

Sem vložte zadání Vaší práce.





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Webová aplikace pro online web scraping**

*Jakub Drahoš*

Katedra softwarového inženýrství  
Vedoucí práce: Martin Podloucký

14. února 2019



---

## 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 14. února 2019

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2019 Jakub Drahoš. 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**

Drahoš, Jakub. *Webová aplikace pro online web scraping*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.



---

## Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

**Klíčová slova** Nahrad'te seznamem klíčových slov v češtině oddělených čárkou.

---

## Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

**Keywords** Nahrad'te seznamem klíčových slov v angličtině oddělených čárkou.



---

# Obsah

Úvod	1
1 Cíl práce	3
2 Analýza a návrh	5
2.1 Co je to vlastně ten web scraping? . . . . .	5
2.2 Analýza konkurence . . . . .	7
2.3 Specifikace požadavků . . . . .	16
3 Realizace	17
Závěr	19
Literatura	21
A Seznam použitých zkratk	23
B Obsah přiloženého CD	25



---

## Seznam obrázků

2.1	ParseHub . . . . .	8
2.2	Octoparse . . . . .	9
2.3	WebScraper . . . . .	11
2.4	Dexi.io . . . . .	13
2.5	Data Scraper . . . . .	15



---

# Úvod





## Cíl práce

Cílem této práce je navržení a tvorba webové aplikace, která bude umožňovat uživatelům vytáhnout požadovaná data z libovolné stránky v reálném čase bez jakékoli nutné znalosti programování.

Hlavním specifikem aplikace bude *přehlednost a jednoduchost uživatelského rozhraní* – je klíčové, aby bylo ovládání intuitivní, rychlé a jednoduché.

Naopak v rozsahu této práce není tvorba web crawlera ani žádného jiného podobného mechanismu, který by procházel danou oblast webu.



## Analýza a návrh

### 2.1 Co je to vlastně ten web scraping?

**Web sraping** (nebo také *web harvesting*, *web data extraction*) je technika získávání nejružnějších dat z webových stránek. Nejčastěji se v tomto kontextu jedná o automatizovaný proces strojového zpracování a získávání dat, nicméně může jít i o manuální extrakci zadanou uživatelem skrze nějaký software (jako je tomu právě v našem případě). [citace z Wiki - web scraping]

Často se také v souvislosti s pojmem web scraping používá spojení *web crawler* (nebo také *bot*, *spider*, *spiderbot*). Jedná se o automatizovaný software, který systematicky prochází danou oblast webu a během toho extrahuje kýžená data. Jak již bylo řečeno v úvodu, touto částí web scrapingu se práce nebude zabývat.

#### 2.1.1 Krátce k historii

Historie web scrapingu sahá k samým počátkům internetu (*World Wide Web*, 1989). Prvním webovým robotem, který byl vyvinut na MIT k měření velikosti webu, byl *World Wide Web Wanderer* (napsaný v jazyce *Perl*) z roku 1993. [citace z Wiki - World Wide Web Wanderer]

O něco později, v roce 2000, se ve velkém začala používat webová APIs – lidé mohli konzumovat čistá data a scraping se tak stal o hodně jednodušším.

Dalším milníkem v historii web scrapingu je rok 2004, kdy byla vydána knihovna pro parsování HTML a XML dokumentů *Beautiful Soup* pro programovací jazyk *Python*. Ta je do dnes považována za nejsofistikovanější a nejpokročilejší knihovnu pro web scraping.

Za zmínku stojí určitě i rok 2006, kdy je datován příchod vizuálního web scrapingu, tedy techniky, kdy uživatel skrze rozhraní nějaké aplikace označí klikáním myši, z kterých oblastí webové stránky chce vytáhnout data. Tímto se otevřely dveře web scrapingu pro všechny. [citace z <https://www.octoparse.com/blog/web-scraping-introduction>]

### 2.1.2 Techniky

Technik, jak z webové stránky vytáhnout data existuje mnoho, podívejme se alespoň na některé z nich:

