentů a také používání již vytvořených. Následně se musí všechny soubory zkompileovat. Výstupním souborem můžou být různé formáty např. PDF, DVI aj.

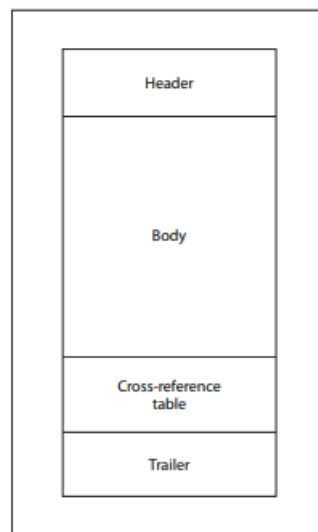
2.3 Java EE

Tato platforma je rozšířením standardní Javy SE a je primárně určená pro vývoj webových aplikací. Java EE potřebuje ke svému fungování aplikační server, který se stará o požadavky, komunikuje s databází apod. Hlavním výstupem aplikace je HTML generované na serveru a zpracované na klientovi, to je rozdíl oproti standardní Javě, kde vše probíhá na klientovi.

2.4 PDF

Portable Document Format(dále jen PDF), jak už název napovídá jedná se o formát dokumentů, jejichž hlavním cílem je poskytování nezávislosti na platformě. Soubor se skládá ze čtyř komponent:

- Objekty(Objects) - Základní jednotka celého souboru např. Pole, Čísla, Řetězce znaků. Jednotlivé objekty se popisují množinou znaků, která je definována lexikálními konvencemi.
- Struktura objektů(File Structure) - Soubor je rozdělen na čtyři části:
 - hlavička - identifikuje verzi PDF
 - tělo - obsahuje použité objekty, které reprezentují obsah dokumentu
 - tabulka odkazů - obsahuje odkazy na objekty v podobě počtu bytů od začátku souboru, kvůli náhodnému přístupu bez potřeby číst celý soubor
 - závěrečná sekce - udává pozici tabulky odkazů a speciálních objektů
- Struktura dokumentu(Document Structure) - Popisuje hierarchickou strukturu objektů v těle dokumentu.



Obrázek 2.1: Struktura objektů

- Content stream - Objekt v kterém se nacházejí instrukce k vykreslování grafických elementů.

2.5 FURPS+

Vzniklo rozšířením klasifikací požadavků FURPS(**F**unctionality, **U**sability, **R**eliability, **P**erformance, **S**upportability) s kterou přišel Robert Grady v roce 1992. Znaménko "+" ve zkratce přidává požadavky a omezení na design, implementaci, rozhraní a hardware.

- Funkční požadavky
 - Functionality - Požadavky popisující všechny hlavní prvky produktu i důležité aspekty z pohledu architektury např. lokalizovaný systém pro více jazyků.
- Nefunkční požadavky
 - Usability - Zaměřuje se na uživatelskou přívětivost nejenom samotné aplikace, ale i dokumentace apod., týkající se estetiky a konzistence
 - Reliability - Spolehlivost systému v podobě času fungování, správnosti fungování a četnosti výpadků
 - Performance - Vypovídá o výkonnosti systému, jak rychle dokáže zpracovávat požadavky, nastartovat atd.
 - Supportability - Popisuje testovatelnost, škálovatelnost, konfigurovatelnost...
 - Design - Omezení na design systému např. požadavek na relační databázi
 - Implementation - Specifikuje typ programovacího jazyku, platformu apod.
 - Interface - Komunikace s externími systémy

- Physical - Definuje požadavky na hardware, na kterém daný software poběží i co se týče velikosti

Toto rozdělení nám pomáhá identifikovat požadavky. Přispívá k vyšší kvalitě systému a snižuje pravděpodobnost přehlédnutí funkcionality.

Kapitola 3

Analýza

3.1 Požadavky

Nyní představíme požadavky kladené na aplikaci. Ty jsou buď požadované zadávajícím nebo odvozené z potřeb, ke kterým bude služba používána. Ale i z omezení plynoucích z prostředí v jakém bude provozována, v tomto případě pro fakultu státní vysoké školy, jmenovitě Fakulta elektrotechnická, ČVUT.

