Masarykova univerzita
Fakulta informatiky

Optimalizace publikace datových sad na portále Open Data

Diplomová práca

Bc. Tomáš Hrnčiar

Brno, jar 2020

Masarykova univerzita
Fakulta informatiky

Optimalizace publikace datových sad na portále Open Data

Diplomová práca

Bc. Tomáš Hrnčiar

Brno, jar 2020

MUNI
FI

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY
BOTANICKÁ 68A, 602 00 BRNO
IČ: 00216224
DIČ: CZ00216224

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Tomáš Hrnčiar
Program:	Aplikovaná informatika
Obor:	Služby - výzkum, řízení a inovace
Garant oboru:	doc. RNDr. Vlastislav Dohnal, Ph.D. (SSME)
Vedoucí práce:	Ing. Leonard Walletzký, Ph.D.
Katedra:	Katedra počítačových systémů a komunikací
Název práce:	Optimalizace publikace datových sad na portále Open Data
Název práce anglicky:	Optimalization of data sets publication on Open Data portal
Zadání:	Student představí problematiku datového portálu pro publikaci dat provozu města. Následně provede analýzu na konkrétní implementaci datového portálu se zřetelem na proces tvorby, aktualizace, publikace a archivace datových sad. Na vybraném vzorku minimálně 3 datových sad provede návrh automatizovaného nebo poloautomatizovaného procesu tak, aby celý přenos dat od zdroje (většinou služba/aplikace třetí strany) až do portálu otevřených dat byl zajištěn automaticky. V závěru práce provede vyhodnocení procesu z hlediska ušetřeného času a nákladů.
Literatura:	CAPUTO, Francesco, Mouzhi GE a Leonard WALLETZKÝ. *Modelling the Service Value Chain for Smart City*. In *Proceedings of the 25th Interdisciplinary Information Management Talks*. Poděbrady, Czech Republic: Trauner Verlag, 2017. s. 375-382, 8 s. ISBN 978-3-99062-119-6. WALLETZKÝ, Leonard, Barbora BÜHNOVÁ a Luca CARRUBBO. *Value-Driven Conceptualization of Services in the Smart City: A Layered Approach*. In *Social Dynamics in a Systems Perspective*. Cham, Germany: Springer, 2018. s. 85-98, 14 s. ISBN 978-3-319-61966-8. doi:10.1007/978-3-319-61967-5_5.

Zadání bylo schváleno prostřednictvím IS MU.

Vyhlásenie

Vyhlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Bc. Tomáš Hrnčiar

Vedúci práce: Ing. Leonard Walletzký, Ph.D.

Poďakovanie

Rád by som sa poďakoval vedúcemu tejto práce pánovi Ing. Leonardovi Walletzkému, Ph.D. za všetky potrebné informácie, ktoré mi dopomohli k jej napísaniu, za odbornú komunikáciu, jeho pripomienky a rady.

Ďakujem tiež zamestnancom mesta Žďár nad Sázavou, najmä Ing. Michalovi Bačovskému za skvelú komunikáciu a pomoc so získavaním dátových sád mesta.

Ďalej chcem poďakovať rodine, priateľke a kolegom z Laboratória servisných systémov FI MUNI za pomoc pri dokončovaní práce.

Zhrnutie

Otvorené dáta, ktoré vznikajú v rámci miest, sú majetkom občanov a preto je žiaduce, aby boli dostupné pre verejnosť. Tento trend je podporovaný na úrovni štátu aj Európskej Únie. Avšak pri zverejňovaní dát, ktoré sú určené na publikovanie, je problémom ich fragmentácia. Rôzne dáta spadajú pod rôzne oddelenia a zamestnancov. Z toho vyplýva závislosť od jednotlivcov a teda aj chybovosť, nedôslednosť, prípadne neaktuálnosť. Zamestnanci častokrát nemajú potrebné technické skúsenosti a je to niečo nad rámec ich bežnej pracovnej činnosti. Cieľom práce bolo automatizovať proces zverejňovania dátových sád.

Automatizácia bola riešená pomocou skriptov, ktoré zabezpečujú celý proces zverejňovania dát. Vytvorenie rôznych algoritmov bolo navrhnuté pre použiteľnosť v rôznych dátových sadách. Získané dáta sú spracovávané tak, aby sa z nich odstránili všetky citlivé dáta, ktoré by mohli brániť ich publikovaniu. Navyše sú dáta upravované podľa odporúčaní Ministerstva vnútra Českej republiky. Boli vytvorené skripty, ktoré sa podľa nastavených pravidiel spúšťajú automaticky na serveri a zabezpečujú vkladanie a aktualizovanie do portálu CKAN.

V spolupráci s mestom Žďár nad Sázavou sa podarilo zautomatizovať zverejňovanie troch vybraných dátových sád vrátane zverejnenia dostupných dát z minulosti. Záverom bol vyhodnotený rozdiel nákladov pri automatickom zverejňovaní dát oproti manuálnemu nahrávaniu.

Kľúčové slová

Smart City, otvorené dáta, verejná správa, CKAN, automatická aktualizácia dát

Obsah

Úvod **1**

1 Smart Cities a otvorené dáta **3**
1.1 *Smart Cities* . . . 3
1.2 *Smart Cities v Českej republike* . . . 5
1.3 *Otvorené dáta v Českej republike* . . . 10
1.3.1 Kvalita otvorených dát . . . 12
1.3.2 Otvorené formálne normy . . . 14
1.4 *Žďár nad Sázavou* . . . 15

2 Proces publikácie otvorených dát **18**
2.1 *Digitalizácia dát* . . . 18
2.2 *Získanie dát* . . . 19
2.3 *Odstránenie citlivých dát* . . . 20
2.4 *Zverejnenie dát* . . . 20
2.5 *Udržiavanie dát* . . . 21

3 Použité technológie **22**
3.1 *Comprehensive Knowledge Archive Network (CKAN)* . . . 22
3.1.1 Organizácia dát . . . 23
3.1.2 CKAN API . . . 24
3.2 *CRON* . . . 25

4 Automatizovaná publikácia dátových sád **26**
4.1 *Dátová sada o využívaní elektronabíjačiek* . . . 26
4.1.1 Zdroj dát . . . 26
4.1.2 Formát očakávaných dát na vstupe . . . 27
4.1.3 Implementácia . . . 27
4.2 *Dátová sada o teplote na mestskom úrade* . . . 35
4.2.1 Zdroj dát . . . 35
4.2.2 Formát očakávaných dát na vstupe . . . 36
4.2.3 Implementácia . . . 36
4.3 *Dátová sada o výsledkoch hlasovania mestského zastupiteľstva* 42
4.3.1 Zdroj dát . . . 42
4.3.2 Formát očakávaných dát na vstupe . . . 42
4.3.3 Implementácia . . . 42

4.4 *Skript pre plánované spúšťanie importu dát* 45

5 Vyhodnotenie prínosov pre mesto **47**
5.1 *Odhad nákladov pre manuálne nahrávanie dát* 47
5.2 *Odhad nákladov pre automatické nahrávanie dát* 49

6 Záver **51**

A Elektronické prílohy **53**

B Konfiguračný súbor pre elektronabijecky.py **54**

C Vzor vstupných dát pre skript elektronabijecky.py **55**

D Konfiguračný súbor pre teplota.py **57**

E Vzor vstupných dát pre skript teplota.py **58**

F Konfiguračný súbor pre uredni-deska.py **59**

G Vzor vstupných dát pre skript uredni-deska.py **60**

Bibliografia **61**

Zoznam tabuliek

4.1 Argumenty pre ovládanie skriptu elektronabijecky.py *30*
4.2 Identifikačné čísla pre nabíjacie stanice a zásuvky *32*
4.3 Návratové hodnoty skriptu elektronabijecky.py *34*
4.4 Argumenty pre ovládanie skriptu teplota.py *38*
4.5 Návratové hodnoty skriptu teplota.py *41*
4.6 Argumenty pre ovládanie skriptu uredni-deska.py *44*
4.7 Návratové hodnoty skriptu uredni-deska.py *45*
5.1 Výpočet času potrebného pre počiatočný manuálny import dát; X: odhadovaný čas pre import jednej dátovej sady (hod), Y: počet existujúcich dátových sád do júna 2020 *48*
5.2 Výpočet nákladov potrebných pre počiatočný manuálny import dát; X: odhadovaný čas pre import jednej dátovej sady (hod), Y: počet existujúcich dátových sád do júna 2020 *49*
5.3 Výpočet nákladov tvorbu automatických skriptov pre import dát *50*

Zoznam obrázkov

1.1 Predikcia vývoja zloženia miest vo svete [8] *5*
1.2 4 úrovne Smart Cities a ich 16 komponentov [11] *9*
1.3 Stupne otvorených dát [19] *14*
1.4 Logo Žďáru nad Sázavou pre agendu Smart City [22] *17*
3.1 Štruktúra dát v systéme CKAN. *24*

Úvod

S rozvojom informačných technológií sa začína aj proces digitalizácie verejnej správy. Už v dnešnej dobe fungujú prvé služby, kde si občania môžu vybaviť základné požiadavky pomocou internetu bez nutnosti návštevy úradov. Prínosy digitalizácie sú jednoznačné. Uľahčenie života občanom, menšia byrokracia, ale aj odľahčenie štátnej správy. Tento trend má podporu aj na úrovni Európskej únie. Európsky parlament a Rada EÚ schválili 20. júna 2019 smernicu 2019/1024 [1] o otvorených dátach a opakovanom použití informácií verejného sektora. Úlohou tejto smernice je podpora používania otvorených dát. Stanovuje súbor minimálnych pravidiel a praktické opatrenia na uľahčenie opakovaného použitia.

Táto diplomová práca sa zaoberá riešením otázky publikovania dát a súčasne zhodnotením prínosu automatického zverejňovania oproti manuálnemu. Po jej prečítaní sa čitateľ dozvie, čo je to Smart City, ako vyzerá proces zverejňovania dát a aké dátové sady sa podarilo, alebo nepodarilo zverejniť. Nadobudnuté poznatky budú prakticky aplikované na príklade mesta strednej veľkosti - Žďáru nad Sázavou. Vďaka tejto spolupráci sa podarilo nastaviť procesy, ktoré zabezpečia automatizované zverejňovanie troch dátových sád na portále otvorených dát[1] postaveného na platforme Comprehensive Knowledge Archive Network (CKAN). Ide o dáta o využívaní elektronabíjacích staníc, o meraní teploty na Mestskom úrade v Žďári nad Sázavou a o hlasovaní mestského zastupiteľstva. Spomenieme aj dôvody, kvôli ktorým sa nepodarilo zverejniť iné dátové sady, čo môže pomôcť ďalším mestám, ktoré sa rozhodnú začať zverejňovať otvorené dáta.

V prvej kapitole bude popísané čo to je Smart City a aký je aktuálny stav Smart City v Českej republike. Budeme sa zaoberať aj problematikou otvorených dát a ako sa definuje ich kvalita. Zadefinujeme, čo sú otvorené formálne normy a načo sa používajú. V závere kapitoly sa nachádza popis situácie v Žďári nad Sázavou v kontexte Smart City. Zameriame sa na to, čo sa tam už podarilo dosiahnuť a aké sú plány do budúcnosti.

1. `http://sc02.fi.muni.cz`

Potom bude nasledovať kapitola, v ktorej sa budeme zaoberať opatreniami - aké opatrenia musia nastať, aby sa z dát stali otvorené dáta. Popíšeme situácie a problémy, s ktorými sme sa stretli, keď sme pracovali na nastavení procesu ich pravidelného automatického zverejňovania. Vzhľadom k tomu, že každá dátová sada má inú východiskovú pozíciu, iné sú aj podmienky, ktoré sa musia naplniť, aby ju bolo možné zverejniť. Niektoré dáta sú na webe už v dnešnej dobe dostupné v podobe na zverejnenie. Stačí ich iba vziať a mierne upraviť. Opačným extrémom sú dáta, ktoré sa spracovávajú ručne a ešte vôbec neboli zdigitalizované, takže potrebujú omnoho väčšiu pozornosť pracovníkov mesta.

V nasledujúcej kapitole budú popísané technológie, ktoré sme museli použiť na implementáciu riešenia. Budeme sa zaoberať portálom CKAN a aplikáciou CRON pre pravidelné spúšťanie skriptov na servery.

Samotná implementácia riešenia bude popísaná v kapitole č. 4. Najskôr bude odprezentovaný všeobecný algoritmus spoločný pre všetky nami vytvorené skripty, kde sa budeme zaoberať najmä všeobecnými funkciami. Tie sa dajú použiť naprieč rôznymi dátovými sadami (jedná sa najmä o kód zabezpečujúci nahrávanie dát do portálu CKAN). Neskôr budú tiež popísané špecifiká jednotlivých dátových sád, ktoré sme museli vykonať, aby sa mohli dáta zverejniť.

V poslednej kapitole prebehne zhodnotenie výsledku práce. Čitateľ sa dozvie o úsporách, ktoré táto práca mestu Žďár nad Sázavou priniesla - či už ide o časovú, alebo finančnú. Záverom zhodnotíme víziu do budúcnosti.

1 Smart Cities a otvorené dáta

1.1 Smart Cities

Smart City (vo voľnom preklade inteligentné mesto) je nový štýl riadenia mesta, ktorý je založený na informáciách, informačno-komunikačných technológiách a ostatných formách fyzického kapitálu. Existuje predpoklad, že s inteligentným manažmentom zdrojov bude možné zvýšiť kvalitu života v mestách, znížiť množstvo odpadu a zlepšiť ekonomické podmienky. [2].

V dnešnej dobe sa čím ďalej, tým viac začína riešiť otázka nasadenia Smart Cities v obciach aj mestách. Súvisí to najmä s aktivitami v rámci Európskej Únie, ktorá podporuje zdieľanie a zverejňovanie otvorených dát. Vychádza z toho, že verejný sektor produkuje obrovské množstvá dát, ktoré sú platené z peňazí daňových poplatníkov a preto by mali byť zverejňované. Politika Európskej komisie sa zameriava na zvyšovanie hodnoty pre ekonomiku a spoločnosť pomocou znovupoužiteľnosti týchto dát [3]. To môže byť dosiahnuté, len ak sú dáta štandardizované a zastrešené jedným portálom. Centrálne miesto pre Európsku Úniu je EU Open Data Portal [4]. Jeho úlohou je centralizovať a zastrešovať dáta z národných katalógov jednotlivých členských štátov Európskej Únie.

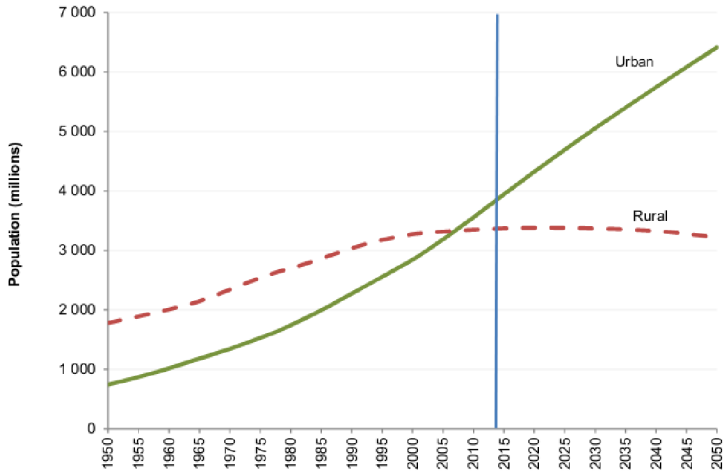
Ďalším z dôvodov podpory Smart Cities zo strany Európskej Únie sú predpovede, podľa ktorých budú do roku 2050 dve tretiny populácie žiť v mestách. Budú spotrebovávať 70% celkovej energie a produkovať rovnaký podiel skleníkových plynov. Ako budú mestské populácie narastať, požiadavky na služby budú vytvárať aj väčší tlak na zdroje. To spôsobí napätie v oblastiach ako je energetika, voda, odpady, mobilita a ďalšie služby nevyhnutné pre prosperitu a udržateľnosť mesta. [5].

Tu prichádza do popredia mesto riadené inteligentnou koncepciou. Je to mesto, kde sú tradičné siete a služby efektívnejšie používané s využitím digitálnych a telekomunikačných technológií. Z toho plynú benefity pre obyvateľstvo a podnikateľské prostredie. Takéto mesto ide nad rámec bežného využívania informačno-komunikačných technológií (IKT), lepšie využíva zdroje, informuje a vytvára menej emisií. Usiluje sa o udržateľnosť pomocou inteligentných transportných sietí, vylepšeným vodným zásobovaním, čističkami vôd a efektívnejšími

spôsobmi vykurovania budov a osvetlenia. Znamená to tiež interaktívnejšiu a flexibilnejšiu mestskú administratívu, bezpečnejšie verejné miesta a napĺňanie potrieb starnúcej populácie [6]. Cieľom Európskej Únie je dosiahnuť uhlíkovú neutralitu, čiže rovnováhu medzi množstvom vyprodukovaného uhlíka a množstvom uhlíka pohlteného z atmosféry, do roku 2050 [7]. Toto nebude možné, pokiaľ sa na riadení miest niečo nezmení. Mesto riadené formou Smart City prináša práve takúto potrebnú zmenu.