- Vyhledávání na základě textové shody – např. pomocí UNIX nástroje *grep* nebo regulárních výrazů
- HTML parsování – základní a stále ještě nejpoužívanější technika extrakce dat. Informace jednoduše získáváme z HTML elementů, popř. pomocí tříd nebo id
- Počítačové vidění, strojové učení, zapojení umělé inteligence – snaha napodobit způsob, jakým vidí a zpracovává webovou stránku člověk, něco podobného zkouší např. projekt *Diffbot*
- Vizuální web scraping – jak již bylo zmíněno výše, požadovaná data se musí ručně naklikat skrze rozhraní nějaké aplikace (značně to však usnadňuje např. inteligentní hledání podobných prvků na základě prvních pár kliknutí)
- Ruční vyhledávání a stahování dat – může se ukázat, že někdy je to tou nejsnazší a nejrychlejší alternativou

### 2.1.3 Využití web scrapingu

Podob pro uplatnění scrapování dat z webu je nespočet, a to obzvlášť v dnešní době, kdy jsme přímo zaplaveni daty (pohybujeme se v řádech Zettabajtů –  $1024^7$  B [citace z <https://www.nodegraph.se/big-data-facts/>]). Mezi ty hlavní patří:

- Získání kontaktních informací (např. e-mail) pro marketingové účely
- Indexování webových stránek (jako příklad můžeme uvést *GoogleBot*)
- Data mining – proces hledání vzorců ve velkých datových setech [odkaz na Wiki]
- Monitorování různých proměnných (např. sledování cen nebo hodnocení produktů)
- Recyklace již někdy použitých dat za účelem vytváření „nového“ obsahu
- Analýza a zpracování dat k výzkumným účelům

### 2.1.4 Je to vlastně legální?

To be done...

## 2.2 Analýza konkurence

První skupinou, na kterou můžeme při hledání na internetu narazit, jsou společnosti, které nabízejí zákazníkům kompletní péči v rámci extrakce dat. Cílí především na velké korporace, jimž postaví scrapovací nástroj přesně na míru, který poté také hostují a spravují. Zákazník tedy dostane data a o nic víc se již nemusí starat. Jako příklad můžeme jmenovat třeba *ContentGrabber*, *Mozenda* a další.

Pro nás mnohem relevantnější kategorií je konkurenční nabídka nástrojů poskytujících uživatelům rozhraní k web scrapingu. Budeme se zaměřovat pouze na takové nástroje, které nevyžadují jakoukoli znalost programování – tedy žádné knihovny, API a nástroje pro budování vlastních scraperů.

Mezi ty největší představitele patří *ParseHub*, *Octoparse*, *WebScaper*, *Data Scraper* a *Dexi.io*. Čtyři z nich jsou volně dostupné nástroje (které mají ale velmi omezenou funkcionalitu a pokročilejší operace se odemknou až s určitým platebním plánem – tzv. *freemium model*) a jeden poskytuje bezplatně pouze 7 denní zkušební verzi.

Předtím, než začneme jednotlivé nástroje porovnávat, musíme si určit nějaká kritéria, podle kterých budeme hodnotit kvalitu daného nástroje. Především nám půjde o jednoduchost používání, celkovou přehlednost a rychlost, se kterou se uživatel dostane k požadovaným datům. Také nás bude zajímat způsob výběru dat, možnosti exportu získaných dat, jak aplikace sama dokáže uživatele seznámit s používáním a také, v jaké formě se nástroj vůbec používá a jestli něčím vybočuje (ať už v pozitivním či negativním smyslu).

Pojďme se tedy na některé nástroje podívat blíže:

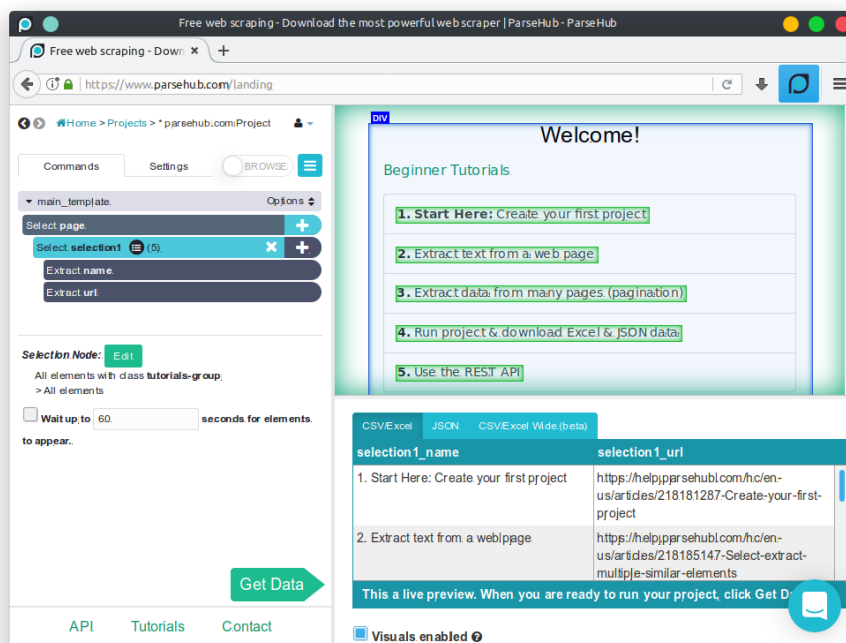
### 2.2.1 ParseHub

#### Výhody:

- Výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí *XPath*, regulárních výrazů nebo CSS selektorů
- Aplikace obsahuje interaktivní tutoriál, který na jednoduchých příkladech ukáže, jak s nástrojem zacházet
- Možnost získání dat různými formami - přes API, jako CSV/XLS, do *GoogleSheets* nebo do *Tableau*
- Různé módy kliknutí (výběr, relativní výběr, kliknutí), zooming in/out na HTML elementy – když se uživatel netrefí (nebo ani trefit nemůže) přesně na požadovaný element, lze na něj lehce přejít pomocí této funkce
- Automatická rotace IP adresy (tedy nedochází k blokování ze strany serveru)

## 2. ANALÝZA A NÁVRH

---

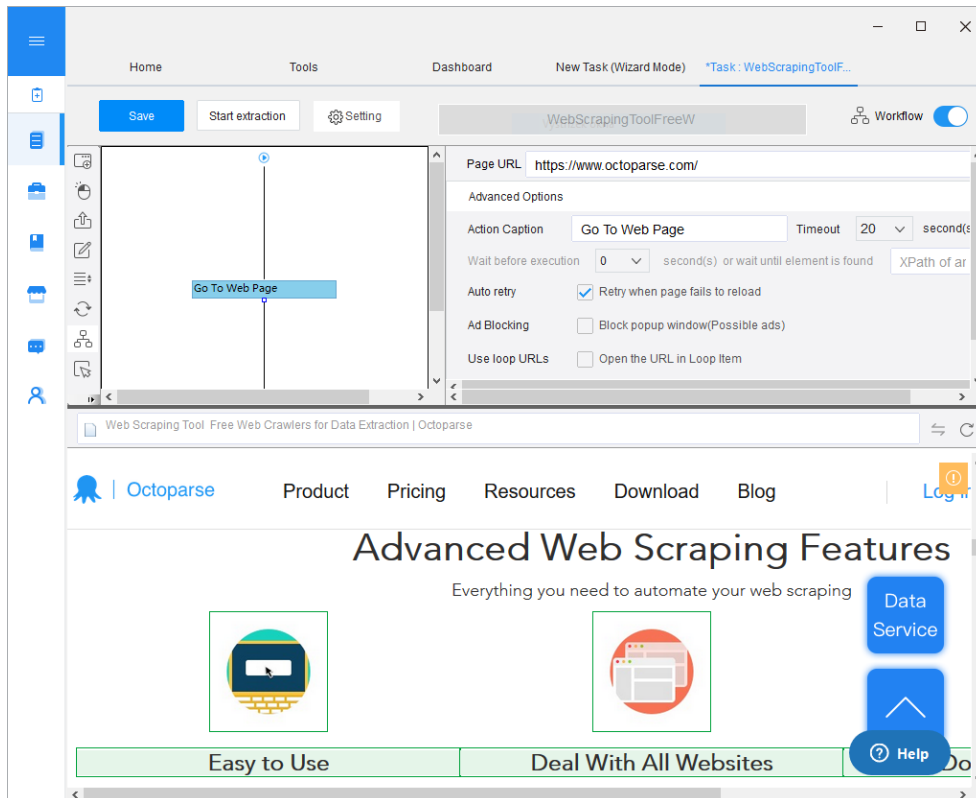


Obrázek 2.1: ParseHub

### Nevýhody:

- Nutnost stažení aplikace (ale je zde podpora pro *Windows*, *Linux* i *Mac*)
- Aplikace je celkově těžkopádná, nemá moc přívětivé uživatelské rozhraní, ovládání působí nepřehledně a přehlčeně – na uživatele se vyvalí hodně informací a možností najednou

