

Gymnázium, Brno, třída Kapitána Jaroše 14

---

# **Vývoj aplikace na kontrolování změn na webových stránkách**

**Závěrečná maturitní práce**

Vedoucí práce:  
Mgr. Marek Blaha

Jiří Kalvoda

Blansko 2020



TODO poděkování



Prohlašuji, že jsem tuto práci vyřešil samostatně s použitím literatury, kterou uvádím v seznamu

V Blansku dne 29. prosince 2019

.....



**Abstract**

Kalvoda, J.

Development an aplication for checking changes on web pagesabstract

**Abstrakt**

Kalvoda, J. Vývoj aplikace na kontrolování změn na webových stránkách

Tato závěrečná práce popisuje vývoj a použití aplikace na monitorování změn na webových stránkách. Aplikace je vyvíjena v jazyce c++ pomocí knihovny Qt. Díky tomu se jedná o multiplatformní software. Je dostupná včetně zdrojového kódu pod licencí GNU LGPL. Tato závěrečná práce obsahuje popis jejího fungování, implementace a použitého softwaru při jejím vývoji.





## Obsah

<b>1</b>	<b>Úvod a cíl práce</b>	<b>11</b>
1.1	Úvod do problematiky . . . . .	11
1.2	Cíl práce . . . . .	11
<b>2</b>	<b>Přehled literatury</b>	<b>12</b>
<b>3</b>	<b>Popis fungování a ovládání aplikace</b>	<b>13</b>
3.1	Instalace . . . . .	13
3.2	Seznam stránek na kontrolu . . . . .	13
3.3	Spuštění kontroly, tabulka změn, informační konzole . . . . .	14
3.4	Historie změn a její procházení . . . . .	16
3.5	Grafické porovnávání verzí stránek . . . . .	17
<b>4</b>	<b>Implementace aplikace</b>	<b>18</b>
4.1	Použitý software . . . . .	18
4.2	Objektový model, rozdělení problému . . . . .	18
4.3	Pozadí aplikace . . . . .	18
4.4	Grafické uživatelské rozhraní . . . . .	18



# 1 Úvod a cíl práce

## 1.1 Úvod do problematiky

Webové stránky se mohou neustále měnit, proto je dobré automaticky monitorovat jejich aktualizace. Tato aplikace umožňuje automatizovat tento problém a tím uživateli ušetřit čas a eliminovat lidský chybový faktor.

Aplikace podporuje různé tolerance při načítání a porovnávání změn stránek. Například na stránce, kde se část neustále mění, mohu tuto část vypustit, nebo přímo porovnávat jen nějaké části a podobně. Díky práci s cookies je možné navázat i složitější spojení se serverem a provést definovanou sekvenci úkolů (např. přihlásit se a načíst nějaký soukromý obsah). Historie stránek se může ukládat a pak lze v ní vyhledávat a zjišťovat rozdíly mezi verzemi pomocí grafického porovnávání napojeného na uživatelův oblíbený prohlížeč.

## 1.2 Cíl práce

Cílem této práce je vyvinout funkční aplikaci umožňující zjišťování aktualizací, archivaci a porovnávání webových stránek (případně i jiných dokumentů) a publikovat ji cílovým uživatelům na různých operačních systémech. K aplikaci také bude vypracována rozsáhlá uživatelská i technická dokumentace, která umožní její další vývoj. Umožním tedy dalším programátorům tuto aplikaci pohodlně modifikovat a upravovat dle svých potřeb. Cílem této práce je také aplikaci rozšířit mezi skupinu testovacích uživatelů a použít jejich připomínky a problémy k dalšímu vývoji a stabilizaci aplikace.

## 2 Přehled literatury

## 3 Popis fungování a ovládání aplikace

### 3.1 Instalace

Linux

Windows

macOS

### 3.2 Seznam stránek na kontrolu

Při spuštění si aplikace načte seznam stránek ke kontrole. Ten je obsažen v souboru `pages.json`, který musí být umístěn v adresáři aplikace (respektive v adresáři, kde se aplikace spouští). Soubor musí být validní json. V případě, že soubor neexistuje nebo není validní, aplikace vypíše upozornění. Očekává se, že soubor obsahuje pole struktur. Každá z nich obsahuje informace o jedné stránce, která se má kontrolovat.

#### Adresa stránky a název

Každou stránku je nutné pojmenovat jednoznačným identifikátorem. Proto je nutné u každé stránky definovat položku označenou klíčem `name`. Toto jméno se pak zobrazuje v seznamu změn a také je nutné při vyhledávání v historii. Jméno se také používá jako identifikátor v databázi i jako identifikátor souborů jednotlivých verzí stránky.