Hlavný prínos je v tom, že vychádza z konceptu rozhodovania sa na základe dát a nie pocitov, ako to dnes často funguje. Toho sa dá dosiahnuť jedine tým, ak má mesto poriadok vo svojich dátach a sú verejne dostupné. Mesto vie, aké dáta zbiera, ako často sa aktualizujú a vie ich aj správne vyhodnotiť tak, aby na ich základe dokázalo robiť správne rozhodnutia. Dáta sú v anonymizovanej podobe zverejnené na internete, aby si ich mohla verejnosť prezerať a využívať ich a týmto sa podieľať na kontrole činnosti mesta. So zverejnenými dátami sa dá ďalej pracovať a vytvárať aplikácie, ktoré uľahčujú život ostatným občanom. Typicky sa toto zverejnenie deje na verejne prístupnom portále otvorených dát, kde sa nachádzajú prehľadne usporiadané všetky dáta, ktoré mesto spravuje. Tieto dáta sú pravidelne aktualizované podľa životného cyklu, ktorý je definovaný pre každú dátovú sadu. Predpokladá sa, že ich budú okrem miest využívať taktiež firmy, vzdelávacie inštitúcie, ale aj občania. Firmy si na nich môžu postaviť podnikanie, z ktorého budú profitovať občania. Stredné a vysoké školy zase môžu otvorené dáta využiť vo vzdelávaní pri práci na vzdelávacích projektoch a záverečných prácach. Vďaka tomu získajú študenti praktické skúsenosti a v konečnom dôsledku z toho bude profitovať ekonomika aj zamestnanosť. Dôležitosť Smart Cities podčiarkuje aj predikcia vývoja, kde je zrejmé, že obyvateľstvo sa bude sústreďovať najmä v mestských oblastiach viď. Obr. 1.1.

Koncept inteligentného mesta teda pokrýva rôzne oblasti riadenia mesta. Ide najmä o energetiku, financovanie, eGovernance, dopravu a mobilitu, informačné a komunikačné technológie, vzdelávanie, sociálne služby a zdravotníctvo, vodné a odpadové/obehové hospodárstvo.



Obr. 1.1: Predikcia vývoja zloženia miest vo svete [8]

1.2 Smart Cities v Českej republike

Myšlienky o začatí aplikovania konceptu Smart City v českých mestách sa prvýkrát objavili na prelome rokov 2007 a 2008, v súvislosti s klimaticko-ekonomickými cieľmi Európskej komisie a zapojením cekovo jedenástich miest do Paktu starostov a primátorov [1]. Cieľom bolo pripojenie sa k celoeurópskym aktivitám na zvýšenie energetických úspor a zníženie vplyvu mestských aglomerácií na klímu. Medzi signatárov tohto paktu patrili do januára 2019 zástupcovia miest Brno, Chrudim, Hlinsko, Jeseník, Liberec, Litoměřice, Lkáň, Mezilesí, Ostrava, Praha a Písek [9]. Gestorom pre Českú republiku v oblasti otvorených dát je Ministerstvo vnútra ČR, ktorého úlohou je implementácia smerníc schválených v Európskom parlamente a Radou EÚ do českej legislatívy.

Česká republika má 3. najvyšší počet obcí a miest v rámci Európskej Únie. Údaje Českého štatistického úradu ukazujú, že v roku 2018 sa v Českej republike nachádzalo celkovo 6 258 obcí, pričom väčšina z nich má menej ako 1000 obyvateľov. V mestách a aglomeráciách veľkých miest žijú dve tretiny všetkých obyvateľov. Zväz miest a obcí

1. Ministerstvo životního prostředí (2019). Pakt starostů a primátorů pro udržitelnou energii a klima. Dostupné on-line: `https://www.mzp.cz/cz/pakt_starostu_a_primatoru`

Českej republiky prišiel s iniciatívou SMART Česko. Tá prináša vízie ako by mohla krajina vyzerať v nasledujúcich rokoch [10].

Rok 2035: SMART Česko je zem využívajúca svoj potenciál naplno, ten je daný vysoko kvalifikovanými ľuďmi, mimoriadnou infraštruktúrou pre výzkum, produktívnou ekonomikou založenou na dlhej priemyslovej tradícii, v európskom kontexte mimoriadne zachovanou biodiverzitou a relatívne málo poškodenou prírodou. SMART Česko je spoločnosť, kde majú ľudia vzťah k svojmu domovu a zvládajú sa veľmi dynamicky prispôsobovať novým technickým riešeniam [10, str. 17].

Rok 2050: Udržateľné Česko je kultivovaná a prosperujúca európska zem sebavedomých a vzdelaných ľudí, ktorí vytvárajú otvorené a funkčné komunity v prirodzene rastúcich obciach, ktoré (ak je to možné) nevytvárajú akýkoľvek typ záťaže (dlhu) pre budúce generácie. Rozvíja sa v troch základných smeroch udržateľného rozvoja k hlavným dlhodobým strategickým cieľom - kultivovaná spoločnosť ľudí, ekonomika založená na znalostiach a uhlíkovo neutrálna zem [10, str. 16].

Na tieto vízie nadväzuje globálny cieľ stratégie **Obec krátkych vzdialeností - Česko ako internet**, ktorý bližšie dokresľuje situáciu v Českej republike. Množstvo malých obcí autori v dokumente prirovnávajú sieti, akú tvoria jednotlivé počítače komunikujúce v internetovej sieti. Chcú dosiahnuť, aby obce vo väčšej miere medzi sebou spolupracovali a aby boli čo najviac využívané moderné technológie, ktoré poskytuje dnešná doba. Absolútne esenciálne je zabezpečiť dostatočne rýchle pripojenie a doručovacie služby do miest, kde žijú. To umožní ľuďom pracovať z domu a vďaka tomu sa znížia náklady na dopravu. Prebehne aj zmena životnej úrovne, zníži sa záťaž na životné prostredie a zvýši sa možnosť uplatnenia v mieste bydliska prípadne v blízkom okolí [10, str. 19].

Metodika Konceptu inteligentných miest, ktorá vyšla v roku 2015, bola prvým dokumentom zaoberajúcim sa oblasťou Smart City. Zavádza rámec, ktorý predstavuje kontrolné položky, ktoré musia byť dosiahnuté, aby postupné nasadzovanie rôznych moderných technológií vyvolalo synergický efekt. V roku 2018 vyšla jej aktualizovaná verzia pod názvom Metodika Smart Cities [11].

Rámec inteligentného mesta [11] pozostáva z 16 hierarchicky usporiadaných komponentov, ktoré sa dajú rozdeliť na 4 na seba nadväzujúce úrovne:

- **A. Organizácia (mesto)** - spočíva v organizácii zložiek mesta. V kontexte Smart City slúži k spracovaniu získaných (nameraných) dát a ich následnej aplikácii v rozhodovacích procesoch mesta.

  Komponenty a príklady ich naplnenia:

  1. Politický záväzok - vízia inteligentného mesta, ktorou vedenie mesta deklaruje dlhodobý smer, ktorým sa bude mesto vyvíjať.
  2. Organizácia a zodpovednosť - vznikne mestský útvar (v menších obciach jedna zodpovedná osoba), ktorý bude vykonávať agendu Smart City.
  3. Stratégia/Akčný plán - vznikne strategický a akčný plán, ktorý plány a zámery mesta premietne do štruktúry a konceptu Smart City.
  4. Spolupráca a dlhodobý partneri - zapojenie externých partnerov z prostredia firiem a univerzít.

- **B. Komunita (občan)** - podporuje život občana-jednotlivca ako súčasť metskej komunity. Zapája ho do jej života a rozhodovania.

  Komponenty a príklady ich naplnenia:

  1. Aktivuje a prepája - podpora občianskym iniciatívam a prepájanie verejného a súkromného sektora.
  2. Vytvára komunity a dáva priestor k sebarozvoju - mesto podporuje vznik a činnosť komunít (profesných, záujmových), do ktorých sa môžu zapájať jednotlivci z radov verejnosti, ale aj z podnikateľského prostredia.
  3. Zdieľa (ekonomika zdieľania) - podporuje, alebo priamo vytvára akékoľvek formy zdieľania, čím sa umožní efektívne napĺňanie potrieb obyvateľov a zároveň dochádza k efektívnemu a udržateľnému využívaniu majetku.

4. Kultivuje verejný priestor - kultivuje a investuje do kvalitného verejného priestoru, čím sa stáva atraktívnym pre občanov a zároveň láka podnikateľov do rozvoja verejného majetku.

- **C. Infraštruktúra** - zaoberá mestskou mobilitou, energetikou a ďalšími mestskými službami dohromady prepojenými a podporovanými informačno-komunikačnými technológiami. Kladie dôraz na synergie medzi jednotlivými implementovanými technológiami a dlhodobý prístup k rozvoju infraštruktúry.

  Komponenty a príklady ich naplnenia:

  1. Plošné pokrytie - je potrebné vytvoriť koncepciu, ktorá bude jednotná pre celé mesto, bude reagovať na požiadavky občanov a komunít, ale jednotlivé riešenia nepôjdu proti sebe.
  2. Viacúčelové riešenie - snaha o investície, ktoré v rámci jedného systému riešia viacero potrieb.
  3. Integrované riešenie - vzájomná previazanosť jednotlivých funkcií v Smart Cities a technológií, ktoré slúžia k jej napĺňaniu.
  4. Otvorené riešenie - verejné obstarávania sa konajú tak, aby ich mohlo dodávať viacero dodávateľov a aby sa nestalo, že bude mesto závislé iba na jednom z nich.

- **D. Výsledná kvalita života a atraktivita mesta** - je cieľom tvorby inteligentných miest v kontexte Smart City predstavuje merateľné a vyčísliteľné hodnoty atraktívnosti mesta z hľadiska jeho otvorenosti, čistoty, ekonomickej výhodnosti a povesti.

  Komponenty a príklady ich naplnenia:

  1. Kvalita života: digitálne, otvorené a kooperatívne mesto - pre vyhodnocovanie úspešnosti nasledovania stratégie Smart City sa využívajú dáta tak, aby pomohli znižovať neistotu pri rozhodovaní. Je nevyhnutné, aby sa dáta využívali objektívne.
  2. Kvalita života: zdravé a čisté mesto - sledovanie kvality verejného priestoru tak, aby bol čo najpríjemnejší a obsahoval,

Obr. 1.2: 4 úrovne Smart Cities a ich 16 komponentov [11]

čo najmenej emisií z verejnej dopravy, minimum hluku, dostatok priestoru pre nemotorovú dopravu, dostatok zelene a iné...

3. Kvalita života: mesto ekonomicky zaujímavé - snaha o príjemné miesto pre bývanie a prácu tak, aby si udržalo aktuálnych obyvateľov a lákalo príchod nových. To zase priláka nových podnikateľov a zlepší celkovú prosperitu mesta.

4. Brand: so skvelou povesťou - mesto sleduje a propaguje svoju činnosť aj navonok, vytvorí sa značka a podporuje sa informovanosť, že v tomto meste je dobré bývať.

Pre vyššie uvedený rámec platí, že čím lepšie sa zvládnu úrovne A a B, tým väčší bude dopad na nasledujúce celky C a D, čo prispeje k potrebe nižších investícií a prevádzkových nákladov.

Každý vyšší celok pozostáva zo štyroch hierarchicky usporiadaných komponent od jednoduchších ku komplexnejším (1, 2, 3, 4), ktorých poradie na seba procesne nadväzuje. Celý koncept potom predstavuje postup pre tvorbu inteligentného mesta a každý projekt požadujúci verejnú podporu by mal naplniť, pokiaľ je to relevantné,

všetkých 16 komponentov. Tento rámec tvorí absolútny základ každého mesta snažiaceho sa fungovať pomocou konceptu Smart City.

1.3 Otvorené dáta v Českej republike

Podľa organizácie Open Knowledge sú dáta definované ako otvorené, pokiaľ k nim môže ktokoľvek voľne pristupovať, používať ich, upravovať a zdieľať ich – za podmienok, ktoré najviac zabezpečia zachovanie pôvodu, teda vlastníctva, a otvorenosti [12]. Zároveň, by podľa organizácie Sunlight Foundation [13] mali napĺňať týchto 10 znakov, pričom pokiaľ niektorý z nich nie je naplnený, je to v poriadku, ale malo by to byť pri danej dátovej sade uvedené.

1. kompletnosť - dátová sada by mala obsahovať všetky dáta, ktoré boli nazhromaždené, okrem citlivých dát. Taktiež by mali obsahovať metadáta, ktoré popisujú štruktúru dát.

2. primárnosť - dáta zverejnené štátom, by mali byť považované za primárny zdroj, čiže by mali obsahovať originálne dáta. Samotné zverejnenie by potom malo prispieť ku verifikácii dát verejnosťou.

3. včasnosť - dáta by mali byť poskytované na verejnosť vo vhodných časových intervaloch, ideálne čo najrýchlejšie odkedy boli získané. Najlepšie využiteľné sú dáta zverejňované v reálnom čase.

4. jednoduchosť fyzického a elektronického prístupu - uvoľňovanie datasetov by malo byť čo najjednoduchšie. Pokiaľ si osoba musí po dáta prísť fyzicky, je nutné napísať žiadosť a fyzicky sa po ne dostaviť, čo zaťažuje aj zamestnancov. Elektronické vyzdvihnutie je teda omnoho efektívnejšie, aj keď tiež je potrebné splniť určité technické podmienky, aby nevznikali komplikácie najmä pre automatizované elektronické prístupy. To sa dá dosiahnuť sprístupnením API, cez ktoré sa dá jednoducho pristupovať vzdialene k dátam.

5. strojová čitateľnosť - informácie by mali byť zverejňované vo formáte, ktorý je jednoduchý na strojové spracovanie. Ak sú dáta

zverejnené vo forme fotky s ručne písanými poznámkami, užívateľ bude musieť použiť OCR softvér na optické rozpoznávanie textu, čo so sebou prináša mnoho komplikácií. Ideálne je preto zverejňovať dáta v takom formáte, ktorý sa jednoducho spracováva (napr. CSV) a zároveň by mal obsahovať aj dokumentáciu, ktorá bude popisovať štruktúru dát.

6. nediskriminovateľnosť - každý by mal možnosť pristupovať k dátam. Pokiaľ sú dáta skutočne otvorené, mali by byť prístupné všetkým bez potreby dodatočnej autentifikácie.

7. používanie štandardov s voľne dostupnou špecifikáciou (otvorené štandardy) - dáta musia byť vo formáte, ktorý je voľne dostupný pre ľubovoľné použitie, alebo do takého formátu prevoditeľný bezplatne dostupnou aplikáciou.

8. získanie licencie - dáta musia obsahovať licenciu, ktorá jasne definuje, že ide o verejné dáta vytvorené štátnym orgánom, ktoré sú dostupné bez obmedzení na ďalšie použitie

9. výkonnosť - schopnosť vyhľadávania dát v čase. Informácie zverejnené štátnym orgánom by mali byť mať sledovanie zmien a archiváciu v priebehu času.

10. cena použitia - jeden z najväčších problémov spojených so zverejňovaním dát je, že štát si pýta za sprístupnenie dátovej sady peniaze. Toto by sa nemalo stávať, pretože to obmedzuje použitie dát len pre tých, ktorí sú ochotní si za to zaplatiť.

Otvorené dáta spadajú pod Ministerstvo vnútra ČR, ktoré spravuje portál Otevřené data [14]. Čitateľ tam nájde aktuálne a najdôležitejšie informácie popisujúce stratégiu Českej republiky v tejto oblasti aj s príslušnou legislatívou, ako má správne vyzerať zverejnená dátová sada a ako ju zverejniť v Národnom katalógu otvorených dát verejnej správy ČR (ďalej "NKOD"), ktorý vznikol v roku 2015. NKOD je centrálnym katalógom otvorených dát v ČR. Obsahuje katalogizačné záznamy o dátových sadách zverejnených jednotlivými subjektmi verejnej správy ČR, vrátane odkazov na stiahnutie dát, ktoré sú uložené na webe príslušného poskytovateľa dát [15].