Za pomoci metody FURPS+ rozdělíme požadavky a omezení na funkční a nefunkční.

3.1.1 Funkční požadavky

Typ	Požadavky
Funkčnost	<ul style="list-style-type: none">• Webová služba s REST rozhraním pro komunikaci• Umět přijmout požadavky a soubory• Zkompilování LaTeX souborů s požadovaným nastavením do PDF• Uložení PDF a jeho poskytnutí• Uživatel může zvolit tyto nastavení:<ul style="list-style-type: none">• Přidat vodoznak na každou stranu PDF• Nakonec souboru přidat stránku s předem stanoveným obsahem• Výsledný soubor PDF bude chráněný• Výsledný soubor PDF nebude kopírovatelný nebo tisknutelný

Tabulka 3.1: Funkční požadavky

Služba má jediný a úzký účel a z toho i plyne menší množství funkcionalit.

3.1.2 Nefunkční požadavky

Typ	Požadavky
Použitelnost	<ul style="list-style-type: none"> Dokumentace k rozhraní na swagger Přístup skrz REST api, pouze s tokenem
Spolehlivost	<ul style="list-style-type: none"> Služba není kritická Uptime 95% Ochrana proti špatnému vstupu Autorizace pomocí tokenů
Výkon	<ul style="list-style-type: none"> Zvládnutí obslužení desítky požadavků najednou Ukládání výsledných dokumentů po dobu jednoho měsíce Maximální doba tvorby dokumentu 5 minut
Podporovatelnost	<ul style="list-style-type: none"> Dokumentace k rozhraní na swagger Služba může být rozšířena o kompilování samotného TeXu a může podporovat vytváření i jiných formátů z LaTeXu
Implementace	<ul style="list-style-type: none"> Platforma - Java EE Komunikace - REST Api
Rozhraní	<ul style="list-style-type: none"> Komunikuje s Moodle a CourseWare, tyto portály posílají data
Fyzické	<ul style="list-style-type: none"> Musí být provozováno na serverech ČVUT, kvůli ochraně dat

Tabulka 3.2: Nefunkční požadavky

Z tabulky je vidět, že služba není nijak kritická a působí jenom jako doplněk do výše zmíněných portálů. Musí ovšem splňovat vyšší bezpečnostní nároky zapříčiněné prostředím v jakém se bude používat.

3.2 Existující řešení

Na základě stanovených požadavků v minulé kapitole přejdeme k analýze již existujících řešení. Mometálně uživatelé Moodle mohou vkládat do svých souborů řádkové příkazy, což není úplně dostačující a nesplňuje to požadavky. Samozřejmě se dají používat pro vytváření PDF z LaTeX souborů kompilátory, které si můžete stáhnout a používat lokálně. To ovšem není předmětem této práce, a proto se podíváme na webové služby, které více odpovídají potřebám. Pár vybraných si představíme.

3.2.1 OverLeaf

<https://www.overleaf.com/for/universities> Placená služba pro tvorbu LaTeX dokumentů, která umožňuje i používání s omezeními zadarmo. Je velmi oblíbená hlavně kvůli hezkému prostředí pro tvorbu dokumentů a jejich správu. Také nabízí výhody pro určité zájmové skupiny, nejzajímavější vzhledem k tématu této práce je předplatitelská služba OverLeaf Commons <https://www.overleaf.com/for/universities>. Ta poskytuje sdílené prostředí se všemi výhodami pro zaměstnance a studenty univerzity.

3.2.2 BlueLaTeX

<http://www.bluelatex.org/> Open source služba pro kompilaci LaTeX souborů. Kompilovat můžete na jejich serveru nebo nabízejí kód pro spuštění na vlastním. Kromě samotné serverové implementace je k dispozici i webový klient. Vše je pouze zatím v Beta verzi a samotní tvůrci varují před možnými nedostatky a problémy. Hlavním lákadlem je real time spolupráce více lidí na jednom dokumentu a také možnost provozovat server lokálně.