Dále je nutné u každé stránky definovat její umístění na webu, tedy její adresu. K tomu slouží položky `server`, `dir`, `file`. Ovšem není zapotřebí definovat všechny tyto položky. Pod klíčem `server` by měla být definována adresa serveru, na kterém je požadovaná stránka včetně přístupového protokolu. Tedy například `https://is.jaroska.cz` nebo pomocí ip adresy může zápis vypadat následovně `http://195.178.65.1` V případě, že není požadovaná stránka na serveru umístěna v jeho kořenové složce, je doporučeno název složky (popřípadě celou cestu několika zanořených složek) umístit do položky `dir`. `file` obsahuje jméno požadovaného souboru včetně přípony V případě, že je požadován přístup základní soubor (`index`) na dané adrese, není potřeba `file` uvádět. Dále zde také můžou být obsaženy parametry stránky, které se předávají metodou `GET`. Stačí je uvést za otazník. Takový zápis může vypadat následovně: `index.php?akce=42&akcicka=0`.

V případě, že je nutné předat stránce informace pomocí protokolu `POST` (typicky při přihlašování na stránky), je možné v zápisu stránky užít klíče `post`.

Nejjednodušší způsob, jak zjistit tyto informace pro požadovanou stránku je využít prohlížeč. Většina prohlížečů totiž umožňuje zobrazit informace o navázaném spojení. Odsud stačí potřebné informace jen zkopírovat. V prohlížečích založených na jádře Chromium lze se pomocí klávesy `F12` dostat do vývojářského panelu. V záložce `Network` je možné najít příslušný soubor, jehož hlavičku je třeba použít.

## Způsoby kontroly změn

U některých stránek často mění její část, i když sledovaný obsah se nezmění. Pro omezení kontrolování je proto vhodné užít v zápisu položky `diff`. Ta by měla obsahovat speciální strukturu popisující způsob ignorování změn<sup>1</sup>. Tato struktura může obsahovat následující položky:

Klíč `ignore` s libovolnou hodnotou znamená, že stránka se vůbec nebude kontrolovat na změny. Toto je vhodné například když se jedná pouze o přihlašovací stránku, která neobsahuje žádaná data.

Tag `ignoreSector` může obsahovat pole struktur. Každá z nich může obsahovat informace o jednom nutném vynechání pomocí tagů `start`, `end` a `countOfEnd`, které znamenají, že text od `start` po `countOfEnd`-tý výskyt řetězce `end` bude při porovnávání vynechán. Začátek i konec může být definován pomocí regulárního výrazu. Na přesnou implementaci regulárních výrazů lze nahlédnout do dokumentace Qt na adrese <https://doc.qt.io/qt-5/qregexp.html>. Tagy `end` a `countOfEnd` nemusí být uvedeny. V takovém případě je `end` nastaveno na konec řádku a `countOfEnd` na 1.

Obdobným způsobem lze použít tag `onlySector`, který také může obsahovat pole struktur složených z `start`, `end` a `countOfEnd`. Při použití tohoto tagu budou porovnávány pouze změny v těchto úsecích (budou ignorovány úseky od začátku k prvnímu výskytu, mezi nimi a od posledního na konec).

Další možností je využít klíče `permutation` s libovolnou hodnotou. Jeho použití znamená, že libovolné permutace znaků budou považovány za stejné. Toto je vhodné v případě, že se na stránce některé objekty náhodně prohazují.

Tyto omezení porovnávání se provádí v zde uvedeném pořadí.

## Skupiny stránek, přihlášení a práce s cookie

Například v případě, že je nutné pro získání informací se na nějakou stránku přihlásit, je možné využít v zápisu stránky položku `cookie`. Pomocí ní lze spojit více stránek do skupiny, ve které si stránky mezi sebou ukládají cookies. Stačí pouze u všech stránek vyplnit stejnou hodnotou tagu `cookie`. Když se některou z stránek nepodaří načíst, v daném průchodu se nebudou načítat ani další stránky ze stejné skupiny.

## 3.3 Spuštění kontroly, tabulka změn, informační konzole

Po spuštění aplikace se zobrazí hlavní okno. Jelikož aplikace by měla běžet jako služba, okno neobsahuje tlačítko zavřít. V případě, že uživatel skutečně chce uzavřít aplikaci a tím i zabránit dalším automatickým kontrolám, je možné aplikaci ukončit z menu kliknutím na `app` → `quit`.