1.3.1 Kvalita otvorených dát

Klasifikáciu kvality otvorených dát definoval Tim Berners-Lee v roku 2006 ako 5 stupňov otvorenosti [16] viď. Obr. 1.3. Česká legislatíva považuje dáta za otvorené, ak spĺňajú požiadavky 3. a vyššieho stupňa. Pochopenie jednotlivých stupňov klasifikácie je dôležité, aby administratívni zamestnanci úradov rozumeli, prečo od nich zamestnanci technického oddelenia vyžadujú dátové sady v (pre nich) zvláštnom formáte CSV a prečo nestačí PDF dokument. Štandardom pre pracovanie s dátami je dodnes používanie programu Microsoft Excel a teda ukladanie vo formáte XLSX. Uvedomenie si minimálnych požiadaviek pre 3. stupeň kvality dát vedie k lepšiemu pochopeniu akú veľkú transformáciu musí podstúpiť dátová sada, kým opustí mestský informačný systém a je možné ju zverejniť na portáli otvorených dát.

Klasifikácia dát podľa T. Berners-Lee [17]:

1. stupeň - vyžaduje, aby bola distribúcia dátovej sady dostupná online a s jasným vymedzením podmienok použitia. Nekladie však žiadne požiadavky na dátové formáty, v ktorých sú distribúcie dátových sád zverejňované. Preto tento stupeň otvorenosti nie je považovaný za dostatočný.

2. stupeň - je charakteristický tým, že dátové sady sú poskytované v strojovo čitateľnom formáte. Príkladom je dátová sada, ktorá má charakter tabuľky. Dátová sada musí byť v takom formáte, ktorý umožňuje pomocou bežných programovacích prostriedkov získavať z tabuľky jednotlivé riadky a obsah buniek v riadkoch. Pokiaľ dátova sada nie je v strojovo čitateľnom formáte, je nutné ju predspracovávať pred čítaním. Napríklad pomocou technológie rozpoznávania textu z obrázku, alebo extrahovaním dát z HTML dokumentu. Z dôvodu, že 2. stupeň nešpecifikuje požiadavky na dátové formáty, v ktorých sú dátové sady zverejňované a umožňuje aj využitie rôznych proprietárnych, či komerčných formátov, nie je považovaný za dostatočný stupeň otvorenosti.

3. stupeň - vyžaduje oproti 2. stupňu naviac aj špecifikáciu formátu, v ktorom je dátová sada vyjadrená, aby bola otvorená. To znamená, aby bola vyhľadateľná a zdarma dostupná v sieti

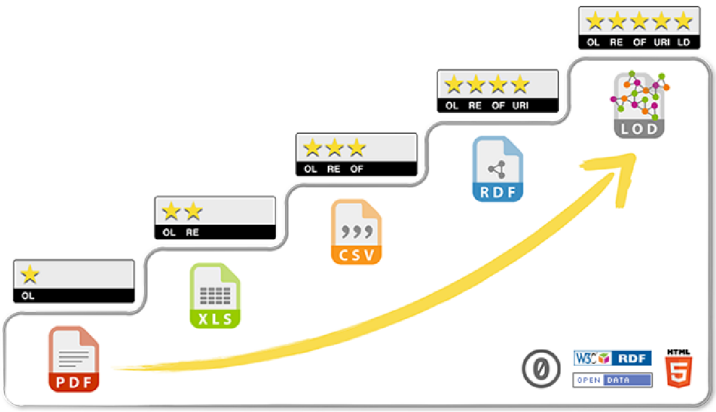
WWW a aby existovali voľne dostupné programovacie nástroje na jej spracovanie. Napríklad CSV, alebo XML. Medzi formáty vhodné pre túto úroveň otvorenosti dát sa príležitostne spomína aj formát XLSX. Ten však nie je vhodný na spracovanie pomocou automatických skriptov. Vo Veľkej Británii hodnotia všetky zverejnené dátové sady vo formáte XLSX stupňom 2 [18]. V dátovej sade môžu byť vyjadrené iba údaje tvoriace dátovú sadu a konštrukcie vyjadrujúce ich typ alebo sémantiku (sémantiku vyjadruje napríklad hlavička tabuľky v CSV súbore, alebo XML tagy a atribúty v XML elemente ohraničujúce údaj s určitým významom). Nepatria tam konštrukcie vyjadrujúce formátovanie (farby alebo štýl písma), konštrukcie s grafickými prvkami, ktoré nie sú v dátovej sade a podobne.

4. stupeň - zavádza povinnosť v dátovej sade identifikovať entity, ktorých sa týkajú údaje obsiahnuté v dátovej sade. Identifikátory musia mať tvar Internationalized Resource Identifier (IRI). Napríklad v dátovej sade o knihách musia mať všetky entity ako knihy, autori či vydavatelia priradený identifikátor v tvare IRI.

5. stupeň - najvyšší stupeň otvorenosti. Ďalej rozširuje 4. stupeň. Vyžaduje, aby distribúcie spĺňali štandardy prepojených dát (anglicky Linked Data). Štandardy prepojených dát sú postavené nad dobre známymi štandardami siete WWW a umožňujú vyjadrovať súvislosti medzi rôznymi dátovými sadami v podobe strojovo spracovateľných odkazov, ktoré sa podobajú hypertextovým odkazom v sieti WWW.

Skúsenosti s mestom Žďár nad Sázavou ukazujú, že bežná samospráva má v súčasnosti dáta otvorené buď na stupni 0, teda vôbec neprístupné verejnosti, alebo na 1. stupni, čiže nedostatočnom pre strojové spracovanie. Cieľom tejto práce bolo dosiahnuť aspoň 3. stupeň, ideálne ešte vyšší.

Do budúcnosti sa očakáva, že budú mestá priamo na začiatku verejného obstarávania požadovať, aby súčasťou dodávaného IT riešenia boli aj exporty dát v podobe vhodnej na zverejnenie. Dnešná situácia v Žďári nad Sázavou je taká, že každá dátová sada musí byť spracovaná predtým, ako bude zverejnená. Najlepšie je na tom dátová sada o hlasovaní mestského zastupiteľstva, ale ani ona neobsahuje



Obr. 1.3: Stupne otvorených dát [19]

API rozhranie na automatické získavanie dát. Medzikrokom medzi týmito dvoma stavmi je to, čo sa snaží dokázať táto diplomová práca. Vytvorenie automatizovaných skriptov, ktoré ušetria nutnosť ručnej úpravy dátových sád a urýchlia proces zverejňovania. Treba spomenúť, že toto riešenie nie je ani zďaleka dostatočné a vytvorené skripty bude treba ďalej udržiavať, lebo akákoľvek zmena v systéme, z ktorého sa dáta získavajú môže spôsobiť nefunkčnosť skriptu.

1.3.2 Otvorené formálne normy

Najväčší problém s dátami pri ich zverejňovaní je ich formát. Každá organizácia si rovnaké dátové sady spracováva inak a aj na úrovni organizácií môže dochádzať k tomu, že sú rovnaké dátové typy spracované rôznym spôsobom. Typickým príkladom je dátum, ktorý má obrovské množstvo formátov ako môže byť zapísaný. Napríklad `13.04.2020`, `13/04/2020`, `2020-04-13` a mnoho ďalších. Vo svete otvorených dát je korektný formát ten posledný z nich, ktorý používa dátový typ `xsd:date` z jazyka XML Schema. Ten umožňuje zaznamenať aj časovú zónu. Z tohto dôvodu je potrebné, aby všetky dátové typy, ktoré sa môžu nachádzať v zverejnených dátových sadách boli rovnaké a navzájom kompatibilné. Úlohou programátorov, ktorí s nimi budú neskôr pracovať nie je, aby písali prekladače schopné pracovať s rôznymi formátmi. Ich úlohou je vybrať si dátovú sadu, použiť ju a získať výsledok založený na dátach z nej.

Kompatibilita dátových sád musí byť zaručená minimálne na národnej, ale ideálne na nadnárodnej úrovni, aby so zverejnenými dátami vedeli pracovať rôzne krajiny. Nástroje, ktoré majú pomôcť toto dosiahnuť sa nazývajú otvorené formálne normy [20]. Sú to záväzné technické inštrukcie ukazujúce, akým spôsobom by sa mali konkrétne príklady zo skutočnosti zverejňovať. Zaisťujú, že rovnaké dáta publikované rôznymi poskytovateľmi budú interoperabilné. Vďaka tomu je umožnené také dáta jednoduchšie využívať nezávisle na tom, od ktorého sú poskytovateľa. Otvorené formálne normy sa naďalej vyvíjajú a postupne by mali pokrývať všetky typické situácie. Dnes sú dostupné normy napríklad pre aktuality, faktúry, turistické ciele, ale nie pre typ dát, ktorým sme sa venovali my.

Vyššie sme uviedli príklad dátumu ako dátového typu, ktorý musí byť presne špecifikovaný, aby nedochádzalo k problémom s nekompatibilnými dátovými sadami. Aj toto riešenie ponúka otvorená formálna norma, ktorá vychádza z medzinárodných štandardov a presne definuje, ako má ktorý dátový typ vyzerať [21].

1.4 Žďár nad Sázavou

Žďár nad Sázavou je mesto nachádzajúce v strednej časti Českej republiky v kraji Vysočina, severovýchodne od krajského mesta Jihlava. Je to okresné mesto s viac ako 20 000 obyvateľmi. Starostom je Ing. Martin Mrkos, ACCA a Ing. Michal Bačovský je garantom pre Smart City.

Mesto Žďár nad Sázavou od roku 2019 spolupracuje s Masarykovou Univerzitou. V rámci Laboratória servisných systémov pod vedením Ing. Leonarda Walletzkého, Ph.D. vznikol tím ľudí, ktorí pracujú na kompletnom nasadení koncepcie Smart City pre toto mesto. Komunikáciu s fakultou a s celým tímom zabezpečoval Ing. Michal Bačovský. Pôvodne boli naplánované mesačné stretnutia v Žďári nad Sázavou, avšak kvôli pandémii COVID-19 sa takéto osobné stretnutie uskutočnilo len jedno. Ostatná komunikácia teda prebiehala iba dištančnou formou pomocou e-mailov a telefonických konferencií.

Táto práca priamo nadväzuje na diplomovú prácu Mgr. Mareka Valka, ktorý sa zaoberal porovnaním open source riešení pre zverejňovanie otvorených dát [2]. Výsledkom jeho práce bolo, že za najvhodnejší

2. https://is.muni.cz/th/xsq6n/

nástroj zvolil systém CKAN, ktorý použil na vytvorenie testovacieho webu, kde odprezentoval, ako by toto zverejňovanie mohlo vyzerať. Dátový portál Žďáru nad Sázavou je stále pod správou Laboratória servisných systémov a v blízkej budúcnosti sa očakáva, že sa tento testovací server presunie pod správu mesta, kde sa bude oficiálne využívať. Paralelne s touto prácou vzniká ešte jedna diplomová práca, na ktorej pracuje Bc. Lucie Bartoňková, ktorá sa zaoberá tvorbou grafov a vizualizáciou otvorených dát.

Logickým ďalším krokom je naplniť dátový portál reálnymi dátami. Úlohou diplomovej práce je nastaviť proces, ktorý zabezpečí, že budú vytvorené a aktualizované aspoň tri dátové sady s čo najmenším (ideálne žiadnym) zásahom človeka. Skúsenosti získané pri tvorbe týchto troch dátových sád budú popísané v neskorších kapitolách a očakáva sa, že budú využité pri nasadzovaní koncepcie Smart City v ďalších mestách a obciach.

Žďár nad Sázovou je v oblasti Smart Cities vyspelým mestom v porovnaní s bežnou úrovňou v Českej republike. Vďačí za to najmä schopnému vedeniu, ktoré od 60. rokov minulého storočia dosiahlo rozvoj v technickej oblasti. Druhým faktorom je vyspelosť obyvateľstva v oblasti informačných a komunikačných technológií. Mesto dostalo podporu z operačného programu Zaměstnanost a pracuje na začatí riadenia mesta pomocou koncepcie Smart City.

Chytrý Žďár [22] je názov koncepcie Smart City pre mesto Žďár nad Sázavou. Momentálne je v procese tvorenia, do ktorého sa môže zapojiť každý občan mesta. Hlavným cieľom je zhodnotiť stav, v akom sa mesto nachádza. To znamená vyhodnotiť oblasti, v ktorých dáva nasadenie koncepcie Smart City najväčší zmysel, a zároveň využiť skúsenosti iných miest, tak aby neopakovalo ich chyby. Okrem vytvorenia koncepcie bude podpora z operačného programu Zaměstnanost využitá aj na vybudovanie hardvérovej a sotvérovej infraštruktúry, generelu, dopravy mesta a pasportu komunikácií.

Niektoré projekty z koncepcie Smart Cities sa im už podarilo uviesť do praxe. Vytvorili navigačnú aplikáciu ZDAR4U, ktorá slúži ako turistický sprievodca mestom. Mesto podporuje cyklistickú mobilitu stavbou cyklotrás a budovaním miest pre nabíjanie elektrických bicyklov. Mestský úrad zaviedol objednávkový systém, takže ľudia nemusia dlho čakať, kým sa dostanú na rad na úrade. Autobusy mestskej hromadnej dopravy sú vybavené systémom pre sledovanie ich polohy,

Obr. 1.4: Logo Žďáru nad Sázavou pre agendu Smart City [22]

takže občania môžu sledovať, za koľko minút príde autobus k ich zastávke.

Koncepcia Chytrý Žďár je nepolitický projekt, ktorého rozsah značne presahuje jedno volebné obdobie. Pre úspech tohto projektu je nevyhnutné zabezpečiť jeho podporu naprieč celým politickým spektrom, aby sa zabezpečilo jeho kontinuálne pokračovanie aj po prípadnej zmene vedenia mesta.

2 Proces publikácie otvorených dát

Táto kapitola bude popisovať proces, ktorý musí prebehnúť, kým sa dáta dostanú na portál otvorených dát od momentu ich vzniku. Čitateľ sa v nej tiež dozvie, aké problémy spojené so zverejňovaním dát mesta Žďár nad Sázavou boli počas tvorby tejto práce riešené.

Mnoho miest a obcí rieši rovnaké situácie inak a nedá sa popísať jeden univerzálny návod, ktorý popíše, ako zverejňovať otvorené dáta a zároveň bude vyhovovať všetkým. Aktuálna situácia v jednotlivých mestách je taktiež rôzna. Snaha o okamžité zverejnenie všetkých dát môže byť kontraproduktívna, najmä ak sa jedná o menšie mestá alebo obce, kde má agendu Smart Cities na starosti jeden človek. Jednoducho sa môže stať, že množstvo existujúcich dát spôsobí zahltenie povereného zamestnanca. Lepším prístupom je postupné zverejňovanie dát. Prvým a nevyhnutným krokom je teda analýza aktuálneho stavu a určenie si priorít, ktoré dáta sa budú zverejňovať ako prvé a postupne pridávať ďalšie.

Tu vzniknú prvé problémy a ťažkosti s akými sa bežný úradník stretne. Dáta nemusia byť v digitálnej podobe a pred zverejnením ich treba pripraviť, čo zase spôsobuje záťaž pre zamestnancov. Správnym prístupom je politicky presadiť nariadenie, ktoré bude záväzné pre všetky mestské podniky. Malo by obsahovať, že všetky budúce projekty budú povinne zverejňovať získané dáta. V ideálnom prípade budú tieto požiadavky popísané už v čase zadávania projektu, aby sa im mohli prispôsobiť všetci účastníci výberového konania. Výsledkom bude mestská aplikácia, ktorá bude schopná automatického zverejňovania dát bez ďalšieho nutného zásahu človeka.

2.1 Digitalizácia dát

Nevyhnutným základom pre zverejnenie dát v 21. storočí je mať ich digitálnu podobu. Samozrejme, mohli by sa zverejňovať aj dáta vo fyzickej podobe, ale ich využiteľnosť by bola značne limitovaná potrebou fyzického prístupu. Ak má inštitúcia veľké množstvo dát, ktoré sa nenachádzajú v digitálnej podobe, je nevyhnutné zvážiť, či má vôbec zmysel ich digitalizovať. Pokiaľ má mesto obrovské množstvo dát o kvalite úrody v 17. storočí, pre historikov a vedcov môže byť ich

dostupnosť užitočná, ale na bežné riadenie mesta ich asi nevyužijeme. Preto je lepšie venovať čas a zdroje dátam, ktoré môžu mať okamžitý efekt na chod mesta. Nevyhnutným základom je nastaviť procesy, aby všetky nové dáta vznikali automaticky v digitálnej podobe.