### 2.2.2 Octoparse



Obrázek 2.2: Octoparse

#### Výhody:

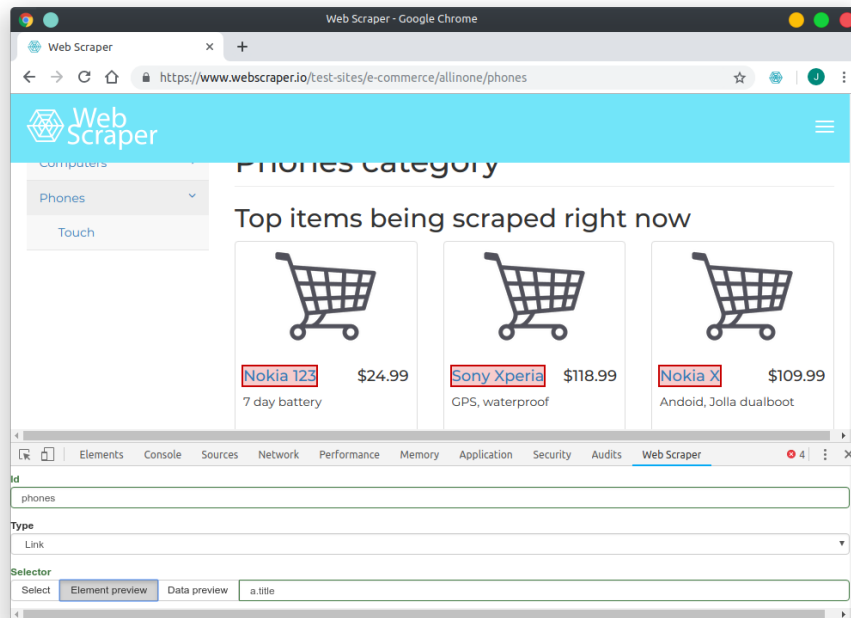
- Výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí *XPath* nebo regulárních výrazů
- Nástroj obsahuje předpřipravené šablony, které mohou velmi urychlit práci
- Pestrá paleta možností (branch judgment, tvoření smyček apod.) – dá se vytvořit téměř jakákoli logika procházení webu a extrakce dat
- Lehký způsob, jak scrapování automatizovat
- Možnost řídit tasky přes API (a získávat tak data taktéž přes API). Data jdou nahrát rovnou i do lokální databáze

### **Nevýhody:**

- Nutnost stažení aplikace (která je navíc pouze pro *Windows*)
- Těžkopádné a pomalé ovládání, neintuitivní rozhraní
- Tutoriál je v podstatě nic neříkající
- Předpřipravených šablon je jenom pár a jsou velmi konkrétní



### 2.2.3 WebScaper



Obrázek 2.3: WebScaper

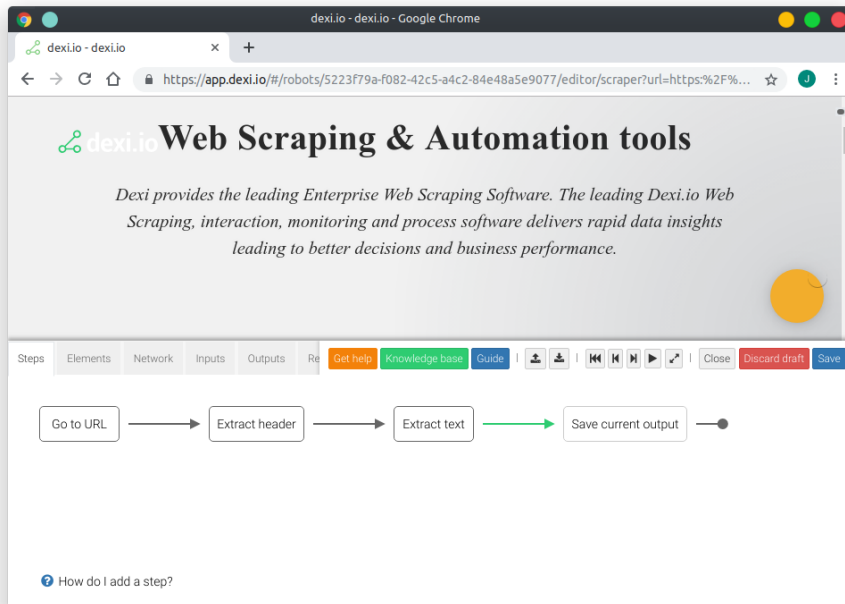
#### Výhody:

- Jednoduchá instalace (jedná se pouze o rozšíření do prohlížeče *Google Chrome*). Nastavování probíhá skrze vývojářskou konzoli
- Výběr dat pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí)
- Tutoriály jsou formou videí – jednoduché, rychlé a naprosto postačující
- Různé typy elementů, které vybíráme (text, odkaz, scroll down), takže lze celkem snadno prolézt celou stránku
- Možnost získání dat různými formami - přes API, jako CSV/XLS nebo do *Dropboxu*
- Klávesové zkratky při výběru elementů velmi usnadňují práci
- Možnost využít jejich cloud k automatizaci celého procesu
- Oproti konkurenci nabízí přehledné rozhraní, rychlé a jednoduché používání

### **Nevýhody:**

- Nutnost používat *Google Chrome*, což pro některé uživatele může být překážka
- Nelze vyhledávat podle klíčových slov ani podle HTML nebo CSS, tudíž všechno se musí poctivě naklikat

### 2.2.4 Dexi.io



Obrázek 2.4: Dexi.io

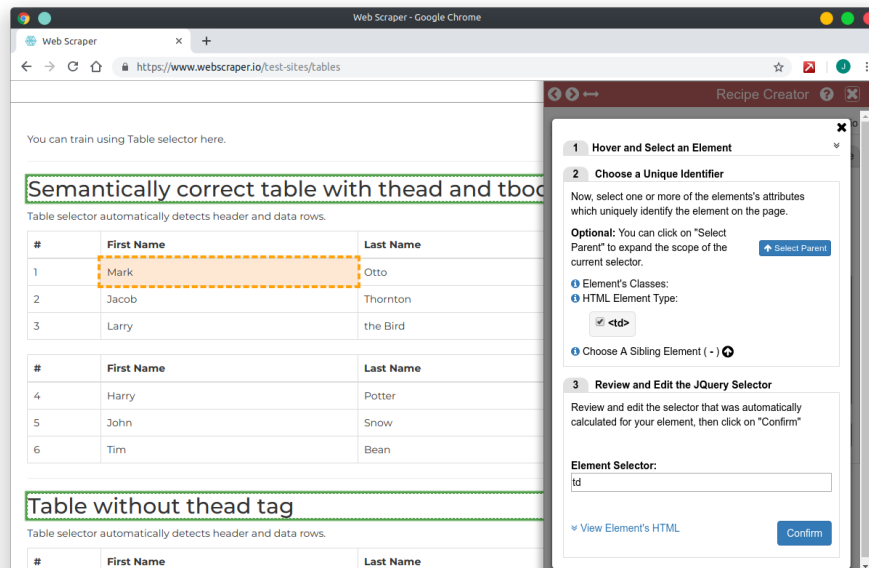
#### Výhody:

- Bez nutnosti stahování aplikace – vše se ovládá přes webové rozhraní
- Výběr dat jak pomocí klikání (inteligentní hledání vzorců/podobností na základě prvních dvou kliknutí), tak pomocí HTML, CSS nebo textové shody
- Hodně návodů dostupných na stránkách, interaktivní rádce přímo při scrapování
- Všechny možné druhy kliknutí, takže lze lehce prolézt celou stránku
- Možnost exportovat data do CSV, JSON, Excel, získat přes API, poslat do *Google Drive*, *Google Sheets* nebo *Amazon S3*
- Různé módy aplikace – scraping, crawler, pipes (skládání menších scrape botů) a autobot (extrahování z více stránek najednou se stejným rozložením). Možnost takto automatizovat celý proces.
- Skvělé jsou různé addony (např. na obcházení *Captcha*)

