

Sem vložte zadání Vaší práce.



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Bakalářská práce

Webový nástroj pro kolaborativní editaci textů

Jiří Šimeček

Katedra softwarového inženýrství

Vedoucí práce: Ing. Petr Špaček, Ph.D.

5. dubna 2018

Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstraňte tento příkaz.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. dubna 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Jiří Šimeček. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Šimeček, Jiří. *Webový nástroj pro kolaborativní editaci textů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato práce se zabývá problémem kolaborativní editace textů a porovnává jednotlivé známé algoritmy, které tento problém řeší. Dále se zabývá návrhem a implementací prototypu pomocí jednoho z vybraných algoritmů.

Klíčová slova návrh webové aplikace, kolaborativní editace textů, web v reálném čase, Javascript, ReactJS, NodeJS

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Technologie	5
2.2 Pull a push technologie	8
2.3 Algoritmy pro kolaborativní editaci	8
2.4 Existující řešení	8
2.5 Funkční požadavky	8
2.6 Nefunkční požadavky	8
2.7 Uživatelské případy	8
2.8 Doménový model	8
3 Návrh	9
4 Realizace	11
5 Testování	13
Závěr	15
Bibliografie	17
A Seznam použitých zkratk	19
B Obsah přiloženého CD	21

Seznam obrázků

Úvod

Webové aplikace, které komunikují s uživatelem v reálném čase dnes nabývají na oblibě. Uživatel již běžně očekává, že se mu na webových stránkách zobrazují nejrůznější upozornění, či se dokonce aktualizují celé části webové stránky. Nově se začínají objevovat webové nástroje pro kolaborativní spolupráci nad texty (případně nad jinými multimédii), které kombinují myšlenku tvorby obsahu webu uživateli a právě odezvu aplikace v reálném čase.

Výstupem této práce je všeobecně nasaditelná komponenta, která bude použita jako jedna z komponent projektu webového IDE s pracovním názvem Laplace-IDE. Komponenta je také určena pro potřeby vývojářů, kteří chtějí vytvořit kolaborativní webový nástroj a nechtějí ho vytvářet od nuly.

Toto téma jsem si zvolil, jelikož většina doposud existujících kolaborativních textových nástrojů je postavena nad uzavřeným kódem nebo nad knihovnamy, jejichž vývoj byl ukončen. Neexistují tak nástroje, či knihovny, které by bylo možné bez větších problémů použít pro vlastní projekty.

V této práci se zabývám analýzou problému kolaborativní spolupráce, porovnáním a výběrem vhodných existujících algoritmů a technologií, návrhem znovupoužitelné komponenty a implementací prototypu včetně navržené komponenty.

Tato práce dále pokračuje v následující struktuře: Nejprve se v části 1 zabývám analýzou a výběrem vhodných algoritmů, z které pak přecházím k návrhu komponenty a prototypu v části 2. Navržený prototyp dále v části 3 implementuji a na konec nad výslednou implementací v bodě 4 provádím uživatelské testování.

Cíl práce

Cílem rešeršní části práce je seznámení se zadanými technologiemi, které budou využity při implementaci prototypu. Dalším cílem je analýza problematiky kolaborativní editace textů a rozbor existujících webových real-time protokolů.

Cílem praktické části je navržení modelu pro uložení dat, implementace prototypu a následné uživatelské otestování a vyhodnocení kvality implementovaného řešení.

Analýza

2.1 Technologie

V této část popisují použité technologie vycházející z nefunkčních požadavků, které jsou uvedeny v kapitole 2.6.

2.1.1 HTML5

HTML5 je značkovací jazyk používaný pro reprezentaci a strukturování obsahu na internetu (přesněji na WWW¹). Jedná se již o pátou verzi standardu HTML² (doporučená verze podle W3C³ v roce 2018 je HTML 5.2 [1]). Tato verze do standardu přidává mimo jiné nové elementy, atributy a funkcionalitu. Pod pojem HTML5 také často zařazujeme rozsáhlou množinu moderních technologií, které umožňují tvorbu více rozmanitých a mocných webových stránek a aplikací. [2]

HTML vytvořil Tim Berners-Lee a HTML standart byl definován ve spolupráci s organizací IETF⁴ v roce 1993 [3]. Od roku 1996 převzala vývoj HTML standartu organizace W3C [4]. Roku 2008 se k W3C přidala organizace WHATWG⁵ a započali vývoj standartu HTML5, který společně vydali v roce 2014 [5].

2.1.2 JavaScript

JavaScript pod pracovním názvem LiveScript vytvořil Brendan Eich v roce 1995, kdy působil jako inženýr ve firmě Netscape. Přejmenování na JavaScript bylo marketingové rozhodnutí a mělo využít tehdejší popularity programova-

¹World Wide Web

²Hypertext Markup Language

³The World Wide Web Consortium

⁴Internet Engineering Task Force

⁵Web Hypertext Application Technology Working Group

cího jazyka Java od Sun Microsystem a to přesto, že tyto jazyky spolu téměř nesouvisí. Javascript byl poprvé vydán jako součást prohlížeče Netscape 2 roku 1996. Později téhož roku představila firma Microsoft pro svůj prohlížeč Internet Explorer 3 jazyk JScript, který byl JavaScriptu velice podobný. [6]

Roku 1997 vydala organizace ECMA⁶ první verzi standartu ECMAScript, který z původního JavaScriptu a JScriptu vycházel [7]. Tento standart prošel v roce 1999 rozsáhlou aktualizací jako ECMAScript 3, která je bez větších změn používána dodnes [6].