2.2 Získanie dát

Existujú mnohé spôsoby na získavanie dát, od ručného kopírovania zo zariadenia, ktoré ich získava, až po plne digitálny prístup k pripravenej dátovej sade. V priebehu tvorby tejto práce sme narazili na mnohé situácie, ktoré boli niekde medzi týmito extrémami. S jednoznačne najhorším prístupom sme sa stretli pri pokuse o získanie dát o využívaní cyklotrás. Tieto dáta boli jedny z prvých, o ktorých zverejnení sme diskutovali s mestom. Tento cieľ sa však nepodarilo naplniť. Kontaktovali sme správcu počítadiel dát so žiadosťou o ich sprístupnenie, ale odpoveď znela, že dáta sú poskytované len vo forme grafov tvorených prevádzkovateľom. Pokiaľ by sme chceli získať prístup k všetkým dátam v surovej podobe s pomocou API muselo by mesto zaplatiť niekoľko desiatok tisíc korún ročne. Toto je ukážkový príklad, ktorý sa v štátnej správe popisuje ako vendor lock. Dodávateľ dodal nejaké riešenie, ktorého funkčnosť je polovičná, a za plnú funkčnosť je potrebné si priplatiť alebo nahradiť počítadlo riešením od iného dodávateľa.

Medzi ďalšie dátové sady, ktoré sa nepodarilo zverejniť boli dáta z mestského vyvolávacieho systému. Z komunikácie s mestom sme sa dozvedeli, že program, ktorý používajú na spracovávanie, nepodporuje automatický import dát. Aplikácia nebola dostupná na webe, takže sa ani nedal napísať skript na ich automatické sťahovanie. Jediným spôsobom bolo poloautomatické riešenie, kedy by zamestnanec raz za mesiac dáta vyexportoval a uložil ich na vopred dohodnuté miesto, odkiaľ by ich už ďalej prevzali skripty na ďalšie spracovanie. Tento návrh však nebol akceptovaný zo strany mesta.

Najlepšia skúsenosť bola s datasetom, ktorý zverejňoval dáta z meračov teploty. Výhodou bolo, že dáta tvorí mesto, takže sa stačilo dohodnúť s pracovníkom IT oddelenia. Toto bol dobrý príklad spolupráce, ktorý ukázal, že pokiaľ existuje vhodne pripravená infraštruktúra, mesto dokáže relatívne rýchlo zverejniť dáta, ktoré boli predtým

neprístupné. V priebehu pár týždňov sme získali prístup ku niekoľkoročným záznamom o teplote.

2.3 Odstránenie citlivých dát

Každá dátová sada pochádza z iného zdroja, preto je nevyhnutné poriadne skontrolovať, aké dáta obsahuje. Nie všetky dáta sú totiž vhodné na zverejnenie, niektoré môžu obsahovať citlivé údaje o občanoch, ktorých zverejnenie by bolo porušovaním ich práva. Vhodné je zverejňovať anonymizované dáta tak, aby sa nedali použiť na identifikáciu konkrétnych osôb. Verejná správa by mala klásť dôraz na to, aby boli všetky dátové sady v súlade s platnou legislatívou.

V súvislosti s krízou spôsobenou ochorením COVID-19 sa na Slovensku stala situácia, kedy Národné centrum zdravotných informácií zverejnilo dáta, ktoré obsahovali údaje o všetkých výskytoch infekcie koronavírusom. Dáta boli natoľko špecifické, že si ktokoľvek mohol pozrieť bydlisko nakazených ľudí až na úroveň ulíc [1]. Chýbali iba popisné čísla. Dalo by sa diskutovať, či sa jednalo o dostatočnú anonymizáciu údajov. Na jednej strane je argument, že len na základe tých dát sa nedalo nikoho identifikovať, ale na druhej strane, najmä v malých obciach by aj takéto informácie mohli stačiť na identifikáciu. To môže vyvolať negatívne emócie medzi občanmi a môžu byť potom všeobecne proti zverejňovaniu dát.

2.4 Zverejnenie dát

Predposlednou fázou je zverejnenie dát. Dátové sady vytvorené mestom zvyčajne neobsahujú všetky údaje alebo nie sú v správnom formáte, ktorý je požadovaný otvorenými formálnymi normami. Preto je potrebné tieto údaje doplniť a upraviť pred zverejnením dátovej sady. Konkrétne sa jedná napríklad o rozdelenie hodnoty a jednotky do dvoch stĺpcov, prevedenie jednotky do formátu zodpovedajúcemu otvoreným formálnym normám (kwh na KWH) alebo doplnenie chýbajúcich údajov v dátovej sade.

---

1. Pravda, 30.03.2020 - Únik či cielený únik? Citlivé informácie o nakazených koronavírusom sú na webe. Dostupné online: `https://bit.ly/2ChNFnV`

Hotová sada sa potom zverejní v informačnom systéme CKAN. Pre prenos dát využívame rozhranie API, ktoré poskytuje jednoduchú komunikáciu so serverom a umožňuje všetky operácie potrebné pre prácu s dátami.

2.5 Udržiavanie dát

Z doterajších skúseností vyplýva, že do tejto fázy sa zvyčajne väčšina stredných a menších obcí nedostane. Deje sa to napríklad kvôli nedostatku financií, prípadne sa zmení vedenie obce a priority sa posunú iným smerom. Taktiež sa stáva, že si vedenie obce myslí, že ich aktivita končí zverejnením dátových sád - neuvedomujú si, že ich treba aj udržiavať. To obnáša pravidelnú aktualizáciu s novými dátami v ideálnom prípade pomocou skriptov, aby to nezaťažovalo pracovníkov obce. Nevýhodou skriptov je ich náchylnosť na zdroj dát. Je dôležité, aby v obci existoval človek, ktorý bude poznať celý kontext a prepojenie všetkých systémov. Nekoordinovaná zmena totiž môže spôsobiť, že skript dostane iné dáta aké očakáva a skončí chybou.

Životnosť dát nie je neobmedzená a užitočnosť niektorých z nich klesá s časom, je preto potrebné nastaviť proces, ktorý v určitom momente rozhodne o ich odstránení. Možnosťou je nastaviť ich ako neverejné, prípadne nainštalovať rozšírenie, ktoré sa bude zaoberať archiváciou dát.

3 Použité technológie

V tejto kapitole budú popísané všetky technológie, ktoré boli využité v tejto práci. Budeme sa zaoberať popisom informačného systému CKAN, ktorý bol zvolený ako vhodné riešenie pre zverejňovanie otvorených dát. Čitateľ sa dozvie o jeho štruktúre a možnostiach, ktoré poskytuje. Spomenieme aj aplikáciu CRON.

3.1 Comprehensive Knowledge Archive Network (CKAN)

CKAN je komunitne vyvíjaný open-source [1] projekt, ktorý slúži na zverejňovanie dátových sád. Výhodou je tiež možnosť inštalovania modulov, ktoré rozširujú jeho funkcionalitu. Základný program obsahuje iba jednoduché funkcie a užívateľ si ich môže inštalovať podľa potreby.

V projekte zverejňovania mestských dát pre mesto Žďár nad Sázavou sa využíva práve táto platforma. Na základe porovnania rôznych dostupných riešení, ktoré vo svojej diplomovej práci popísal Mgr. Marek Vaľko vychádza CKAN v súčasnosti ako najlepšie riešenie pre mestskú samosprávu [23]. Umožňuje dostatočné množstvo funkcií, ktoré by mali pokryť aj najzložitejšie požiadavky. Jeho kvalitu dokazuje aj množstvo zahraničných miest, ktorých infraštruktúra ohľadne otvorených dát je postavená práve na CKANe. V čase písania tejto práce existuje po svete celkovo 199 inštancií medzi, ktorými sa nachádzajú mestá ako Amsterdam, Berlín, Záhreb, ale aj celonárodné portály Veľkej Británie, Rakúska, Holandska a mnoho ďalších miest a krajín po celom svete [24].

Táto kapitola si nekladie za cieľ napísať vyčerpávajúcu príručku na používanie CKANu. Čitateľ v nej nenájde žiadne inštrukcie ako ho nainštalovať a na čo všetko sa dá použiť. Myslíme si, že to bolo dostatočne popísané v práci Mgr. Vaľka a počas riešenia tejto práce sme ani nepotrebovali využiť všetky dostupné možnosti. Naopak, bude zameraná najmä na funkcie využité pri riešení tejto diplomovej práce

1. Počítačový softvér, ktorý má zdrojový kód zverejnený pod licenciou, ktorej vlastník umožňuje používateľovi študovať, meniť a distribuovať tento softvér komukoľvek za akýmkoľvek účelom.

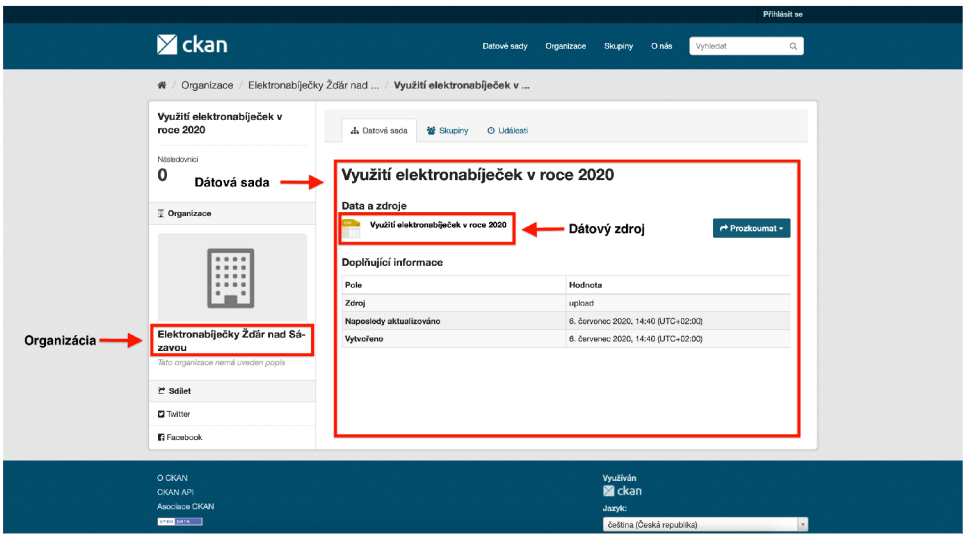
s dostatočným kontextom pre ďalších ľudí snažiacich sa vkladať nové dátové sady.

3.1.1 Organizácia dát

Organizácia dát v CKANe má logickú štruktúru. Pre používateľa poskytuje prehľadný systém, takže orientácia na webe je jednoduchá. Z pohľadu vývojára to už tak jednoduché nie je. Aj v dokumentácii samého CKANu je množstvo pojmov, ktoré znejú podobne a je potrebné ich mať ujasnené predtým ako sa človek pustí do písania skriptov pre jeho ovládanie. Základnou stavebnou jednotkou je dataset, v terminológii CKANu označovaný ako "package". Toto označenie sa využíva aj v popise jednotlivých metód API rozhrania[2]. Každá dátová sada môže obsahovať rozličný počet dátových zdrojov. V prostredí CKAN sú označované ako resource. Pod pojmom dátový zdroj si môžeme predstaviť dáta v rôznych formátoch (CSV, XLSX, PDF a iné). Toto je vhodné najmä pre používateľov, pretože každý ocení možnosť použitia dát v inom formáte. Dôležité je, aby aspoň jeden zdroj zabezpečoval vysokú úroveň otvorenosti dát. Samotný CKAN nevykonáva žiadnu kontrolu vložených dát. Je na meste, aby zabezpečilo, že dáta vychádzajúce z mestských systémov sú zverejniteľné, neobsahujú žiadne citlivé dáta a spĺňajú všetky legislatívne požiadavky. Každý zdroj môže mať vlastný popis, formát a dátový slovník popisujúci štruktúru dát. Skupina zdrojov tvorí dátovú sadu - ucelený súbor, ktorý vyjadruje určitý kontext. Napríklad pre každý kalendárny rok je jeden zdroj. Rôzne skupiny zdrojov, ktoré sa líšia významovo, ale majú spoločný pôvod sa združujú v organizáciách.

Pri dizajnovaní štruktúry portálu je dobré myslieť aj na úrovne oprávnení. CKAN poskytuje robustný systém správy práv a užívateľov. Dá sa nastaviť, že konkrétnu organizáciu budú spravovať iba vybraní užívatelia a verejnosť ju bude môcť iba čítať. Pri použití API celá autorizácia prebieha jednoducho pomocou kľúča, ktorý má každý používateľ zobrazený v profile.

2. Application Programming Interface označuje rozhranie pre programovanie aplikácií. Ide o zbierku funkcí, ktoré môže programátor využívať. API určuje, akým spôsobom sú funkcie programu volané.



Obr. 3.1: Štruktúra dát v systéme CKAN.

3.1.2 CKAN API

Pre vzdialenú automatizáciu mestských dát je nevyhnutné naučiť sa API, ktoré CKAN poskytuje. V dokumentácii [3] sú prehľadne popísané všetky možnosti, ktoré rozhranie podporuje. V tejto práci sme využili najmä funkcie na vkladanie nových a upravovanie existujúcich dát. Celý projekt je písaný v jazyku Python a na posielanie požiadaviek cez API sme využili modul `requests`. Požiadavka sa posiela vo formáte JSON, pričom každá funkcia má vlastne špecifikované vlastnosti, ktoré musia byť prítomné pre úspešné vykonanie príkazu. Volania API sa dajú rozdeliť na dve skupiny. Prvá z nich slúži na získavanie dát tzv. "get"metódy, ktoré žiadnym spôsobom nemodifikujú dáta uložené na serveri a teda sú voľne prístupné pre verejnosť. Patria sem najmä volania pre získanie zoznamu organizácií, dátových sád, zdrojov, prípadne samého obsahu dátových sád. V druhej skupine sa nachádzajú funkcie, ktoré upravujú informácie uložené v CKANe. Ako už bolo spomenuté v predchádzajúcej sekcii, autorizácia prebieha pomocou API kľúča. Ten je základným predpokladom pre úspešné vykonanie príkazu. Pokiaľ je kľúč v poriadku, užívateľ môže sťahovať dáta, ktoré

3. `https://docs.ckan.org/en/2.8/api/index.html`

ešte nie sú verejne prístupné, upravovať informácie o dátových sadách (názov, popis, tag a iné), prípadne zmazať celú dátovú sadu.

3.2 CRON

Cron je utilita vyskytujúca sa na operačných systémoch Unix. Slúži na periodické spúšťanie programov podľa zadaných požiadaviek. Pre potreby skriptov využívaných v tejto práci je nastavenie programu `cron` nasledovné:

```
0 0 1 * * <cesta ku~skriptu>/job.sh
```

Význam tohto zápisu znamená, že skript `job.sh` sa spustí v nultej minúte, nultej hodiny (čiže v čase 00:00), prvého dňa v mesiaci. V prípade, že by užívateľ potreboval spúšťanie každého skriptu v iných intervaloch, doporučujeme vytvoriť viacero samostatných úloh pre `cron`. Nižšie sa nachádza schéma možností nastavenia úloh.

```
 --------- minúty (0 - 59)
|  --------- hodiny (0 - 23)
| |  --------- dni mesiaca (1 - 31)
| | |  --------- mesiac (1 - 12)
| | | |  --------- deň týždňa (0 - 6) (od~nedele do~soboty;
| | | | |   7 je taktiež nedeľa na~niektorých systémoch)
| | | | |
| | | | |
* * * * * <príkaz na~vykonanie>
```

4 Automatizovaná publikácia dátových sád

Najdôležitejšou časťou tejto práce bolo dosiahnutie praktickej ukážky, že je možné automatizovať proces publikácie dát. To sa nám aj podarilo a v tejto kapitole bude popísaná implementácia skriptov, ktoré slúžia na zverejňovanie dátových sád o využívaní elektronabíjačiek, o teplote nameranej v Žďári nad Sázavou a o výsledkoch hlasovania v mestskom zastupiteľstve.

Pre svoje fungovanie skripty nevyžadujú zložitú prípravu. Po rozbalení archívu, prípadne naklonovaniu repozitára z portálu GitHub, so zdrojovými súbormi je potrebné nainštalovať knižnice, ktoré skripty používajú. Každý skript vo svojom priečinku obsahuje súbor `requirements.txt`, ktorý obsahuje všetky závislosti pre daný skript. Ich inštalácia potom prebieha pomocou programu `pip` a príkazu `pip install -r requirements.txt`.

4.1 Dátová sada o využívaní elektronabíjačiek

Od augusta 2017 má Žďár nad Sázavou k dipozícii elektrické nabíjačky. Občania si v nich môžu nabíjať autá a iné elektrické dopravné prostriedky. Nabíjacie stanice sa nachádzajú v dvoch lokalitách. Pri Starej radnici sa nachádza šesť 230V nabíjačiek pre dobíjanie elektrobicyklov, skútrov a iných jednostopových vozidiel. Druhá nabíjačka sa nachádza na centrálnom parkovisku a poskytuje špeciálnu nabíjaciu zásuvku Mennekes typ 2 s výkonom 22 kW a tri zásuvky 230V.

