Requester

Nástroj pro testování REST API

David Nápravník

Zápočtová práce pro předměty
Jazyk C# a platforma .NET - NPRG035 a
Pokročilé programování pro .NET I - NPRG038,
Programování uživatelských rozhraní v .NET - NPRG064

KDSS (Katedra distribuovaných a spolehlivých systémů)
Matematicko-fyzikální fakulta
Univerzita Karlova
26.07.2020

Contents

1 Dokumentace požadavků

1.1 Předpokládané využití

Program by měl primárně pomoci při vývoji REST API a jeho testování, pomocí posílání zkušebních dotazů (tkz. Requestů).
Dále by měl pomoci se stahováním dat z webu, ať už manuálně, nebo jako součást jiného programu, kdy lze mít předdefinovanovaou šablonu (tkz. Template) a tu obměňovat přes parametry, pro každý request.

1.2 Předpokládaná funkcionalita

Program musí umět posílat requesty typu **GET** a **POST**.
V requestu se musí nechat měnit hlavička (tkz. Header), tělo a parametry v URL.

2 Dokumentace architektury/designu

2.1 Core

Jádrem je třída *Requester* a další pomocné struktury.
Nejdůležitější metodou jádra je *Send*, která dostane co a kam odeslat a vrací odpověď ve formátu *RequestResponse* obsahujcící status code, jak číselně tak jeho slovní formát, hlavičku, tělo odpovědi (tkz. Content) a čas od odeslání dotazu k jeho plnému přijetí.

2.2 GUI

Uživatelé se budou nejčastěji potýkat s GUI postaveným nad jádrem.
GUI pro odeslání requestu schromáždí data od uživatele a předá je jádru, přesněji metodě Send, po příchodu odpovědi ji zpracuje a na contentu spustí automat (buď XML, nebo JSON), který ji LAZY zpracovává a vrací jak symbol zpracovat (text, řídící znak atd.), což je podstatné pro zobrazení ”pretty” odpovědi, která je obarvená a je indentovaná podle hloubky zanoření.

2.2.1 DataGrid

Seznam hodnot v hlavičce a parametrů pro url jsou zobrazeny pomoci DataGridu, což je vpodstatě tabulka do které uživatel může přidávat / odebírat řádky, měnit jejich hodnoty a nebo je před odeláním zakázat.

2.2.2 Prohlížeč

Content lze zobrazit i tak jak by jej viděl prohlížeč. K tomu je použit defaultní prohlížeč z knihovny WPF, s podobným jádrem jako prohlížeč IE9.

2.2.3 Design

Uživatelské rozhraní se nese v duchu Material designu vytvořeného společností Google. Dbá na jednoduché, přehledné a uživatelsky přívětivé prostředí. Všechny části jsou responzivní a připravené na různé velikosti displaye. Barvy v GUI jsou v tmavém režimu, jež šetří oči při dlouhodobějším používaní.

2.3 CLI

CLI verze programu je Headless a tudíž nepožaduje po svém spuštění, žádnou interakci a sama vždy skončí, buď neúspěšně, nebo vrátí do konzole odpověď. Popř. uloží tělo requestu do souboru.
Lze spustit buď s Templatem nebo s parametry, nebo s obojím zároveň, s tím že nejdříve se načte konfirurace z Templeatu a pak je doplněna / pozměněna daty z parametrů.

3 Technická dokumentace

3.1 HTTP Client

Pro komunikaci samotnou byla použita knihovna *System.Net.Http*.
Requesty jsou typu SEND a POST a posílají se asynchronně.