Další verze ECMAScript standartu podle [8] jsou:

- ECMAScript 5 z roku 2009,
- ECMAScript 5.1 z roku 2011 (ISO/IEC 16262:2011),
- ECMAScript 2015,
- ECMAScript 2016,
- ECMAScript 2017 a
- připravovaný standart ECMAScript 2018.

Dnes pod označením JavaScript běžně chápeme právě standardizovaný ECMAScript [6] a i já ho tak budu dále používat.

Javascript je navržen k běhu jako skriptovací jazyk v hostitelském prostředí, které musí zajistit mechanismy pro komunikaci mimo toho prostředí. Nejčastějším hostitelským prostředím je webový prohlížeč, ale Javascript můžeme nalézt i na místech jako je Adobe Acrobat, Adobe Photoshop, SVG vektorová grafika, serverové prostředí NodeJS, NoSQL databáze Apache CouchDB, nejrůznější vestavěné systémy a další. [6]

Javascript je více paradigmový, dynamický jazyk s datovými typy, operátory, standardními vestavěnými objekty a metodami. Jeho syntaxe je založena na jazycích Java a C. JavaScript podporuje objektově orientované programování pomocí objektových prototypů namísto tříd jako je tomu například u jazyku Java. Dále také podporuje principy funkcionální programování – funkce jsou také objekty. [6]

2.1.3 Node.js

Node.js je JavaScriptové běhové prostředí (anglicky runtime environment), které používá událostmi řízenou architekturu umožňující asynchronní přístup k I/O⁷ operacím. Tato architektura umožňuje optimalizovat propustnost a škálovatelnost webových aplikací s mnoha I/O operacemi, ale také webových

⁶European Computer Manufacturer's Association

⁷Vstupní/výstupní

aplikací ve skutečném čase (například komunikační programy nebo hry v prohlížeči). [9]

Toto je v kontrastu s dnešními více známými modely souběžnosti, kde se využívají vlákna operačního systému. Síťová komunikace založená na vláknech je relativně neefektivní a její použití bývá velmi obtížné. [10]

Node.js využívá V8 JavaScript interpret vytvořený společností Google pod skupinou The Chromium Project pro prohlížeč Google Chrome a ostatní prohlížeče postavené na Chromium⁸ [11]. V8, který je napsaný v C++, kompiluje JavaScriptový kód do nativního strojového kódu namísto jeho interpretace až za běhu programu. To umožňuje vytvořit rychlé běhové prostředí, které nemusí čekat na překlad potřebného kódu. [9]

2.1.4 React

React je Javascriptová knihovna pro tvorbu uživatelského rozhraní [12]. React byl vytvořen Jordanem Walkem, inženýrem ve společnosti Facebook, a byl poprvé použit v roce 2011. Původně byl React určen výhradně pro použití na Facebook Timeline, ale Facebook inženýr Pete Hunt se rozhodl React použít i v aplikaci Instagram. Postupně tak React zbavil závislostí na kód Facebooku a tím napomohl vzniku oficiální React knihovny. React byl představen veřejnosti jako knihovna s otevřeným zdrojovým kódem v květnu roku 2013. [13]

Základním prvkem Reactu jsou takzvané komponenty, které přijímají neměnné vlastnosti a mohou definovat vlastní stavové proměnné. Na základě těchto vlastností a stavu, pak komponenta může rozhodnout co bude jejích výstupem pro uživatele (pomocí metody render). Tato vlastnost se nazývá jednosměrný datový tok (anglicky One-way data flow) a architekturu, kterou React implementuje, nazýváme Flux (ta je součástí Reactu už od samého počátku). [12] Existují však komunitou vytvořené alternativní nástroje, které řeší datový tok v aplikaci, jako je například knihovna Redux [14].

2.1.5 MongoDB

MongoDB je nerelační databáze s otevřeným zdrojovým kódem vyvíjena společností MongoDB, Inc. Díky zjednodušené reprezentaci dat pomocí dokumentů a jejich rychlému mapování na JSON⁹ není potřeba využívat složité ORM¹⁰ nástroje pro mapování dat na objekty, jako tomu je například u SQL¹¹ databází, a tím umožňuje urychlit celkový vývoj aplikací. [15]

Pro MongoDB jsem se rozhodl převážně s ohledem na jeho rychlost, která se bude u aplikace pro editaci textů ve skutečném čase hodit, ale také pro jeho rozšířenou komunitu mezi vývojáři [16, 17].