4.1.1 Zdroj dát

Nabíjacie stanice sú začlenené do portálu Evmapa, ktorý združuje stanice po celej strednej Európe. Je to česká firma, takže sa primárne zameriava na tie v Českej republike. Poskytovateľom staníc umožňuje prevádzkovanie platieb a klientom stačí jednotné prihlásenie pre nabíjanie vo všetkých zapojených staniciach. Môžu tiež vidieť obsadenosť zapojených staníc v reálnom čase. Žďár nad Sázavou je začlenený v tomto portáli a využíva jeho služby pre manažment prístupov. Portál Evmapa si berie províziu z každého dobíjania, ale v súčasnoti je

nabíjanie na staniciach zdarma, takže ani mesto neplatí žiadne poplatky.

Pre problematiku zverejňovania dát je najzaujímavejšia podstránka "Provize". Mestom poverená osoba si tu môže pozerať využitie jednotlivých staníc, množstvo spotrebovanej energie a informácie o začiatku a konci nabíjania. Ak by bolo nabíjanie spoplatnené, užívateľ by taktiež videl cenu odoberanej energie rozloženú na jednotlivé zložky. Informačnému systému chýba iba jediná funkcia, ktorou je API pre jednoduchý prístup k dátam. Jediný spôsob exportu je uloženie do PDF, ale to neposkytuje vhodnú úroveň otvorenosti. Druhou možnosťou je ručné kopírovanie a vkladanie do tabuľkového editora. To by mohlo stačiť na základné operácie, ale pre potreby pravidelného zverejňovania na portáli otvorených dát je to absolútne nevhodné. Ak chceme zabezpečiť pravidelné zdieľanie dát nesmie to pridávať zamestnancom nadbytočnú prácu. Cieľom tejto práce bolo implementovať riešenie, ktoré by tento proces zjednodušilo a urýchlilo.

4.1.2 Formát očakávaných dát na vstupe

Dáta sa získavajú spracovaním HTML kódu z portálu Evmapa. Štruktúru stiahnutej stránky tvoria dve tabuľky, pričom sa použije iba druhá z nich. V druhej tabuľke sa nachádzajú všetky dáta. Odstraňuje sa však prvý a posledný riadok, kde sa nachádza záhlavie a zápätie. Zo stĺpcov sa zahodia všetky okrem druhého (začiatok a koniec nabíjania) a piateho (množstvo odobratej energie). V prílohe C je ukážka zjednodušenej štruktúry očakávaného vstupu. Pokiaľ by sa v budúcnosti vstup zmenil, bude potrebné vykonať zmeny vo funkcii `clean_data()`.

4.1.3 Implementácia

Tu budú podrobne popísané kroky, ktoré sa vykonávajú počas behu skriptu pre automatický import dát. Na začiatku je dôležité ujasniť si ako má vyzerať štruktúra dátovej sady. Konzultovali sme túto problematiku so zamestnancom Ministerstva vnútra Českej republiky. Prioritne by sa malo vychádzať z otvorených formálnych noriem, ak také existujú pre danú doménu. V prípade, že otvorená formálna norma nie je k dispozícii, bolo by dobré zvážiť jej pridanie do verejného repozitára na stránke GitHub, kde sa všetky normy vyvíjajú.

Ako už bolo spomenuté vyššie, nie všetky zdroje dát disponujú rozhraním API, takže už len samotné získanie dát nie je triviálna úloha a líši sa od konkrétnej dátovej sady. Pre portál Evmapa sme využili metódu dolovania dát (z angličtiny web scraping), ktorá zabezpečuje strojové sťahovanie dát z webových stránok. Toto okrem iného obnášalo aj ananlýzu portálu Evmapa. Museli sme odsledovať URL adresy používané na odosielanie dát a tiež zachytávať hlavičky požiadaviek s ich obsahom. Výsledkom tohto pozorovania bolo, že prihlasovanie prebieha POST požiadavkou na adresu `/users_login.php`, pričom obsahuje argumenty `em` pre prihlasovací email a `ps` pre heslo. Pre získanie dát bolo potrebné toto chovanie napodobniť. Nemenej dôležitou úlohou bolo samotné získavanie dát, ktoré sa robí pomocou GET požiadavky. V tomto prípade je cieľová adresa `/muj-ucet`, ku ktorej sa pridávajú nasledujúce argumenty:

- owner - význam nezistený.
- period - obdobie za ktoré sa majú zobraziť dáta v tvare RRRRMM napríklad 202006.
- station - identifikátor stanice.
- pump - identifikátor zásuvky.

Ich využitie bude popísané neskôr.

Táto metóda získavania dát je veľmi efektívna, ale problémom je jej životný cyklus, ktorý sa momentálne nedá odhadnúť. Ak sa nič zásadné nezmení, aplikácia by mala fungovať niekoľko rokov. Stačí však redizajn informačného systému a zásahy do zdrojového kódu budú nevyhnutné. Čitateľ by mal v tomto texte nájsť všetky rizikové miesta, ktoré môžu nastať a v prípade potreby by nemalo byť zložité modifikovať zdrojový kód, tak aby zodpovedal aktuálnej situácii. Samozrejme, závisí to však na rozsahu zmien, ktoré webový portál podstúpi. Medzi ďalšie z rizík patrí cielené znefunkčnenie dolovania dát prevádzkovateľom webu. Poskytovanie API rozhrania je celkom jednoduchý spôsob ako pridať novú funkcionalitu, ktorá sa dá spoplatniť. V takom prípade by bolo jednoduchšie zaplatiť za prístup ako kontinuálne upravovať skript do funkčnej podoby.

Úlohou skriptu je automatizovať všetku činnosť, ktorá slúži pre vkladanie dát do systému CKAN. Ovládanie prebieha pomocou

argumentov pri spustení programu viď. tabuľka 4.1. Užívateľ si definuje rozsah vkladaného obdobia, ktoré sa môže pohybovať od jedného mesiaca (v prípade rovnakého počiatočného a koncového rozsahu) až po import kompletného rozsahu poskytovaného portálom Evmapa. To sa dá využiť pre inicializačné vloženie dát do systému CKAN. Vzájomne vylučujúce sa argumenty `-head/-no-head` zase slúžia na určenie, či sa bude do dátovej sady zapisovať aj hlavička tabuľky, alebo iba samotné dáta. Je to dôležité pre pravidelné spúšťanie skriptu, kedy už dátová sada existuje a iba sa do nej vkladajú nové dáta. Skript vykonáva dve operácie, ktoré sa vzájomne nedajú kombinovať. Ak je vytváraná nová dátová sada, je potrebné použiť argument `-head`, čím sa zabezpečí, že všetky dátové sady budú obsahovať hlavičku. Pokiaľ potrebujeme vkladať dáta do už existujúcej, tak sa použije argument `-no-head`. Zjednodušene by sa dalo povedať, že prvý spôsob je pre hromadné vkladanie nových dát a druhý pre priebežné dopĺňanie. Je dôležité myslieť na to, že dátové sady neobsahujú žiadnu kontrolu integrity dát. Užívateľ by sa mal pred spustením skriptu zamyslieť a uvedomiť si, aké dáta sa v systéme CKAN už nachádzajú a vyvarovať sa opakovaného vkladania rovnakých dát. Riešenie všetkých možných situácií, ktoré môžu nastať by bolo na samostatnú prácu a za daných okolností kedy sa očakáva relatívne krátka životnosť skriptu to nemá veľký význam. V prípade, že dôjde k poškodeniu integrity dát, preto doporučujeme zmazať túto dátovú sadu a opätovne nahrať dáta z celého zasiahnutého roka.

Pre správnu funkčnosť skriptu je potrebné, aby boli nastavené všetky požadované hodnoty v konfiguračnom súbore. V tejto sekcii bude vysvetlené, načo sa jednotlivé položky používajú. Cieľom bolo, aby užívateľ nemusel zasahovať do zdrojového kódu a základné hodnoty premenných vedel nastaviť aj človek, ktorý zdrojovému kódu nerozumie.

- url - adresa na server, na ktorom sa prevádzkuje systém CKAN.
- url_api - adresa na endpoint[1] API rozhrania serveru so systémom CKAN.

1. endpoint - miesto systému, ktoré je otvorené pre komunikáciu s vonkajším svetom

Tabuľka 4.1: Argumenty pre ovládanie skriptu elektronabijecky.py

skrátený argument	dlhý argument	popis
-h	–help	vypíše pomoc o spustení programu
-sy ROK	–start-year ROK	import od roku
-sm MESIAC	–start-month MESIAC	import od mesiaca
-ey ROK	–end-year ROK	import do roku
-em MESIAC	–end-month MESIAC	import do mesiaca
–head/–no-head	–head/–no-head	s/bez zapísania hlavičky CSV súboru

- apikey - kľúč pre autorizáciu činnosti v prostredí CKAN, typicky nie je potrebná pokiaľ sa jedná o operácie čítania, ale nevyhnutná pokiaľ sa jedná o akúkoľvek modifikáciu dát v systéme.
- post_login_url - url adresa, na ktorú sa odosiela požiadavka na prihlásenie sa do portálu Evmapa.
- request_url - url adresa, z ktorej sa získavajú dáta.
- package - systémový názov dátovej sady. Nemal by obsahovať diakritiku a končiť by sa mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.
- package_name - ľudský názov dátovej sady aký bude vidieť v portále CKAN, končiť by sa mal medzerou, pretože skript na koniec automaticky pridáva aktuálny rok.
- owner_org - názov organizácie, do ktorej sa budú dáta pridávať.
- filename - názov súbora, do ktorého sa bude dátová sada ukladať lokálne na serveri. Nemal by obsahovať diakritiku a končiť by

mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.

- extension - formát súboru dátovej sady.
- table_head - pole názvov stĺpcov tabuľky.
- resource_iri - IRI identifikátor dátovej sady.
- station_dict - slovník staníc.
- socket_dict - slovník druhov zásuviek.
- payload - slovník s údajmi pre prihlásenie sa do systému.

Príklad konfiguračného súboru s reálnymi hodnotami sa nachádza v prílohe B.

Po spustení skriptu sa vykoná kontrola zadaných argumentov. Rok by mal mať štyri a mesiac dva znaky, pričom počiatočný rok musí byť menší ako koncový. Ak to tak nie je, skript sa ukončí a do logovacieho súboru sa zaznamená informáciu o chybnom ukončení programu. Taktiež sa vykoná záloha existujúcich súborov, ak by neskôr program zlyhal. Skript obsahuje aj logovanie, takže užívateľ môže sledovať beh skriptu pomocou rôznej úrovne správ, prípadne sa dá pomocou neho spätne zistiť, čo spôsobilo zlyhanie skriptu.

Ešte pred začatím spracovávania dát sa vytvoria zálohy lokálnych súborov, ktoré budú použité, ak nastane nejaký problém (napríklad sa preruší spojenie so serverom počas behu skriptu). Ak skript vyhodnotí, že takáto situácia nastala, zavolá sa funkcia `rollback()`, ktorá ukončí beh skriptu, zaloguje chybu a nahrá zálohované dáta naspäť na server.

Primárny beh programu prebieha v cykle viď. Listing 4.1, ktorý má na starosti spracovanie dát za každú časovú jednotku. V tomto prípade ide o jeden mesiac. Funkcia `month_year_iter()` vracia spracované dáta pripravené na zapísanie do súboru, mesiac a rok a počítadlo pre pokračovanie, ak už sú zapísané všetky dáta v danom roku.

Listing 4.1: For cyklus pre riadenie programu

```
for data, y, m, counter in month_year_iter(args.
    start_month, args.start_year, args.end_month,
    args.end_year):
```

Dôležité je uvedomiť si logiku, na akej je postavený portál Evmapa. Pokiaľ má dátová sada obsahovať informácie o tom, ktoré konktrétne stojany a zásuvky boli použité je potrebné ich diferencovať. Skúmaním portálu Evmapa sme zistili, že toto sa dá dosiahnuť, ak požiadavka na stiahnutie dát obsahuje argument `station` a/alebo `pump`. V tabuľke 4.2 sú ukázané identifikačné čísla, ktoré sa používajú pre identifikáciu nabíjacích staníc v Žďári nad Sázavou. Zároveň platí, že pred Starou radnicou sa nachádza iba zásuvka s názvom Stojan 1 a pred Centrálnym parkoviskom sú zvyšné dve.

Tabuľka 4.2: Identifikačné čísla pre nabíjacie stanice a zásuvky

názov	ID
Žďár nad Sázavou Centrální parkoviště	319
Žďár nad Sázavou Stará Radnice	351

názov	ID
Konektor 2 Mennekes 22kW	343
Zásuvka 230V	344
Stojan 1	391

Na to, aby sme stiahli všetky požadované dáta, je potrebné spraviť celkovo 3 požiadavky. Vďaka tomu dokážeme rozlíšiť, ktorý odber energie prebehol kde. Pre sťahovanie dát sa využíva funkcia `get_data(url, period, station, pump)`, ktorá zabezpečuje nadviazanie spojenia, autentifikáciu, odoslanie požiadavky na konkrétne dáta a ich prijatie. Pre odosielanie a prijímanie GET a POST požiadaviek využívame knižnicu `requests`. Výstupom z tejto funkcie sú nespracované dáta. V prípade zlyhania dotazu sa dôvod chyby zaznamená do logovacieho súboru a skript sa ukončí.

Stiahnuté dáta pokračujú do funkcie `clean_data(raw_data)`, kde prebehne ich vyčistenie a odstránenie dát, ktoré nie sú vhodné na publikovanie. Na prácu s HTML obsahom používame knižnicu `Beatiful Soup`, ktorá umožňuje jednoduchú prácu s HTML objektami. Pomocou jednoduchého dotazu z neho bola vybratá iba tabuľka, ktorá obsahuje pre nás zaujímavé dáta. Vo zvyšku funkcie `clean_data()` program

vykonáva ďalšie operácie pre úpravu dát. HTML objekt je pretransformovaný do poľa, kde každý prvok je pole obsahujúce dáta patriace do jednotlivých riadkov pôvodnej tabuľky. Takto pripravené dvojrozmerné pole už je očistené od všetkých nepotrebných znakov a obsahuje iba dáta samotnej tabuľky. To nám umožňuje jednoduchú manipuláciu s riadkami a zbavujeme sa tých nepotrebných - napríklad hlavička tabuľky, ktorá bude nahradená tou z konfiguračného súboru.

Pole s vyčistenými dátami pokračuje na ďalšie spracovanie do funkcie `prepare_data()`, kde dochádza k ich ďalšej modifikácii. Sú tu upravované tak, aby spĺňali zákonom definované požiadavky a taktiež aj požiadavky popísané v otvorených formálnych normách. V zdrojových dátach je čas označený ako `23.08.2019 19:26 - 19:53`, ale otvorené formálne normy vyžadujú, aby sa rozdelil do dvoch samostatných stĺpcov, kde jeden bude pre počiatočný a druhý pre koncový dátum. Formát dátumu je taktiež dôležitý. Oddeľovačom by nemala byť bodka, ale pomlčka a hodiny, minúty a sekundy by mali byť oddelené dvojbodkou. Dátum a čas by mal byť oddelený pomocou znaku `T`, takže výsledný formát po všetkých modifikáciách je nasledovný: `23-08-2019T19:26:00`. Okrem dátumu sa v tejto funkcii pracuje aj s pomenovaním stanice. Namiesto bežného ľudsky čitateľného názvu bol použitý identifikátor v podobe IRI. Množstvo a jednotka spotrebovanej energie by tiež malo byť rozdelené do dvoch samostatných stĺpcov pričom jednotka sa zapisuje pomocou veľkých písmen. Namiesto `5.373kWh` teda dostaneme dva stĺpce, jeden s hodnotou `5.373` a druhý s jednotkou `KWH`. Takto upravené dáta sú už vo svojej finálnej podobe. Opäť sa ukladajú do dvojrozmerného poľa, kde prvý rozmer sú riadky tabuľky skladajúce sa z poľa hodnôt, kde hodnoty sú jednotlivé stĺpce daného riadku.

Takto spracované dáta sú nachystané a pripravené na zverejnenie. Skript sa vracia naspäť do hlavného cyklu 4.1, ktorého úlohou je uložiť dáta do súboru a nahrať ich do portálu CKAN. Názov súboru je možné nastaviť pomocou premennej `filename` a `extension` pre príponu v konfiguračnom súbore. Dáta v jednom súbore sú za celý kalendárny rok, takže za názov sa ešte pridáva rok, v ktorom boli dáta vytvorené. My sme zvolili ukladanie do zložky `backup`, aby sa odlíšili dátové súbory od ostatných. Výsledné meno teda vyzerá napríklad `backup/elektronabijecky_2020.csv`.