<sup>1</sup>Případně může obsahovat řetězec `"ignore"`, který má stejný efekt jako `{"ignore":1}`

Pro manuální spuštění kontroly je možné kliknout na tlačítko **Start checking**. V případě, že již kontrola běží, po kliknutí se ukončí a ihned začne znovu od začátku. Po kliknutí na **Stop checking** se neprodleně prohledávání ukončí.

Při průchodu se nalezené změny ihned zobrazují do tabulky změn. Tedy i v případě, že je prohledávání ukončeno v průběhu, doposud nalezené změny budou uloženy a zobrazeny.

Aktuální stav průchodu je vidět na stavovém baru nad tlačítky. V případě, že bar má červenou barvu, alespoň jedna stránka nebyla při posledním průchodu úspěšně načtena. Po kliknutí na zaškrťovací políčko **view more information** se zobrazí záznam provedených kontrolách a případně důvod jejich neúspěchu. Všechny chybové hlášky jsou obsaženy na rádcích uvozených několika vykřičníky. Nejčastěji se může uživatel setkat s těmito chybami:

**Connection error - TIME OUT** nastane v případě, že načítání stránky bylo moc pomalé a tedy byl překročen časový limit na přístup na jednu stránku.

**Cookie error - cookie not available** se zobrazí v případě, nastala chyba při načítání některé z předcházejících stránek ze stejné cookie skupiny.

**Connection error - Network access is disabled.** znamená nedostupnost síťového připojení.

**Connection error - Host not found.** informuje o nedostupnosti daného serveru. S největší pravděpodobností se jedná o chybě zadanou stránku, nebo byla přesunutá.

V boxu vpravo od tlačítek je možno nastavit automatické spouštění kontrolování. Do boxu stačí napsat počet minut mezi automatickými kontrolami. Automatické kontroly lze vypnout napsáním do políčka 0. Datum a čas poslední úspěšné (tedy takové, ve které se načetly všechny stránky) kontroly je vidět mezi tlačítky boxem automatické aktualizace. V tomto místě je také vidět čas příští plánované kontroly.

Informace o poslední kontrole a periodě automatické kontrole si aplikace ukládá do souboru databáze v pracovním adresáři aplikace. Při spuštění si tento soubor načte (pokud existuje). Nastavení automatické kontroly tedy vydrží i vypnutí a zapnutí aplikace. V případě, že měla kontrola proběhnout v momentě, kdy byla aplikace vypnutá, proběhne neprodleně po jejím spuštění.

Všechny nalezené změny se zobrazují v tabulce změn umístěné v horní části hlavního okna aplikace. O každé změně se zobrazí řádek obsahující jméno stránky, čas detekování změny a jméno souboru, v němž je uložena aktuální verze. V případě, že uživatel chce data setřídít podle některé z těchto položek, může tak učinit kliknutím na hlavičku daného sloupce tabulky. Kliknutím na jméno souboru se ve výchozím prohlížeči otevře daná verze stránky. Aby bylo zajištěno správné fungování zobrazení, jsou všechny relativní odkazy (v rámci serveru) přepsány na absolutní doplněním názvu. Před všechny odkazy v attributech **href** s **src**, které neobsahují absolutní cestu je doplněn název serveru a složky. Díky tomu se při kontrole načítá pouze samostatná stránka, ale při zobrazení se načtou i obrázky, styly a další odkazované elementy.

V případě, že už si uživatel danou změnu prohlédl, kliknutím na buňku v sloupci

Delete ji může z tabulky změn odstranit. Tímto odstraněním nedojde k smazání záznamu o změně ani odstranění souboru s danou verzí stránky. Změna se již nabude zobrazovat v tabulce změn. Jiným způsobem odstranění je vybrat jeden nebo několik řádků a pak kliknout na tlačítko **Hide** těsně pod tabulkou.

Vybráním řádků a kliknutím na tlačítko **Delete** dojde k smazání vybraných záznamů změn včetně souborů obsahujících dané verze stránek.

Všechny záznamy o změnách se ukládají do databáze a při spuštění aplikace se načtou všechny neskryté záznamy.

### 3.4 Historie změn a její procházení

Zobrazení již odstraněných záznamů z tabulky změn lze pomocí nástroje historie. Otevřít ho lze pomocí menu v hlavním okně kliknutím na **pages** → **history**. Po kliknutí se otevře další okno obsahující formulář na dotaz a pod ním prázdnou tabulku se stejnou strukturou jako původní tabulka změn.