3.2.3 ScienceSoft

<http://sciencesoft.at/latex/index?lang=en> Starší služba, která mimo kompilace LaTeX souborů vložených přímo na stránky poskytuje i jiné rozhraní pro posílání souborů a to skrz REST Api a SOAP. Jak bylo řečeno, tak tato služba je už starší tudíž pro kompilaci používá TexLive 2008.

3.2.4 ShareLaTeX

<https://github.com/sharelatex/clsi-sharelatex> Velmi podobný BlueLaTeX, tedy open source nabízející jejich hostovanou verzi nebo lokální verzi pro vlastní potřebu, obě verze obsahují mnoho funkcí včetně grafického prostředí pro uživatele. Pod jménem ShareLaTeX se vyskytuje jenom výše zmíněné, <https://www.overleaf.com/for/enterprises> ovšem společně s OverLeaf také spravují službu Pro, která je určená pro firmy, ale i univerzity. Ta se pyšní možností nasazení na vlastních serverech, správou uživatelů, podporou a zabezpečením.

3.2.5 Porovnání

Každé z těchto řešení má své problémy, ať už že je zastaralé a neaktualizované nebo nesplňuje zanalyzované požadavky (kapitola 3.1). Do první skupiny patří ScienceSoft a BlueLaTeX, kde první z jmenovaných ani neposkytuje kód pro implementaci na lokálním serveru a druhý je v nedodělaném stavu, což může vést k nefunkčnosti a bezpečnostním rizikům. Služba Overleaf Commons nesplňuje stejný požadavek jako řešení od ScienceSoft, ale nabízí zajímavé prostředí pro vytváření a správu LaTeX dokumentů pro studenty i zaměstnance, o čemž by univerzita mohla popřemýšlet. Nejnadějnější možností se zdá být ShareLaTeX, který poskytuje k použití i jenom backend pro kompilaci LaTeX a komunikaci, ale celý je implementovaný v CoffeeScriptu, což si rozporuje s požadavkem na implementaci, kde je Java EE.

■ 3.3 Kompilátory

Nejdůležitější částí celé aplikace bude kompilátor, který samozřejmě provádí kompilaci, ale může poskytnout i podporu pro operace s výsledným PDF. Tyto operace vycházejí ze stanovených funkčních požadavků (kapitola 3.1.1). Jelikož kompilace bude probíhat na serveru, kde bude nainstalován Linuxový operační systém.

■ 3.3.1 MikTeX

<https://miktex.org/kb/just-enough-tex> “Just enough TeX” neboli jen to nejnútnejší poskytuje tato instalace, tedy bez zbytečných balíčků navíc. Ovšem MikTeX nabízí dostahovávat balíčky za běhu, podle potřeby.

■ 3.3.2 TeXLive

<https://www.tug.org/texlive/pkginstall.html> Existují dvě různé cesty jak nainstalovat TeXLive a od toho se odvíjí správa balíčků. Verze Native TeX Live a distribuce přímo pro daný operační systém. Liší se ve způsobu aktualizací balíčků a základní po instalaci. V Native verzi probíhají aktualizace pomocí TeX Live Manager nebo-li tlmgr