Tabuľka 4.3: Návratové hodnoty skriptu elektronabijecky.py

hodnota	konštanta	popis
1	EXIT_REQUEST_ERROR	nastala chyba počas komunikácie so servrom
2	EXIT_ROLLBACK_SUCCESS	nepodarilo sa nahrať dáta, dátová sada bola vrátená do pôvodného stavu
3	EXIT_ROLLBACK_ERROR	nepodarilo sa nahrať dáta ani vrátiť dátovú sadu do pôvodného stavu
5	EXIT_MISSING_CONFIG	chýbajúci konfiguračný súbor
6	EXIT_ARGUMENT_ERROR	zadaný nevhodný argument pri spustení programu
7	EXIT_FILE_ERROR	chyba pri otváraní súboru

Hlavný cyklus sa ukončuje v prípade, že sú splnené dve podmienky. Sú spracované všetky mesiace v danom roku (štandardne 12, respektíve menej ak spracovávame aktuálny rok) a zároveň, boli spracované všetky tri požiadavky pre stiahnutie všetkých dát. Ak sa podmienka splní, skript sa dostáva do poslednej časti, ktorou je odosielanie dát na portál CKAN.

Skript najskôr skontroluje, či v systéme už existuje dátová sada s rovnakým názvom ako aktuálne vkladaná. Ak áno, tak získa jej identifikátor (`package_id`) a bude sa aktualizovať existujúca, pokiaľ nie, tak sa vytvorí nová. Identifikátor zdroja `resource_id` slúži na určenie existujúceho zdroja, ktorý bude aktualizovaný. Ak dátová sada s identifikátorom `package_id` zatiaľ nemá žiaden zdroj, vytvorí sa v nej nový.

Pri vytváraní alebo aktualizovaní dátovej sady je potrebné odosielať nasledujúce informácie:

- id - identifikátor aktualizovaného zdroja (ak sa vytvára nový zdroj tak sa tento parameter vynechá).
- package_id - identifikátor aktualizovanej dátovej sady.
- name - meno zdroja dát.
- format - formát vkladaného súboru.
- url - url so zdrojom dát ak sa vkladá pomocou url adresy. V našom prípade nahrávame lokálne súbory, ktoré nemajú url adresu, ale CKAN tento parameter vyžaduje, preto tam je reťazec üpload".

Po úspešnom nahratí údajov na portál otvorených dát sa vymažú zálohované dáta a ukončí sa vykonávanie skriptu. Návratová hodnota v takomto prípade je 0. Pokiaľ nastane problém skript vráti nenulovú hodnotu, jej významy sú popísané v tabuľke 4.3.

4.2 Dátová sada o teplote na mestskom úrade

Druhú dátovú sadu, ktorú sme zverejnili v spolupráci so Žďárom nad Sázavou, bola dátová sada o nameranej teplote na mestskom úrade. Od roku 2006 má mesto nainštalované teplotné senzory, ktoré merajú teplotu a zverejňujú ju na stránke mesta [2], kde sú grafy zverejňované v podobe obrázkov, takže získanie konkrétnych dát bolo nemožné. Toto sa teraz mení a verejnosť dostane prístup k hodnotám nameraným v 15-minútových intervaloch pomocou dvoch teplomerov.

4.2.1 Zdroj dát

Na rozdiel od dát o využití elektronabíjačiek, ktoré poskytuje externá firma, dáta o teplote spravuje priamo mesto. Na požiadanie nám zamestnanec Oddelenia informatiky sprístupnil celú databázu nameraných hodnôt od roku 2006. Dáta sú dostupné vo formátoch CSV a XML.

2. https://www.zdarns.cz/online-sluzby/webkamera-teplomer-pocasi

4.2.2 Formát očakávaných dát na vstupe

Na vstupe sa očakáva CSV súbor, ktorý obsahuje štyri stĺpce oddelené bodkočiarkou. Na prvom riadku sa nachádza záhlavie a na ďalších nasledujú dáta. Pre naše potreby sú dôležité najmä prvé tri hodnoty - čas merania, teplota na prvom senzore a teplota na druhom senzore. Štvrtý stĺpec obsahuje informáciu o tom, ktorý stĺpec nameral nižšiu hodnotu. To však nie je pre nás zaujímavé, pretože interpretácia dát je na používateľovi. V prílohe E je ukážka očakávaného vstupu. Pokiaľ by sa v budúcnosti vstup zmenil, bude potrebné vykonať zmeny vo funkcii `clean_data()`.

4.2.3 Implementácia

Implementácia vychádzala zo skriptu o elektronabíjačkách, takže bude do istej miery podobná a preto bude popísaná zjednodušene. Viac informácií nájde čitateľ v podkapitole 4.1.3. Pre ovládanie aplikácie so zdrojovými dátami, dostupnej na adrese [3], sa používajú GET parametre. Sú to nasledujúce:

- P - kontrolný reťazec pre prihlásenie sa do aplikácie
- F - výstupný formát
  - X - XML
  - T - text
- R - rok (2006 a viac)
- M - mesiac (1 až 12)
- D - nepovinný parameter deň (1-31) - pokiaľ je neuvedený, exportuje sa celý mesiac

Pred prvým spustením skriptu je potrebné vytvoriť konfiguračný súbor s názvom `config.toml`, ktorý sa nachádza v rovnakom priečinku ako samotný skript. Konfiguračný súbor je miesto, kde užívateľ môže nastaviť všetky informácie bez toho, aby musel upravovať zdrojový kód.

3. `https://data.zdarns.cz/temp.asp`

- url - adresa na server, na ktorom sa prevádzkuje systém CKAN.
- url_api - adresa na endpoint[4] API rozhrania serveru so systémom CKAN.
- apikey - kľúč pre autorizáciu činnosti v prostredí CKAN, typicky nie je potrebná pokiaľ sa jedná o operácie čítania, ale nevyhnutná pokiaľ sa jedná o akúkoľvek modifikáciu dát v systéme.
- request_url - url adresa, z ktorej sa získavajú dáta, je potrebné špecifikovať parametre P (kontrolný reťazec) a F (formát dát).
- package - systémový názov dátovej sady, nemal by obsahovať diakritiku a končiť by sa mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.
- package_name - ľudský názov dátovej sady, aký bude vidieť v portále CKAN. Končiť by sa mal medzerou, pretože skript na koniec automaticky pridáva aktuálny rok.
- owner_org - názov organizácie, do ktorej sa budú dáta pridávať.
- filename - názov súbora, do ktorého sa bude dátová sada ukladať lokálne na servery. Nemal by obsahovať diakritiku a končiť by sa mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.
- extension - formát súboru dátovej sady.
- table_head - pole názvov stĺpcov tabuľky.
- senzor1-name - názov senzoru 1.
- senzor1-iri - IRI adresa pre identifikáciu senzoru 1.
- senzor1-lat - súradnice zemepisnej šírky umiestnenia senzoru 1.
- senzor1-long - súradnice zemepisnej dĺžky umiestnenia senzoru 1.
- senzor2-name - názov senzoru 2.

4. miesto systému, ktoré je otvorené pre komunikáciu s vonkajším svetom

Tabuľka 4.4: Argumenty pre ovládanie skriptu teplota.py

skrátený argument	dlhý argument	popis
-h	–help	vypíše pomoc o spustení programu
-sy ROK	–start-year ROK	import od roku
-sm MESIAC	–start-month MESIAC	import od mesiaca
-ey ROK	–end-year ROK	import do roku
-em MESIAC	–end-month MESIAC	import do mesiaca
–head/–no-head	–head/–no-head	s/bez zapísania hlavičky CSV súboru

- senzor2-iri - IRI adresa pre identifikáciu senzoru 2.
- senzor2-lat - súradnice zemepisnej šírky umiestnenia senzoru 2.
- senzor2-long - súradnice zemepisnej dĺžky umiestnenia senzoru 2.

Ukážka konfiguračného súboru aj s príkladom hodnôt je v prílohe D

Pre ovládanie skriptu má užívateľ k dispozícii argumenty viď. tabuľka 4.4. Pomocou argumentov `-start-year`, `-start-month` a `-end-year`, `-end-month` sa definujú počiatočné a koncové rozsahy dát, ktoré sa budú vkladať. Argumenty `-head` a `-no-head` slúžia na určenie, či budú vkladané dáta obsahovať hlavičku alebo nie. Tak, ako pri dátach o elektronabíjačkách, očakávané je používanie `-head` pre vkladanie nových dátových sád a `-no-head` pre aktualizovanie už existujúcich.

Po spustení skriptu sa najskôr vykoná kontrola argumentov. Pokiaľ niektorý z nich chýba alebo obsahuje chybnú hodnotu (napríklad koniec intervalu je menší ako začiatok), skript sa ukončí a zaloguje chybu. V opačnom prípade pokračuje ďalej a vytvorí zálohu všetkých súborov s dátami pre roky, ktoré sa budú upravovať.

Hlavná časť programu sa vykonáva vo `for` cykle, ktorý nasleduje ďalej. Ten slúži ako iterátor počas všetkých rokov obdobia, ktoré bolo zadané na vstupe. Stručne sa dá jeho funkcia pre každý rok zhrnúť ako - získanie a upravenie dát do správnej podoby, uloženie do súboru a nahratie na server. Podobne ako v skripte pre dátovú sadu o využívaní elektronabíjačiek aj tu sa dáta získavajú a spracovávajú po jednotlivých mesiacoch vo funkcii `month_year_iter()`. Samotná funkcia je však omnoho jednoduchšia. Dôvodom je najmä to, že v tejto dátovej sade netreba tri rôzne dotazy na získanie všetkých dát zo spracovávaného mesiaca, ale stačí iba jeden. Dáta sa získavajú pomocou funkcie `get_data()`, kde sa vykoná HTTP metóda typu GET. Url adresa je špecifikovaná v konfiguračnom súbore a pridávajú sa k nej argumenty pre špecifikáciu požadovaného roku a mesiaca. Celkový dopyt pre získanie dát teda vyzerá nasledovne: `url + '&R=' + str(year) + '&M=' + str(month)`. V prípade, že sa dáta získať nepodarí, dôvod zlyhania sa zaznamená do logovacieho súboru.

Získané dáta pokračujú na prvotné spracovanie do funkcie `clean_data()`. Tu prebehne odstránenie nepotrebností a rozdelenie reťazcov do poľa pre jednoduchšie spracovanie. Finálne úpravy pred zapísaním dát do súboru sa vykonávajú vo funkcii `prepare_data()`. V zdrojovom dátovom súbore obsahuje každý záznam jednu alebo dve hodnoty s teplotou podľa počtu dostupných teplotných senzorov v danom čase (štandardne sú dva, ale v prvom mesiaci po spustení bol dostupný iba jeden). Teoreticky môže nastať situácia, že by v budúcnosti pribudli ďalšie senzory. Rozhodli sme sa preto rozdeliť jednotlivé záznamy a na každom riadku tak bude hodnota iba z jedného z nich. Ak by takáto situácia nastala, bude potrebné modifikovať funkciu na tomto mieste. Výstupom tejto funkcie je pole s dátami pripravenými na publikovanie, ktoré sa vracajú do hlavného `for` cyklu.

Skript skontroluje, či už na disku existuje súbor pre spracovávaný rok, a ak áno, tak ho otvorí na zápis, inak sa tento súbor vytvorí. Predtým ako prebehne zápis do súboru sa ešte skontroluje podmienka na zápis hlavičky tabuľky. V tomto momente sú dáta pripravené na zápis do CKANu. Postup je podobný ako v predchádzajúcom skripte, najskôr sa skontroluje či už v systéme existuje dátová sada pre dané obdobie, ktorá by sa aktualizovala, alebo či je potrebné vytvoriť novú. Na to sa využíva funkcia `ckan_post_request()`, ktorá slúži ako wrapper pre všetky požiadavky typu POST pre CKAN. Funkcia vracia `0`

pokiaľ skončili úspešne, alebo `1` ak nastala chyba. V niektorých prípadoch je potrebné výstup ďalej analyzovať a vtedy sa vracia celý objekt požiadavky. Toto sa deje pri API akciách `package_show`, alebo `package_create`. Pokiaľ funkcia vrátila hodnotu `1`, alebo objekt obsahuje hodnotu `deleted` v poli `['result']['state']`, znamená to, že je potrebné dátovú sadu vytvoriť. Inak sa bude v poli `['result']['id']` nachádzať identifikátor dátovej sady, ktorú budeme upravovať. Pre vytvorenie a získanie tohto identifikátora je potrebné zavolať API metódu `package_create` s vhodnými dátami:

- name - názov dátovej sady bez medzier a diakritiky.
- title - ľudsky čitateľný názov, môže obsahovať medzery a diakritiku.
- private - `TRUE`/`FALSE` hodnota, či má byť hneď zverejnená.
- url - akýkoľvek reťazec.
- owner_org - meno organizácie, pod ktorú dátová sada patrí.

Dátová sada s nameranými teplotami je relatívne veľká. Celkový počet záznamov za celý rok, keď sa teplota meria v 15-minútových intervaloch, presahuje číslo tridsať tisíc. Aj keď by mal CKAN zvládať takéto množstvo, narazili sme na problém s rozšírením DataStore, ktoré slúži na zobrazovanie náhľadov dát v CKANe. Aj keď boli všetky dáta nahrané v systéme a po stiahnutí bol dokument kompletný, z nejakého dôvodu v náhľade chýbal určitý počet riadkov, ktorý sa menil nedeterministicky. Komunikovali sme tento problém s vývojármi CKANu[5], ale nepodarilo sa im ho zreprodukovať. Nedokážeme povedať, či bol problém s našou inštanciou CKANu, alebo bol problém niekde inde. Riešenie asi nie je úplne ideálne, ale CKAN zobrazuje správne hodnoty. Počas nahrávania nových dát do CKANu si ukladáme identifikátor dátových sád, ktoré boli upravené do lokálneho súboru. Po skončení nahrávania voláme pre všetky identifikátory v ňom skript, ktorý v Docker kontajnéri zavolá program `paster datapusher submit \$id`, čím sa spustí opätovné nahranie dátovej sady do DataStore a dáta budú kompletné.

5. https://github.com/ckan/ckan/issues/5352

Tabuľka 4.5: Návratové hodnoty skriptu teplota.py

hodnota	konštanta	popis
1	EXIT_REQUEST_ERROR	nastala chyba počas komunikácie so servrom
2	EXIT_ROLLBACK_SUCCESS	nepodarilo sa nahrať dáta, dátová sada bola vrátená do pôvodného stavu
3	EXIT_ROLLBACK_ERROR	nepodarilo sa nahrať dáta ani vrátiť dátovú sadu do pôvodného stavu
5	EXIT_MISSING_CONFIG	chýbajúci konfiguračný súbor
6	EXIT_ARGUMENT_ERROR	zadaný nevhodný argument pri spustení programu
7	EXIT_FILE_ERROR	chyba pri otváraní súboru

Z POST požiadavky, ktorou sme získali identifikátor dátovej sady, sa dozvieme aj informácie o zdrojoch v nej. Ak sa v poli `['result']['resources']` žiaden zdroj nenachádza je potrebné vytvoriť nový zdroj. V prípade, že sa tam nejaký záznam nachádza, porovnáme meno zdroja (`resource['name']`) s názvom spracovávaného súboru. Ak sa zhodujú, tak stačí zobrať identifikátor z `resource['id']` a využiť ho na aktualizáciu konkrétneho zdroja. Pokiaľ žiadna zhoda nenastane, musíme vytvoriť nový zdroj. Pre vytvorenie/aktualizovanie zdroja opäť použijeme funkciu `post_ckan_request()` so zavolaním vhodnej API metódy. V prípade, že všetko prebehlo v poriadku a skript sa neukončil kvôli inému problému, tak sa vymažú zálohy súborov a zapíšu sa identifikátory aktualizovaných dátových sád do súboru.

Po úspešnom nahratí údajov na portál otvorených dát sa vymažú zálohované dáta a ukončí sa vykonávanie skriptu. Návratová hodnota

v takomto prípade je `0`. Pokiaľ nastane problém, skript vráti nenulovú hodnotu. Jej významy sú popísané v tabuľke 4.5.

4.3 Dátová sada o výsledkoch hlasovania mestského zastupiteľstva

Poslednou dátovou sadou, ktorou sme sa zaoberali, boli dáta o výsledkoch hlasovania metského zastupiteľstva. Táto implementácia bola zo všetkých troch najjednoduchšia. Bolo to tak najmä preto, že dáta, ktoré mesto produkuje, nebolo potrebné vôbec upravovať, ale dali sa zverejniť v pôvodnom stave. Ďalším rozdielom je, že zverejnené dáta sú vo formáte XML.