⁸Prohlížeč s otevřeným zdrojovým kódem od společnosti Google

⁹JavaScript Object Notation

¹⁰Object-relational mapping

¹¹Structured Query Language

2.2 Pull a push technologie

2.2.1 Pull technologie

2.2.2 Push technologie

2.2.2.1 Long pooling

2.2.2.2 Data streaming

2.2.2.3 WebSocket

2.3 Algoritmy pro kolaborativní editaci

2.3.1 Operational transformation

2.3.2 Differential synchronization

2.4 Existující řešení

2.4.1 Apache Wave

2.4.2 Google Apps

2.4.3 Etherpad

2.4.4 Codeshare.io

2.5 Funkční požadavky

2.6 Nefunkční požadavky

2.7 Uživatelské případy

2.8 Doménový model

Návrh

V lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem
v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem
v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem
v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem
v lorem v lorem v lorem v lorem v lorem v lorem v lorem v lorem.

KAPITOLA **4**

Realizace

Testování

Závěr

Bibliografie

1. LEITHEAD, Travis; EICHOLZ, Arron; MOON, Sangwhan; DANILO, Alex; FAULKNER, Steve. *HTML 5.2* [online]. 2017 [cit. 2018-04-04]. Dostupné z: <https://www.w3.org/TR/2017/REC-html52-20171214/>. W3C Recommendation. W3C.
2. MOZILLA; INDIVIDUAL CONTRIBUTORS. *HTML5* [online]. 2018 [cit. 2018-04-04]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>.
3. BERJON, Robin; NAVARA, Erika Doyle; LEITHEAD, Travis; PFEIFFER, Silvia; HICKSON, Ian; O'CONNOR, Theresa; FAULKNER, Steve. *Hypertext Markup Language (HTML) A Representation of Textual Information and MetaInformation for Retrieval and Interchange* [online]. 1993 [cit. 2018-04-04]. Dostupné z: <https://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>. Internet Draft. IIIR Working Group (IETF).
4. RAGGETT, Dave. *HTML 3.2 Reference Specification* [online]. 1997 [cit. 2018-04-04]. Dostupné z: <https://www.w3.org/TR/2018/SPSD-html32-20180315/>. W3C Recommendation. W3C.
5. BERJON, Robin; NAVARA, Erika Doyle; LEITHEAD, Travis; PFEIFFER, Silvia; HICKSON, Ian; O'CONNOR, Theresa; FAULKNER, Steve. *HTML5* [online]. 2014 [cit. 2018-04-04]. Dostupné z: <http://www.w3.org/TR/2014/REC-html5-20141028/>. W3C Recommendation. W3C.
6. MOZILLA; INDIVIDUAL CONTRIBUTORS. *A re-introduction to JavaScript (JS tutorial)* [online]. 2018 [cit. 2018-04-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript.
7. *ECMAScript: A general purpose, cross-platform programming language* [online]. 1997 [cit. 2018-04-04]. Dostupné z: <http://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262,%201st%20edition,%20June%201997.pdf>. ECMA Standard. ECMA.

8. MOZILLA; INDIVIDUAL CONTRIBUTORS. *JavaScript language resources* [online]. 2018 [cit. 2018-04-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Resources.
9. ORSINI, Lauren. What You Need To Know About Node.js. *Readwrite* [online]. 2013 [cit. 2018-04-04]. Dostupné z: <https://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/>.
10. NODE.JS FOUNDATION. *About Node.js®* [online]. 2018 [cit. 2018-04-04]. Dostupné z: <https://nodejs.org/en/about/>.
11. NODE.JS FOUNDATION. *ECMAScript 2015 (ES6) and beyond* [online]. 2018 [cit. 2018-04-04]. Dostupné z: <https://nodejs.org/en/docs/es6/>.
12. FACEBOOK INC. *A JavaScript library for building user interfaces* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://reactjs.org>.
13. FISHER, Bill. *How was the idea to develop React conceived and how many people worked on developing it and implementing it at Facebook?* [online]. 2015 [cit. 2018-04-05]. Dostupné z: <https://www.quora.com/How-was-the-idea-to-develop-React-conceived-and-how-many-people-worked-on-developing-it-and-implementing-it-at-Facebook>.
14. COMMUNITY. *Read Me* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://redux.js.org>.
15. MONGODB, INC. *MongoDB and MySQL Compared* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://www.mongodb.com/compare/mongodb-mysql>.
16. WEINBERGER, Claudius. NoSQL Performance Benchmark 2018: MongoDB, PostgreSQL, OrientDB, Neo4j and ArangoDB. *Performance Zone* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://dzone.com/articles/nosql-performance-benchmark-2018-mongodb-postgresq>.
17. STACK EXCHANGE INC. *Stack Overflow Developer Survey 2018* [online]. 2018 [cit. 2018-04-05]. Dostupné z: <https://insights.stackoverflow.com/survey/2018/#technology-databases>.

Seznam použitých zkratek

GUI Graphical user interface

XML Extensible markup language

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	exe.....	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf.....	text práce ve formátu PDF
	thesis.ps.....	text práce ve formátu PS