■ 3.3.3 Porovnání

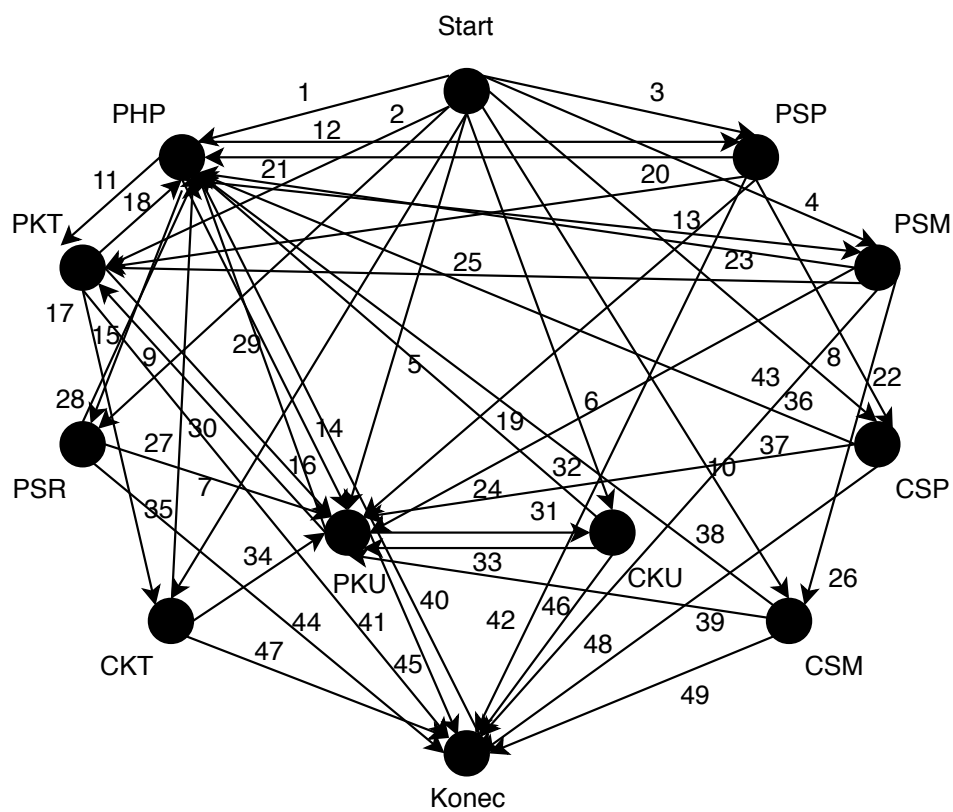
Oba kompilátory podporují aspoň n

Kapitola 4

Návrh

4.1

$$R = (A \vee B \vee C \vee D)$$



Obrázek 4.1: Vizualizace průchodu aplikací



Kapitola 5

Implementace



Kapitola 6

Závěr

Text follows



Literatura

- [1] FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, PhD Dissertation, University of California, Irvine, 2000. Online, Přístup 17.11.2018.
- [2] HESAM, MASNE, S., HARRY, AND RAJAN. REST API Tutorial. <https://restfulapi.net/>. Online, Přístup 17.11.2018.
- [3] SATRAPA, P. *LaTeX pro pragmatiky*. 2011.

Příloha A

Seznam zdrojů

- Obrázek 3.1 – Vytvořeno autorem 15. 4. 2018.
- Obrázek 4.1 – Vytvořeno autorem 15. 4. 2018.
- Obrázek 7.1 – Vytvořeno autorem 15. 4. 2018.
- Obrázek 7.2 – Vytvořeno autorem 12. 3. 2018.
- Obrázek 8.1 – Vytvořeno autorem 12. 3. 2018.
- Obrázek 8.2 – Vytvořeno autorem 12. 3. 2018.
- Obrázek 10.1 – Vytvořeno autorem 17. 5. 2018.
- Obrázek 10.2 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 7.1 – Vytvořeno autorem 26. 2. 2018.
- Tabulka 7.2 – Vytvořeno autorem 26. 2. 2018.
- Tabulka 9.1 – Vytvořeno autorem 8. 5. 2018.
- Tabulka 9.2 – Vytvořeno autorem 8. 5. 2018.
- Tabulka 9.3 – Vytvořeno autorem 8. 5. 2018.
- Tabulka 10.1 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 10.2 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 10.3 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 10.4 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 10.5 – Vytvořeno autorem 17. 5. 2018.
- Tabulka 10.6 – Vytvořeno autorem 18. 5. 2018.
- Tabulka 10.7 – Vytvořeno autorem 18. 5. 2018.
- Tabulka 11.1 – Vytvořeno autorem 18. 5. 2018.



Příloha B

Seznam zkratek

LMS – Learning Management System

SWOT – Strengths, Weaknesses, Opportunities, Threats

GDPR – General Data Protection Regulation

HTTP – Hypertext Transfer Protocol

API – Aplikační rozhraní

CW – CourseWare

JPA – Java Persistence API

JDBC – Java Database Connectivity



Příloha C

Testování