Dotazy do historie se zadávají formou dotazu do databáze. Do políčka lze doplnit podmínky hledání a po kliknutí na tlačítko **Load** se načte tabulka odpovídajících záznamů. Tato načtení je jednorázové, tedy v případě, že se například vygeneruje další záznam, tabulka historie se sama neobnoví. Obnovení lze vynutit opětovným stiskem tlačítka **Load**. Jedinou výjimkou jsou změny provedené v daném okně historie – smazání a schování. Pod tabulkou historie jsou stejné ovládací prvky jako pod tabulkou změn. Tlačítko **Delete** smaže vybrané záznamy, ovšem tlačítko **Hide** funguje pouze na zatím neodstraněné záznamy z tabulky změn a to tak, že je schová v tabulce změn. V tabulce historie jsou schované záznamy zobrazeny červeně bez vyplněného sloupce **Delete**.

Do dotazů je možné psát libovolný SQL dotaz. Zdrojová tabulka by měla být **newPage** obsahující informace o všech změnách. Její struktura je:

- **id** INTEGER PRIMARY KEY AUTOINCREMENT – jednoznačný identifikátor záznamu
- **pageName** VARCHAR – identifikátor stránky (podle json zdrojového souboru)
- **time** DATETIME – čas zachycení dané změny
- **fileName** VARCHAR – jméno souboru obsahujícího danou změnu
- **del** bit – obsahuje 1 v případě, že stránka byla odstraněna z tabulky změn.

Dotaz by ovšem měl vracet tabulku o stejných sloupcích jako má **newPage**, aby mohl být správně graficky vykreslen. V opačném případě je chování aplikace nedefinované. Databáze je implementovaná pomocí **SQLITE**. Podrobná dokumentace je uveden na adrese [https://www.sqlite.org/lang\\_select.html](https://www.sqlite.org/lang_select.html).



## 3.5 Grafické porovnávání verzí stránek

V případě, že uživatel chce vidět rozdíly mezi dvěma verzemi stránek (popřípadě i mezi dvěma stránkami) je možné využít nástroje porovnání. Tento nástroj lze spustit kliknutím na tlačítko **Compare difference** pod tabulkou změn nebo pod tabulkou historie. Před kliknutím je potřeba vybrat jeden nebo dva řádky v příslušné tabulce. V případě, že jsou vybrány dva záznamy, provede se porovnání jejich souborů. V případě, že je vybrána pouze jedna stránka, provede se pouze porovnání mezi zvolenou verzí zvolené stránky a předchozí verzí dané stránky. Zobrazí se tedy rozdíly, které byly detekovány v daný moment.

Při použití porovnání se ve výchozím prohlížeči otevře stránka obsahující dva rámy obsahující obě porovnávané stránky. Dále se také otevře ovládací okno. V ovládacím okně je možno nastavit dva přepínače:

**View source code**, Při jeho zaškrtnutí se v prohlížeči zobrazí místo stránek jejich zdrojové kódy. Přepínač **Two page in one** místo dvou rámu zobrazí jen jeden, do kterého zobrazí sjednocení obou stránek. Tato funkce je zatím pouze v testovací verzi a může dojít k nesprávnému propojení stránek. Stabilní je pouze při propojení s zobrazením zdrojového kódu. Pro vytvoření grafického výstupu do prohlížeče dle zadaných kritérií je nutné zmáčknout tlačítko **Generate**. K znovuootevření stránky v prohlížeči slouží tlačítko **Open**.

V jakékoliv verzi porovnávání se graficky zobrazují změny. V režimu se dvěma rámy se v každém z nich zobrazí přebývajíc text s červeným zvýrazněním. V případě režimu jednoho rámu se přibitý text zobrazí se zeleným podbarvením a odstraněný text červeně. V režimu zdrojového kódu se změny zobrazí na čemkoliv – textu i html elementu či jiném prvku stránky. Když je ovšem porovnání formou stránek, jsou zvýrazněny pouze změny v čistém textu.

Porovnávání stránek se provádí na úrovni slov a tagů. Aplikace se snaží najít jejich zobrazení na sebe s největším překryvem, tedy nejmenším počtem rozdílů,

## **4 Implementace aplikace**

### **4.1 Použítý software**

Qt

Git

### **4.2 Objektový model, rozdělení problému**

### **4.3 Pozadí aplikace**

### **4.4 Grafické uživatelské rozhraní**