|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Termostat řízený rozvrhem ze Školy online** | | |
| Filip Růžička | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2021/2022 | |

#### Poděkování

*Rád bych poděkoval panu Ing. Petru Grussmannovi za určení směru a za doporučení ně-kterých technologií*

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité   
informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové   
a umělecké škole v Opavě, Praskova 399/8.

V Opavě 31. 12. 2021

*podpis autora práce*

**ANOTACE**

V tomto projektu popisuju postup mojí práce s tvorbou scriptu pro parsování kalendáře ze Školy online a následné implementace do Home Assistantu. Poté následné nastavení samotného Home Assistanu a tvorby automatizace. Poté dokument poučuje, jak aplikaci používat a kde co doplňovat. Následně dokument popisuje, co jsem mohl vylepšit a co bych mohl přidat a všeobecně moje pocity o projektu.

**Klíčová Slova**

Home Assistant, parsování, ChromeDriver, Beautiful Soup, script

OBSAH

[Úvod 5](#_Toc92825820)

[1 Využité technologie 6](#_Toc92825821)

[1.1 Beautiful Soup 6](#_Toc92825822)

[1.2 ChromeDriver – WebDriver pro Chrome 6](#_Toc92825823)

[1.3 Home Assistant 6](#_Toc92825824)

[2 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY 7](#_Toc92825825)

[2.1 Instalace Home Assistanta 7](#_Toc92825826)

[2.2 Tvorba scriptu pro parsování ze Školy online 7](#_Toc92825827)

[2.2.1 Verifikace na Školu online 7](#_Toc92825828)

[2.2.2 Parsování kalendáře ze Školy online 8](#_Toc92825829)

[2.2.3 Zpracování dat 8](#_Toc92825830)

[2.3 Základní nastavení pro Home Assistenta 9](#_Toc92825831)

[2.4 Nastavení automatizace 10](#_Toc92825832)

[2.5 Vytvoření uživatelského rozhraní 10](#_Toc92825833)

[3 Výsledky řešení, výstupy, uživatelský manuál 11](#_Toc92825834)

[Závěr 12](#_Toc92825835)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 13](#_Toc92825836)

Úvod

Téma tohoto projektu jsem si vybral, jelikož jsem chtěl nějak pracovat s chytrou domác-ností. Poté co jsem svůj nápad předložil panu učiteli Petru Grussmannovi, navrhl mi, že bych místo tvorby komponentů pro chytrou domácnost, mohl vytvořit program, který by parsoval data ze Školy online a poté podle přítomností žáků reguloval teplotu radiátorů.

Ze začátku, než jsem s projektem začal, měl jsem v plánu vytvořit také svojí vlastní we-bovou stránku kde se bude zobrazovat aktuální teplota a také se budou nastavovat dané teploty.

Nakonec jsem se rozhodnul na doporučení pana učitele Petra Grussmanna použít program zvaný Home Assistant, který mojí práci mnohonásobně zjednodušil.

V této dokumentaci budu popisovat tvorbu scriptu pro parsování dat ze Školy online a propojení scriptu s Home Assistantem

# Využité technologie

## Beautiful Soup

Beautiful Soup je balíček Pythonu pro analýzu dokumentů HTML a XML (včetně chybného označení, tj. neuzavřených značek, pojmenovaných podle tag soup). Vytváří strom analýzy pro analyzované stránky, který lze použít k extrahování dat z HTML, což je užitečné pro web scraping

## ChromeDriver – WebDriver pro Chrome

WebDriver je open source nástroj pro automatické testování webových aplikací v mnoha prohlížečích. Poskytuje možnosti pro navigaci na webové stránky, vstup uživatele, spouštění JavaScriptu a další. ChromeDriver je samostatný server, který implementuje standard W3C WebDriver. ChromeDriver je k dispozici pro Chrome pro Android a Chrome pro stolní počítače (Mac, Linux, Windows a ChromeOS).

## Home Assistant

Home Assistant je bezplatný a open-source software pro domácí automatizaci, který je navržen jako centrální řídicí systém pro chytrá domácí zařízení se zaměřením na místní ovládání a soukromí.

# ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

## Instalace Home Assistanta

Jako první věc co jsem se rozhodl udělat je nainstalovat si službu Home Assistant (zkráceně “HA”). Toho jsem dosáhl pomocí návodu na stránkách HA přesněji v sekci Installation. Jelikož máme v plánu instalovat HA na Ubuntu přejdeme do sekce Linux.