### **Nevýhody:**

- Široká nabídka možností a tak chvíli trvá, než se člověk zorientuje
- Placený nástroj, zadarmo je dostupná pouze týdenní zkušební verze
- Úvodní tutoriál je velmi strohý a žádné velké seznámení s nástrojem se nekoná

### 2.2.5 Data Scraper



Obrázek 2.5: Data Scraper

#### Výhody:

- Jednoduchá instalace (jedná se pouze o rozšíření do prohlížeče *Google Chrome*).
- Velmi jednoduché ovládání a přehledné rozhraní
- Výběr dat probíhá pomocí klikání. Skvělé je, že klikáním se utváří *jQuery* selektor, který si uživatel může podle svého upravit a doladit tak drobné detaily, které by jinak nutně zahltily uživatelské rozhraní. Tedy je možné vyhledávat i podle HTML tagů, id, CSS selektorů – zkrátka vše, co umí klasické *jQuery*
- Různé druhy kliknutí, možnost spustit na stránce libovolný *JavaScriptový* kód v rámci scrapování

#### Nevýhody:

- Nutnost používat *Google Chrome*, což pro některé uživatele může být překážka
- Oproti ostatním nástrojům se může zdát velmi chudý na různé funkce

### 2.3 Specifikace požadavků

Jak jsme viděli v předchozí analýze konkurenčních nástrojů, největšími neduhy, které se prolínají napříč valnou většinou aplikací, jsou **těžkopádné uživatelské rozhraní, neintuitivní ovládání a rychlost** (nebo spíš pomalost), se kterou se uživatel dostane k požadovaným datům. Pro aplikaci, jíž se tato práce zabývá, bude klíčové se těmito nedostatky vyhnout a nabídnout jejich přesný opak. Také jsme se přesvědčili, že nejpříjemnější cestou je celou aplikaci ovládat přes webové rozhraní a **nemuset nic stahovat a instalovat**.

Na druhou stranu se můžeme u konkurence i inspirovat. Za vyzdvižení stojí určitě **různé druhy výběru dat – klikání** přímo na stránce spolu s inteligentním hledáním podobných prvků jistě tvoří mocný mechanismus. Avšak je potřeba zajistit i ostatní způsoby výběru (jako je např. **textová shoda, HTML tagy, CSS selektory**) pro případ, kdy je pouhé klikání zdoluhavé či nevyhovující. Rovněž široký výběr způsobů exportu dat, intuitivní klávesové zkratky a zooming in/out na prvky může uživatelům zpříjemnit práci s nástrojem.

Neméně důležitou vlastností aplikace je také schopnost sebe sama kvalitně, ale svižně představit, **seznámit uživatele s používáním** a poskytnout mu alespoň pro začátek nějaké vodítko. Pro většinu ovládacích prvků by ale mělo platit to stejné, co platí pro správný kód – měly by být tzv. **self-explanatory**. Tedy každému by mělo být na první pohled jasné, co který element dělá.

Pojďme si tedy rozdělit požadavky na funkční a nefunkční tak, jak by to každý správný softwarový projekt měl udělat:

#### 2.3.1 Funkční požadavky

- 

#### 2.3.2 Nefunkční požadavky

- Aplikace bude primárně cílit na celkový zážitek uživatele – grafické rozhraní bude přehledné a co nejjednodušší, ovládání intuitivní
- Čas, za který se uživatel dostane k požadovaným datům (tedy čas, který stráví vybíráním dat), bude co nejmenší

## **Realizace**





---

## **Závěr**



---

## Literatura



## Seznam použitých zkratk

**HTML** HyperText Markup Language

**XML** eXtensible Markup Language

**API** Application Programming Interface

**MIT** Massachusetts Institute of Technology

**CSS** Cascading Style Sheets

**CSV** Comma-Separated Values

**IP** Internet Protocol

**JSON** JavaScript Object Notation

**XLS** formát souboru používaný aplikací Microsoft Excel



## 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 L <sup>A</sup> T <sub>E</sub> X
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS