

# Projektová dokumentace

## Klient pro OpenWeatherMap API

Dominik Večeřa (xvec23)

### 1 Popis programu

Program slouží jako klient používající rozhraní meteorologického serveru OpenWeatherMap, z něž získá informace o počasí v uživatelem zadané lokalitě. Je napsán v jazyce Python s použitím knihoven `sys`, `socket` a funkce `loads` pocházející z knihovny `json`. Komunikace s API OpenWeatherMap je zajištěna pomocí protokolu HTTP. Výstupní data jsou vypsána v metrických jednotkách.

### 2 Spuštění programu

Program se spouští pomocí souboru `Makefile` z příkazové řádky následovně:

```
make run api_key=<API klíč> city=<Město>
```

kde `<API klíč>` je unikátní identifikátor přiřazený každému uživateli při vytvoření bezplatného účtu na OpenWeatherMap využívaný k autentizaci uživatele odesílajícího daný požadavek a `<Město>` je jméno města, v němž chceme zjistit aktuální údaje o počasí. Lze také specifikovat stát, v němž se dané město nachází, aby se vybrala správná lokalita pro případ existence více měst se stejnými jmény v jiných zemích. Za název města se poté přidá kód státu podle ISO 3166 oddělený čárkou, například takto: `London,uk`. Příklad použití:

```
make run api_key=b498767252de12f92574d288a9c4fdc2 city=Prague
```

### 3 Implementace programu

Nejdříve je zkontrolován správný počet zadaných argumentů programu. Poté jsou v případě jejich korektního zadání načteny a je vytvořena zpráva, která se bude odesílat serveru.

Dále je vytvořen nový socket s využitím knihovny `socket` a její stejnojmenné funkce, jíž jsou předány konstanty `AF_INET`, která specifikuje, že socket může komunikovat s adresami typu IPv4, a `SOCK_STREAM`, která specifikuje typ socketu TCP.

Poté se inicializuje připojení k socketu na adrese OpenWeatherMap API s portem 80 a je mu odeslána dříve vytvořená zpráva s žádostí o data a poté jsou přijata data, která daný socket zašle. V případě chyby při komunikaci program skončí s chybou.

Pokud byl přijat jiný HTTP kód než 200, což značí neúspěch při získávání dat, program vypíše přijatý chybový kód a pro několik nejčastějších chybových kódů detailnější zprávu o chybě a pro všechny ostatní informaci, že došlo k chybě.

Pak jsou z přijatých dat odstraněny informace protokolu HTTP, provedeny menší úpravy tak, aby šlo přijatá data reprezentovaná v notaci JSON správně načíst pomocí funkce `loads` knihovny `json` a nakonec jsou tato data vypsána na standardní výstup v uživatelsky čitelném formátu.