4.3.1 Zdroj dát

Dáta sa získavajú z mestskej aplikácie, ale žiadnym spôsobom sa nekontroluje ich obsah. Preto je dôležité, aby mesto vedelo, čo zverejňuje, a aby sa tam nedostali žiadne citlivé dáta.

4.3.2 Formát očakávaných dát na vstupe

V podstate jedinou požiadavkou je, aby boli na vstupe dáta vo formáte XML a ich obsah sa v čase nezmenil, ale stále obsahoval dáta o výsledkoch hlasovania mestského zastupiteľstva. V prílohe G je zjednodušená ukážka očakávaného vstupu.

4.3.3 Implementácia

Pre správnu funkčnosť je opäť potrebné mať vytvorený konfiguračný súbor `config.toml` nachádzajúci sa v priečinku, kde je uložený aj spúšťaný skript a vyplnené nasledujúce položky:

- url - adresa na server, na ktorom sa prevádzkuje systém CKAN.
- url_api - adresa na endpoint[6] API rozhrania serveru so systémom CKAN.

6. miesto systému, ktoré je otvorené pre komunikáciu s vonkajším svetom

- apikey - kľúč pre autorizáciu činnosti v prostredí CKAN, typicky nie je potrebná pokiaľ sa jedná o operácie čítania, ale nevyhnutná pokiaľ sa jedná o akúkoľvek modifikáciu dát v systéme.
- request_url - url adresa, z ktorej sa získavajú dáta.
- package - systémový názov dátovej sady. Nemal by obsahovať diakritiku a končiť by sa mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.
- package_name - ľudský názov dátovej sady, aký bude vidieť v portále CKAN. Končiť by sa mal medzerou, pretože skript na koniec automaticky pridáva aktuálny rok.
- owner_org - názov organizácie, do ktorej sa budú dáta pridávať.
- filename - názov súbora, do ktorého sa bude dátová sada ukladať lokálne na serveri. Nemal by obsahovať diakritiku a končiť by sa mal pomlčkou, pretože skript na koniec automaticky pridáva aktuálny rok.
- extension - formát súboru dátovej sady.

Príklad konfiguračného súboru s reálnymi hodnotami je dostupný v prílohe F.

V tabuľke 4.6 sú zobrazené argumenty, ktoré sa používajú pre ovládanie skriptu. Webová stránka mesta, odkiaľ sú čerpané dáta, používa na označenie jednotlivých hlasovaní identifikačné čísla. Tie však nie sú nikde zverejnené a je potrebné preskúmať zdrojový kód webovej aplikácie [7]. Identifikačné číslo júnovej schôdze je `43` a zvyšuje sa o jedna s každou ďalšou schôdzou. Z tohto dôvodu sa aj používajú ako argumenty pre spúšťanie skriptu. Zamýšľané použitie je, že pre prvotný import dát sa použije rozsah viacerých identifikačných čísel a neskôr sa už budú pridávať iba po jednom a teda sa počiatočná aj koncová hodnota rozsahu budú zhodovať.

Celý beh skriptu sa odohráva vo `for` cykle, kde jedna iterácia obslúži kompletný import jednej dátovej sady. Vo funkcii `get_data()` sa najskôr stiahnu požadované dáta podľa identifikačného čísla, ktoré je

7. `https://data.zdarns.cz/zm/`

Tabuľka 4.6: Argumenty pre ovládanie skriptu uredni-deska.py

skrátený argument	dlhý argument	popis
-h	–help	vypíše pomoc o spustení programu
-sid ID	–start-id ID	import od id
-eid ID	–end-id ID	import do id

zadané ako parameter funkcie. Dátovú sadu nie je potrebné upravovať, pretože neobsahuje žiadne dáta, ktoré by sa nemohli zverejniť. Z tohto dôvodu sa v tejto funkcií okamžite ukladajú do súboru.

Algoritmus pokračuje veľmi podobne ako v predchádzajúcich skriptoch. Najskôr sa skontroluje, či v CKANe existuje dátová sada s názvom aktuálne spracovávanej. Pre volanie API metódy `package_show` sa využije funkcia `ckan_post_request()`. Ak sa tam nachádza, funkcia vráti objekt, z ktorého získame jej identifikátor (uložený v poli `['result']['id']`). Pokiaľ volanie skončí chybou (`EXIT\_REQUEST\_ERROR`) alebo vrátený objekt obsahuje informáciu o tom, že dátová sada bola vymazaná (v poli `['result']['state']` je hodnota `deleted`), je potrebné ju vytvoriť. To dosiahneme volaním metódy `package_create` s vhodnými dátami:

- name - názov dátovej sady, bez medzier a diakritiky.
- title - ľudsky čitateľný názov, môže obsahovať medzery a diakritiku.
- private - `TRUE`/`FALSE` hodnota, či má byť sada hneď zverejnená.
- url - akýkoľvek reťazec.
- owner_org - meno organizácie, pod ktorú dátová sada patrí.

Výsledkom úspešného volania je opäť objekt, ktorý obsahuje identifikátor dátovej sady uložený v poli `['result']['id']`. Okrem identifikátora získame z tohto objektu aj informáciu o zdrojoch, ktoré patria k tejto dátovej sade. Pokiaľ sa niektorý prvok z poľa

Tabuľka 4.7: Návratové hodnoty skriptu uredni-deska.py

hodnota	konštanta	popis
1	EXIT_REQUEST_ERROR	nastala chyba počas komunikácie so servrom
2	EXIT_ROLLBACK_SUCCESS	nepodarilo sa nahrať dáta, dátová sada bola vrátená do pôvodného stavu
3	EXIT_ROLLBACK_ERROR	nepodarilo sa nahrať dáta ani vrátiť dátovú sadu do pôvodného stavu
4	EXIT_NOTHING_TO_UPLOAD	neexistujú dáta so zadaným identifikátorom
5	EXIT_MISSING_CONFIG	chýbajúci konfiguračný súbor
6	EXIT_ARGUMENT_ERROR	zadaný nevhodný argument pri spustení programu

`['result']['resources']['name']` zhoduje s názvom aktuálne spracovávaného súboru, uložíme si identifikátor zdroja `['result']['resources']['id']` a využijeme ho na jeho aktualizáciu novými dátami pomocou API metódy `resource_update`. Ak sa nenájde žiadna zhoda pomocou metódy `resource_create()`, vytvoríme nový. Pokiaľ všetko prebehne korektne, dáta sa úspešne naimportovali a skript sa ukončí s návratovou hodnotou `0`. Pokiaľ nastane problém skript vráti nenulovú hodnotu, jej významy sú popísané v tabuľke 4.7.

4.4 Skript pre plánované spúšťanie importu dát

V podkapitole 3.2 sme popísali, ako sa používa utilita CRON pre pravidelné spúšťanie skriptov. V tejto podkapitole bude popísaná im-

plementácia skriptu `job.sh`, ktorý slúži na spúšťanie konkrétnych skriptov pre jednotlivé dátové sady.

Očakávaná súborová štruktúra je, že skript `job.sh` sa bude spúšťať z priečinka, ktorý bude obsahovať samostatný priečinok pre každú dátovú sadu. Skript je implementovaný v jazyku `bash`. Pokiaľ aplikácia CRON vyhodnotí, že je polnoc prvého dňa v mesiaci, spustí sa skript `job.sh`. Podľa aktuálneho dátumu získa mesiac a rok, ktorý si uloží do premennej. Postupne sa prepne do jednotlivých zložiek dátových sád, odkiaľ zavolá konkrétny skript. Keď skončí vykonávanie jedného, okamžite spúšťa ďalší.

Špecifická situácia je pri dátovej sade o hlasovaní mestského zastupiteľstva. Ako bolo spomenuté vyššie, na identifikáciu dátovej sady sa nepoužíva mesiac a rok, ale iba identifikátor, ktorý sa zaznamenáva do súbora `uredni-deska/.ID`. Tento súbor je potrebné vytvoriť pred prvým použitím. Mal by obsahovať identifikátor naposledy importovanej dátovej sady. Skript sa teda pokúsi stiahnuť dátovú sadu s identifikátorom o jedna vyšším ako je posledný spracovaný. Ak je toto volanie úspešné, do súboru sa zaznamená aktualizovaný identifikátor. Môže nastať situácia, že v meste boli dve hlasovania počas jedného mesiaca, preto sa skript pokúsi zopakovať ešte jednu iteráciu so zvýšeným identifikátorom. Ak bolo toto volanie úspešné, opäť sa aktualizuje záznam v súbore.

5 Vyhodnotenie prínosov pre mesto

Pokiaľ majú mestá v budúcnosti pristúpiť k automatizovanému zverejňovaniu dát, je potrebné vyhodnotiť, či to je vôbec prospešnejšie v porovnaní s manuálnym zverejňovaním. V tejto kapitole spravíme odhad nákladov pre obe alternatívy a navzájom ich porovnáme z časového aj finančného hľadiska.

5.1 Odhad nákladov pre manuálne nahrávanie dát

Pre manuálne zverejňovanie nie je potrebný technicky kvalifikovaný zamestnanec, ale zvládne to aj bežný administratívny pracovník. Jeho práca bude obnášať vykonanie všetkých krokov popísaných v 2. kapitole. Komplikované to je najmä pri dátach, ktoré nie sú v správnom formáte. Príkladom sú dáta o využívaní elektronabíjačiek. Keďže portál Evmapa neposkytuje žiadnu možnosť exportu dát, je potrebné sa tam prihlásiť, celú tabuľku skopírovať, vložiť do tabuľkového editoru, spracovať a zverejniť. Vieme si predstaviť, že by to mohlo fungovať pre priebežné zverejňovanie nových dát, ale aj tu existuje množstvo rizík, ako sú napríklad chybovosť a aktuálnosť. Manuálne spracovanie prináša zvýšené riziko zanesenia chýb do výslednej dátovej sady. Pokiaľ sa majú tieto dáta v budúcnosti používať aj v komerčnej sfére, je nutné garantovať ich kvalitu. Problém s aktuálnosťou zase spočíva v tom, že dátová sada je závislá na konkrétnych ľuďoch. Stačí keď na to poverený zamestnanec zabudne, alebo nebude v práci dátová sada nebude aktualizovaná s novými dátami. To je problém pre všetkých konzumentov dát, ktorí budú očakávať ich aktuálnosť k určitému dňu v mesiaci.

Ďalší aspekt, na ktorý sa nesmie zabudnúť, je spracovanie už existujúcich dát. V čase, keď sa mesto rozhodne, že chce spracovať nejaké dáta, neznamená to, že zamestnanec raz za mesiac spracuje nové dáta, ktoré za ten čas vznikli a tým končí jeho práca. V momente ako mu je takáto úloha zadaná, musí okrem nových dát priebežne spracovávať aj všetky, ktoré existovali už predtým. Napríklad dáta o elektronabíjačkách majú históriu od roku 2017, dáta o teplote existujú už od roku 2006.

Tabuľka 5.1: Výpočet času potrebného pre počiatočný manuálny import dát; X: odhadovaný čas pre import jednej dátovej sady (hod), Y: počet existujúcich dátových sád do júna 2020

dátová sada	X	Y	celkový počet hodín
elektronabíjačky	1	31	31
teplota	2	167	334
výsledky hlasovania	0.02	43	0,86
spolu			365,86

Časová náročnosť na zverejnenie dátovej sady sa nedá generalizovať, líši sa v závislosti od toho v akom stave sú dáta, ktoré zverejňujeme. Čas potrebný na zverejnenie jednej dátovej sady sa môže pohybovať od jednotiek minút, až po hodiny v prípade tých zložitejších. Z troch spracovaných dátových sád je na manuálne zverejňovanie najjednoduchšia dátová sada o hlasovaní mestského zastupiteľstva. Keďže neobsahuje žiadne dáta, ktoré by bolo potrebné pred zverejnením odstrániť, jej manuálne spracovanie trvá menej ako jednu minútu. Zložitejšie to je v prípade dát o teplote, zdrojové dáta sú v celkom dobrom stave, ale je ich potrebné upraviť, aby spĺňali požiadavky Ministerstva vnútra ČR na otvorené dáta. Tu už môže dochádzať aj k chybám, preto je potrebné výstup skontrolovať. Pre predstavu, ide o úpravy typu prepísania dátumu z jedného formátu (`2020-03-01 00:00:00`) do druhého (`2020-03-01T00:00:00`), pretože tak to je definované v otvorených formálnych normách a konzument dát bude práve takýto formát očakávať. Tieto a ďalšie potrebné úpravy nie sú náročné. Problém je, že tých záznamov je takmer 5800 pre dáta v rozsahu jedného mesiaca a teda na spracovanie treba až približne 120 minút. Dáta o využití elektronabíjačiek sú zložitejšie a potrebujú ešte viac úprav, ale za jeden mesiac sa vygeneruje niekoľko stoviek záznamov, takže ich spracovanie je rýchlejšie. Aj s kontrolou dát to trvá približne 60 minút.

Pre výpočet nákladov na spracovanie všetkých existujúcich dátových sád sme stanovili vzorec

$$cena = X * Y * Z$$

Tabuľka 5.2: Výpočet nákladov potrebných pre počiatočný manuálny import dát; X: odhadovaný čas pre import jednej dátovej sady (hod), Y: počet existujúcich dátových sád do júna 2020

dátová sada	X	Y	Z	cena
elektronabíjačky	1	31	145,96 Kč	4525,03 Kč
teplota	2	167	145,96 Kč	48753,56 Kč
výsledky hlasovania	0.02	43	145,96 Kč	125,53 Kč
spolu				53404,12 Kč

kde, X je odhadovaný čas na import jednej dátovej sady v hodinách, Y je počet existujúcich dátových sád pre danú problematiku do konca júna 2020 a Z je hodinová sadzba zamestnanca. Z tabuľky 5.1 vyplýva, že na nahratie všetkých doteraz existujúcich dát manuálne by sme potrebovali odhadom takmer 366 človekohodín. Hodinová sadzba programátora (4110 Všeobecní administratívni pracovníci) z mediánu hrubej mesačnej mzdy je v ČR 145,97 Kč za hodinu [25]. Odhad ceny takejto práce je 53404,12 Kč, viď. tabuľku 5.2.

5.2 Odhad nákladov pre automatické nahrávanie dát

Hlavnou výhodou automatického zverejňovania dát v porovnaní s manuálnym je nižšia chybovosť a vyššia aktuálnosť. Pokiaľ sa skripty dobre nachystajú, otestujú a máme istotu, že vstupné dáta sa nebudú meniť, nemalo by dochádzať k chybám. Skripty je možné spúšťať plánovane, takže nie je potrebný zamestnanec, ktorý by bol zodpovedný za ich spúšťanie, ale o všetko sa postará systém, pokiaľ je vhodne nastavený. To zabezpečí, že zverejnené dáta sú vždy tie najaktuálnejšie.

Vytváranie skriptov nie je triviálna úloha, ale po naštudovaní danej problematiky sa to dá zvládnuť relatívne rýchlo. Písanie skriptov a testovanie nám dokopy trvalo asi 25 človekohodín, ďalších 15 sme potrebovali na pochopenie ako funguje CKAN. To bolo zapríčinené najmä nedostatkom informácií a čerpať sa dalo iba z oficiálnej dokumentácie. Tento časový odhad je bez chápania širšieho kontextu

Tabuľka 5.3: Výpočet nákladov tvorbu automatických skriptov pre import dát

činnosť	počet hodín	hodinová sadzba	cena
analýza	15	260,07 Kč	3901,05 Kč
implementácia	25	260,07 Kč	6501,75 Kč
spolu			10402,8 Kč

problematiky otvorených dát. Po prečítaní tejto publikácie sa môže tento čas pre priemerného programátora ešte skrátiť, pretože tu nájde väčšinu informácií, ktoré k tomu bude potrebovať. V tabuľke 5.3 vidíme výpočet nákladov na automatické spracovanie. Ak vynásobíme 40 hodín s hodinovou sadzbou programátora (3511 Technici prevádzky ICT, technici programátori) z mediánu hrubej mesačnej mzdy v ČR (260,07 Kč za hodinu [25]) dostaneme cenu práce 10402,8 Kč. V porovnaní s manuálnym nahrávaním je to lacnejšie až o 80,52%. K tomu treba ešte zdôrazniť fakt, že toto je len úspora pri počiatočnom spracovaní už existujúcich dát a ďalej sa bude zvyšovať v čase. Náklady na spracovanie ďalších mesiacov sú prakticky nulové, ale za manuálne nahrávanie dát bude musieť mesto platiť aj naďalej.

6 Záver

Cieľom práce bolo navrhnúť automatizovaný proces publikácie dát do portálu CKAN pre aspoň tri dátové sady. Ďalšou úlohou bolo zhodnotiť časovú a finančnú úsporu pri využití automatického publikovania oproti manuálnemu.