Naše nároky jsou ale celkem vysoké proto sjedeme na stránce dolů až k instalaci HA Supervised a přejdeme na GitHubovou stránku.

Zde máme příkazy, které si budeme postupně kopírovat, poté resetujeme náš systém.

Nyní jenom zjistíme ip adresu našeho počítače a připojíme se pomocí webového prohlížeče na tu danou adresu s přidaným portem 8123.

Nyní máme nainstalovaný Home Assistant.

## Tvorba scriptu pro parsování ze Školy online

Jelikož Škola online nemá žádný API, musíme si získat data která potřebujeme.

Toho dosáhneme pomocí parsování webové stránky.

Na parsování jsem se rozhodnul použít balíček v Pythonu Beautiful Soup. Rozhodl jsem se Python použít kvůli dobré komunikaci s HA.

Pro verifikaci na Školu online jsem se naopak rozhodl použít technologii ChromeDriver. Pro mě to byla ve chvíli nejjednodušší možnost, jak se dokážu dostat přes přihlašování do Školy online.

### Verifikace na Školu online

Nejdřív budeme potřebovat si náš ChromeDriver nainstalovat.

Z oficiálních stránek ChromeDriveru si stáhneme verzi pro náš operační systém (tedy Linux), vyextrahujeme, a zkopírujeme soubor chromedriver do /usr/local/share/, /usr/local/bin/, /usr/bin/.

Poté budeme potřebovat si pro náš OS nainstalovat Google Chrome, aby měl ChromeDriver s kým komunikovat. To provedeme jednoduchým stáhnutím deb souboru s oficiálních stránek Google Chrome a následné instalaci pomocí dpkg.

Nyní jak už máme vše nainstalované, vytvoříme si náš pythonový soubor. Kde začneme psát náš script.

driver.find\_element(By.ID, 'JmenoUzivatele').send\_keys(username)

driver.find\_element(By.ID, 'HesloUzivatele').send\_keys(password)

button = driver.find\_element\_by\_id("dnn\_ctr994\_SOLLogin\_btnODeslat")

driver.execute\_script("arguments[0].click();", button)

*Kód, který za nás kliká a vypisuje data které mu budeme později posílat*

### Parsování kalendáře ze Školy online

K parsování stránky jsem použil balíček Beautiful Soup.

K tomuto nemám moc co říct proto si jenom ukážeme kousek kódu.

//kořen naší stránky který budeme parsovat

soup=BeautifulSoup(driver.page\_source,'lxml')

//vyparsování prvního řádku v kalendáři

row = soup.find('tr', class\_="RowOdd")

*Kód, který parsuje data pro následné zpracování*

### Zpracování dat

Následně náš vyparsovaný řádek zpracujeme tak že když v buňce nic nebude, vypíšeme False, a naopak jestli v něm něco bude, vypíšeme True.

for tds in row.find\_all('td', class\_="DctCell"):

if tds.text == "":

final.append(False)

else:

if tds.has\_attr('colspan'):

for x in range(int(tds['colspan'])):

final.append(True)

else:

final.append(True)

*Kód, který parsuje data pro následné zpracování*

## Základní nastavení pro Home Assistenta

Nyní si nastavíme .yaml soubory v našem HA. Jako první si ukážeme configuration.yaml, zde jsem přidal naše potřebné inputy pro nastavování teplot a uživatelského jména pro školu online a heslo.

input\_number:

tempidle:

name: TeplotaIdle

initial: 19

min: 10

max: 30

step: 1

tempactive:

name: TeplotaActive

initial: 22

min: 10

max: 30

step: 1

*Příklad: přidání inputu pro zadání teplot*

Následně jsem přidal nasimulovaný termostat, se kterým budu pracovat, a také senzor ke kterému bude později naslouchat.

climate:

- platform: generic\_thermostat

name: Ucebna

heater: switch.ucebna\_heater

target\_sensor: sensor.ucebna

*Příklad: přidání nasimulovaného termostatu*

sensor:

- platform: file

name: temperature

file\_path: /config/python/ofile.csv

value\_template: "{{ value\_json.temperature }}"

unit\_of\_measurement: "°C"

*Příklad senzoru*

A poté jsem nastavil aby v HA bylo možné číst externě vytvořené soubory.

homeassistant:

allowlist\_external\_dirs:

- /config/python

- /config

*Nastavení pro čtení externě vytvořených souborů*

## Nastavení automatizace

Nyní potřebujeme automatizovat náš HA ať nám poskytuje naše potřebná data z inputu a aby následně četl naše data z pythonu.

alias: Posli data

description: ""

trigger:

- minutes: /30

platform: time\_pattern

condition: []

action:

- service: notify.filenotify

data\_template:

message:

"{{ states.input\_text.username }};{{ states.input\_text.password }};{{

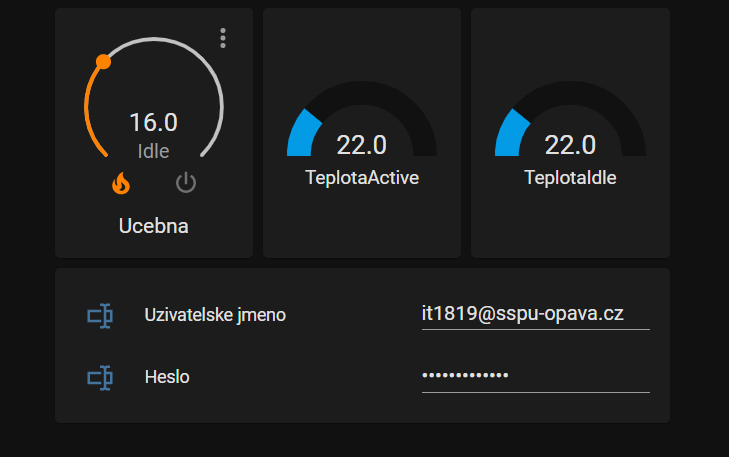
states.input\_number.tempidle }};{{ states.input\_number.tempactive }};"

mode: single

*Nastavení v automations.yaml*

## Vytvoření uživatelského rozhraní

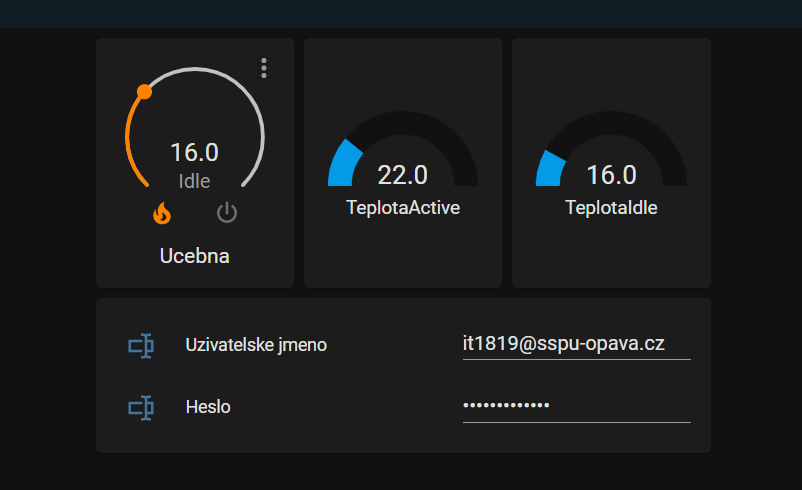
Při vyváření uživatelského prostředí musíme projít klikačkou na HA.



*Uživatelské rozhraní*

# Výsledky řešení, výstupy, uživatelský manuál

Nyní si ukážeme, jak nastavit teploty a přihlašovací údaje.



*Ukázka uživatelského prostředí*

V kolonce TeplotaActive si nastavíme, jakou teplotu chceme při přítomností hodiny v učebně.

V kolonce TeplotaIdle si nastavíme, jakou teplotu chceme při nepřítomností hodiny v učebně.

Do kolonky uživatelské jméno vypíšeme email od Školy online.

Do kolonky heslo vypíšeme heslo od Školy online.

V kolonce Ucebna vidíme aktuálně nastavenou teplotu.

# Závěr

Projekt mi přinesl větší přehled o skvělé aplikaci Home Assistant a taky nějaké ty znalosti o parsování webových stránek.

Myslím si, že projekt by mohl mít mnoho vylepšení, jako například lepší uživatelské pro-středí pro přidání dalších termostatů a aktualizování parametrů, lepší implementace scriptu do HA, a také lepší kontrola nad samostatným termostatem.

Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] https://en.wikipedia.org/wiki/Home\_Assistant

[2] https://chromedriver.chromium.org/

[3] https://en.wikipedia.org/wiki/Beautiful\_Soup\_(HTML\_parser)

[4] https://github.com/home-assistant/supervised-instal