Diplomová práca vznikala v spolupráci s mestom Žďár nad Sázavou. Okrem implementačnej časti bola dôležitá aj komunikácia so zamestnancami mesta, s ktorými sme museli dohodnúť ako a k akým dátam môžeme dostať prístup. Skúsenosti získané z komunikácie s mestom sú zaznamenané v práci a môžu pomôcť ďalším projektom zaoberajúcim sa publikáciou otvorených dát. Konkrétne dátové sady, ktoré boli v pláne zverejniť sa niekoľkokrát zmenili vďaka informáciám, ktoré sme sa priebežne dozvedali. Priebeh bol tiež ovplyvnený pandémiou COVID-19, ktorá spomalila komunikáciu s mestom. Pôvodné plány na pravidelné stretnutia v Žďári nad Sázavou sa nakoniec nepodarilo uskutočniť a vycestovali sme tam len raz pred vypuknutím pandémie.

Napriek týmto komplikáciám si myslíme, že sa zadanie práce podarilo naplniť. Podarilo sa nám zverejniť dátové sady o využívaní elektronabíjacích staníc od roku 2017, dátové sady o nameranej teplote v meste od roku 2006 a výsledky hlasovania mestského zastupiteľstva od decembra 2014. Pre každú z týchto dátových sád vznikol skript, ktorý umožňuje hromadne publikovať doterajšie vytvorené dáta, ale aj priebežné publikovanie raz za mesiac. Celý algoritmus je zdokumentovaný v tejto práci aj s vysvetleniami, takže v prípade záujmu je možné pokračovať v rozvíjaní skriptov aj do budúcnosti. V skripte, ktorý publikuje dáta z portálu Evmapa, vidíme potenciál pre jeho použitie aj v iných mestách, ktoré používajú tento portál na administráciu elektronabíjacích staníc. Zdrojové kódy sú súčasťou diplomovej práce (zoznam odovzdaných súborov je v prílohe A) a zároveň sú zverejnené na portále GitHub [1].

Na úrovni mesta sa ukázalo, že automatická publikácia dát je jednoznačne výhodnejšia z časového aj finančného hľadiska. Podarilo sa nám automatizovať tri dátové sady, avšak mesto produkuje dát omnoho viac. Dá sa pokračovať v ich identifikovaní pre ďalšiu automa-

1. `https://github.com/hrnciar/diplomova-praca`

tizáciu, prípadne pomôcť mestu pri zadávaní verejných obstarávaní tak, aby každá ďalšia zákazka tvoriaca dáta mala povinnosť vytvoriť aj vlastné API rozhranie pre prístup k nim.

Na Ministerstve vnútra ČR postupne vznikajú otvorené formálne normy, ktoré sa snažia štandardizovať, ako by mali výsledné dátové sady vyzerať. Pre dáta, ktoré sme zverejnili zatiaľ neboli dostupné. Práve naša práca by mohla poslúžiť v budúcnosti ako návrh, na základe ktorého môžu otvorené formálne normy vzniknúť.

Z tejto práce budú vychádzať aj ďalší študenti v Laboratóriu servisných systémov FI MUNI zapojení do práce pre Žďár nad Sázavou, alebo v iných mestách a obciach.

A Elektronické prílohy

- elektronabijecky/
  - backup - priečinok pre ukladanie dátových sád
  - config.toml.template - vzor konfiuračného súboru
  - elektronabijecky.py - zdrojový kód skriptu
  - requirements.txt - súbor so zoznamom požiadaviek
- teplota/
  - backup - priečinok pre ukladanie dátových sád
  - config.toml.template - vzor konfiguračného súboru
  - requirements.txt - súbor so zoznamom požiadaviek
  - teplota.py - zdrojový kód skriptu
- uredni-deska/
  - backup - priečinok pre ukladanie dátových sád
  - config.toml.template - vzor konfiuračného súboru
  - requirements.txt - súbor so zoznamom požiadaviek
  - uredni-deska.py - zdrojový kód skriptu
- .gitignore - zoznam súborov ignorovaných gitom
- README.md
- job.sh - skript pre automatické ovládanie skriptov

B Konfiguračný súbor pre elektronabijecky.py

```
url = "http://sc02.fi.muni.cz/"
url_api = "http://sc02.fi.muni.cz/api/action/"
apikey = ''
post_login_url = 'https://www.evmapy.cz/users_login.php'
request_url = 'https://www.evmapy.cz/muj-ucet?action=stats'
package = 'vyuziti-elektronabijecek-'
package_name = 'Využití elektronabíječek v roce '
owner_org = 'elektronabijecky-zdar-nad-sazavou'

filename = 'backup/elektronabijecky_'
extension = '.csv'

table_head = [
    'nabíjecí_stanice', 'nabíjení_interval_začátek_datum_a_čas',
    'nabíjení_interval_konec_datum_a_čas', 'spotřeba_hodnota',
    'spotřeba_jednotka'
]

resource_iri = 'https://data.zdarns.cz/zdroj/'

[station_dict]
319 = 'centrální-parkoviště'
351 = 'stará-radnice'

[socket_dict]
343 = 'konektor-2-mennekes-22kw'
344 = 'zasuvka-230v'
391 = 'stojan-1'

[payload]
em = ''
ps = ''
```

C Vzor vstupných dát pre skript elektronabijecky.py

```
<table>
...
</table>
<h1>Detail</h1>
<table>
    <!-- prvý riadok tabuľky sa zahodí -->
    <tr>
        <td>ID</td>
        <td>Čas od - do</td>
        <td>Aktivace</td>
        <td>Tarif</td>
        <td>Spotřeba</td>
        <td>Provozovatel</td>
        <td>Evmapa<br />provize</td>
        <td>Celkem<br />tržba</td>
    </tr>
    <tr>
        <td>elektřina</td>
        <td>poplatek za<br />využ. stanice</td>
        <td>celkem</td>
    </tr>
    <tr>
        <td><a><b>8082</b></a></td>
        <!-- tento stĺpec sa zachová -->
        <td>23.08.2019 19:26 - 19:53</td>
        <td>  web</td>
        <td>zdarma</td>
        <td>1.516 kWh</td> <!-- tento stĺpec sa zachová -->
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
```

```
    </tr>
    <tr>
        <td><a><b>8082</b></a></td>
        <!-- tento stĺpec sa zachová -->
        <td>23.08.2019 14:26 - 14:54</td>
        <td>  web</td>
        <td>zdarma</td>
        <td>1.546 kWh</td> <!-- tento stĺpec sa zachová -->
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
    </tr>
    <!-- posledný riadok tabuľky sa zahodí -->
    <tr>
        <td>Celkem CZK</td>
        <td>3.062 kWh</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
        <td>0.00 CZK</td>
    </tr>
</table>
```

D Konfiguračný súbor pre teplota.py

```
url = "http://sc02.fi.muni.cz/"
url_api = "http://sc02.fi.muni.cz/api/action/"
apikey = ''
request_url =
    'https://data.zdarns.cz/temp.asp?P=XXXXXXXX&F=T'
resource_iri = 'https://data.zdarns.cz/zdroj/'
package = 'historie-teploty-'
package_name = 'Historie teploty v roce '
owner_org - 'mestsky-urad'

filename = 'backup/teplota_'
extension = '.csv'

table_head = [
    'datum a čas měření', 'čidlo', 'teplota',
    'jednotka', 'latitude', 'longitude'
]

senzor1-name = 'TME ethernetový teploměr p4602'
senzor1-iri = 'https://data.zdarns.cz/zdroj/tme-p4602-west'
senzor1-lat = 49.56520148
senzor1-long = 15.93545973

senzor2-name = 'TME ethernetový teploměr p4602'
senzor2-iri = 'https://data.zdarns.cz/zdroj/tme-p4602-east'
senzor2-lat = 49.56464828
senzor2-long = 15.93555093
```

E Vzor vstupných dát pre skript teplota.py

```
ts;t1;t2;sonda
2020-03-01 00:00:00;4;4;0
2020-03-01 00:15:00;3.8;4;1
2020-03-01 00:30:00;4;4;0
2020-03-01 00:45:00;4.1;4.3;0
2020-03-01 01:00:00;3.9;4.4;0
2020-03-01 01:15:00;3.9;4;0
2020-03-01 01:30:00;3.8;3.9;0
2020-03-01 01:45:00;3.6;4;1
2020-03-01 02:00:00;3.7;3.8;1
2020-03-01 02:15:00;3.9;4;1
2020-03-01 02:30:00;4.1;4.1;0
2020-03-01 02:45:00;4.1;4.3;0
2020-03-01 03:00:00;4;4.3;0
2020-03-01 03:15:00;4;4;0
2020-03-01 03:30:00;3.9;4;0
2020-03-01 03:45:00;3.8;3.8;0
2020-03-01 04:00:00;3.9;4;1
...
2020-03-31 21:00:00;0.5;0.5;0
2020-03-31 21:15:00;0.5;0.6;0
2020-03-31 21:30:00;0.3;0.6;0
2020-03-31 21:45:00;-0.1;0;0
2020-03-31 22:00:00;-0.1;0.1;0
2020-03-31 22:15:00;-0.1;0;0
2020-03-31 22:30:00;-0.2;-0.2;0
2020-03-31 22:45:00;-0.8;-0.5;0
2020-03-31 23:00:00;-1.5;-1.4;0
2020-03-31 23:15:00;-1.3;-1.1;0
2020-03-31 23:30:00;-1.5;-1.4;1
2020-03-31 23:45:00;-1.5;-1.2;0
```

F Konfiguračný súbor pre uredni-deska.py

```
url = "http://sc02.fi.muni.cz/"
url_api = "http://sc02.fi.muni.cz/api/action/"
apikey = ''
request_url = 'http://deska.zdarns.cz/zm/export.asp'
package = 'hlasovani-mestskeho-zastupitelstvi-'
package_name =
    'Výsledky hlasování městského zastupitelství '
owner_org = 'mestsky-urad'

filename = 'hlasovani_'
extension = '.xml'
```

G Vzor vstupných dát pre skript uredni-deska.py


```
<?xml version="1.0" encoding="UTF-8"?>
<schuze>
    <datum>18.6.2020</datum>
    <nazev>12. zasedání zastupitelstva města</nazev>
    <bod>
        <bod_id>483</bod_id>
        <cj>12/2020/OS/1</cj>
        <cislo>1.</cislo>
        <nazev>Vyhodnocení dotačního programu... </nazev>
        <usneseni>
            <id>832</id>
            <cislo></cislo>
            <text>
                Zastupitelstvo města po projednání schvaluje...
            </text>
            <hlasovani>
                <pro>
                    <clen>Mgr. Tomáš Augustýn</clen>
                    <clen>MUDr. Romana Bělohlávková</clen>
                    ...
                    <clen>Ing. Rudolf Voráček</clen>
                </pro>
                <proti><clen></clen></proti>
                <zdrzel_se><clen></clen></zdrzel_se>
                <nehlasoval>
                    <clen>Mgr. Jaromír Brychta</clen>
                </nehlasoval>
                <omluven><clen></clen></omluven>
                <nepritomen><clen></clen></nepritomen>
            </hlasovani>
        </usneseni>
    </bod>
</schuze>
```

Bibliografia

1. *Smernica Európskeho parlamentu a Rady (EÚ) 2019/1024* [online]. 2019 [cit. 2020-06-26]. Dostupné z: https://eur-lex.europa.eu/legal-content/SK/TXT/HTML/?uri=CELEX:32019L1024.
2. STIMMEL, Carol L. *Building smart cities: Analytics, ICT, and Design Thinking*. 1st ed. New York: Auerbach Publications, 2015. ISBN 9780429068140.
3. *Open Data* [online]. 2020 [cit. 2020-04-10]. Dostupné z: https://ec.europa.eu/digital-single-market/en/open-data.
4. *EU Open Data Portal* [online]. 2020 [cit. 2020-04-13]. Dostupné z: https://data.europa.eu/euodp/en/home.
5. *Smart Cities - Smart Living* [online]. 2019 [cit. 2020-07-03]. Dostupné z: https://ec.europa.eu/digital-single-market/en/policies/smart-cities.
6. *Smart Cities - Smart Living* [online]. 2019 [cit. 2020-04-18]. Dostupné z: https://ec.europa.eu/digital-single-market/en/smart-cities.
7. *The European Green Deal* [online]. 2020 [cit. 2020-07-03]. Dostupné z: https://www.europarl.europa.eu/doceo/document/TA-9-2020-0005_EN.html.
8. *World Urbanisation Prospects* [online]. New York: United Nations, 2014 [cit. 2020-04-06]. Dostupné z: https://population.un.org/wup/Publications/Files/WUP2014-Report.pdf.
9. AUTOROV, Kolektív. *Strategický rámec Svazu měst a obcí v oblasti Smart City: analytická část* [online]. 2019 [cit. 2020-04-25]. Dostupné z: http://prosperujiciobecbudoucnosti.cz/wp-content/uploads/2020/03/Strategicky-ramec-Svazu-mest-a-obci-v-oblasti-Smart-City_analyticka-cast.pdf.
10. AUTOROV, Kolektív. *SMART Česko: základní teze* [online]. 2019 [cit. 2020-04-24]. Dostupné z: http://prosperujiciobecbudoucnosti.cz/wp-content/uploads/2020/03/SMART-Cesko_zakladni-teze.pdf.

11. *Metodika Smart Cities: Metodika pro přípravu a realizaci konceptu Smart Cities na úrovni měst, obcí a regionů* [online]. 2018 [cit. 2020-04-19]. Dostupné z: https://mmr.cz/getmedia/f76636e0-88ad-40f9-8e27-cbb774ea7caf/Metodika_Smart_Cities.pdf.aspx?ext=.pdf.

12. BAČOVSKÝ, Ing. Michal. *Open Definition 2.1* [online]. 2019 [cit. 2020-04-26]. Dostupné z: https://opendefinition.org/od/2.1/en/.

13. *Ten Principles For Opening Up Government Information* [online]. 2010 [cit. 2020-04-26]. Dostupné z: https://sunlightfoundation.com/policy/documents/ten-open-data-principles/.

14. *Otevřená data v ČR: Portál pro poskytovatele* [online]. 2020 [cit. 2020-04-13]. Dostupné z: https://opendata.gov.cz/.

15. *Národní katalog otevřených dat* (*NKOD*) [online]. 2018 [cit. 2020-04-13]. Dostupné z: https://opendata.gov.cz/nastroj:narodni-katalog-otevrenych-dat.

16. *Linked Data Glossary* [online]. 2013 [cit. 2020-04-13]. Dostupné z: https://dvcs.w3.org/hg/gld/raw-file/default/glossary/index.html#x5-star-linked-open-data.

17. *Národní katalog otevřených dat* (*NKOD*) [online]. 2020 [cit. 2020-04-13]. Dostupné z: https://opendata.gov.cz/informace:stupn%C4%9B-otev%C5%99enosti-datov%C3%BDch-sad.

18. READ, David. *5 Stars Rating Algorithm* [online]. 2012 [cit. 2020-04-24]. Dostupné z: https://data.blog.gov.uk/2012/11/16/5-stars-rating-algorithm/.

19. *5-Star Open Data* [online]. 2015 [cit. 2020-07-11]. Dostupné z: https://5stardata.info/en/.

20. *Otevřené formální normy* (*OFN*) [online]. 2020 [cit. 2020-04-13]. Dostupné z: https://opendata.gov.cz/informace:stupn%C4%9B-otev%C5%99enosti-datov%C3%BDch-sad.

21. *Základní datové typy* [online]. 2020 [cit. 2020-04-13]. Dostupné z: https://ofn.gov.cz/z%C3%A1kladn%C3%AD-datov%C3%A9-typy/draft/.

22. BAČOVSKÝ, Ing. Michal. *CHYTRÝ ŽĎÁR* [online]. 2019 [cit. 2020-04-26]. Dostupné z: https://www.zdarns.cz/mesto-zdar/smart-city.

23. VAĽKO, Marek. *Výzkum Open Source Smart CIty platforem* [*online*]. 2020 [cit. 2020-04-15]. Dostupné tiež z: Dostupn%C3%A9%20z%20WWW%20%3Chttps://is.muni.cz/th/xsq6n/%3E. Diplomová práce. Masarykova univerzita, Fakulta informatiky, Brno. Vedúci práce Leonard WALLETZKÝ.

24. *Instances* [online] [cit. 2020-04-15]. Dostupné z: https://ckan.org/about/instances/.

25. *Mzdová sféra ČR - rok 2019* [online]. 2020 [cit. 2020-07-11]. Dostupné z: https://www.ispv.cz/cz/Vysledky-setreni/Archiv/2019.aspx.