3.2 Algoritmy

3.2.1 Automat na parsování JSON

Automat přijímá pokaždé jeden znak a vrací jakého typu znak je a hloubku zanoření.
Typy znaku:

- objectBracket
- arrayBracket
- paramName
- JSONstring
- number
- logic
- specialChar
- whitespace

3.2.2 Automat na parsování XML

Automat přijímá jeden znak a vrací v jakém stavu automat je a hloubku zanoření (odhad hloubky v případě nedeterministického stavu).
narozdíl od Automatu pro JSON, je tento automat nedeterministický a to v případě znaku ”¡” protože nemůžeme vědět, zda je to začátek nového tagu, nebo ukončení starého, z čehož nelze vyvodit hloubka zanoření a je třeba počkat na další symbol.
Stavy automatu:

- left
- tagName
- whitespace
- metadata
- right
- data
- slash

3.3 Ukládání Templateu

Templaty se ukládají ve formátu JSON a obsahují vše co uživatel zadá do rozhraní GUI.
Ukázka templeatu requestu pro získání IP odesílatele:

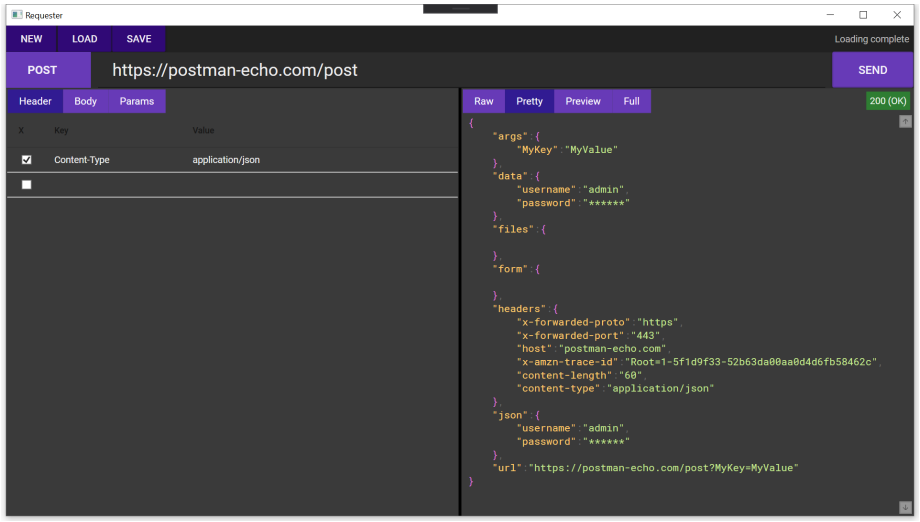
```
{
    "url": "https://api.ipify.org",
    "method": "GET",
    "parameters": [
    ],
    "header": [
        {
        "active": true,
        "key": "Content-Type",
        "value": "application/json"
        }
    ],
    "content": ""
}
```

3.4 Testy

Všechny public metody z Jádra mají vlastní testy, jež ověřují korektnost výstupu jak pro běžné použití, tak i pro pár edge-*case případů.

4 Uživatelská dokumentace

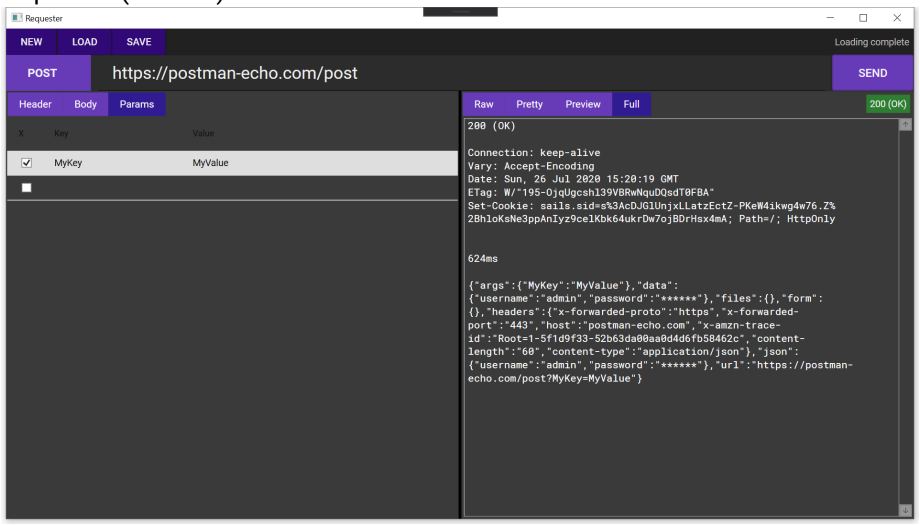
4.1 Rozhraní GUI



V horní části vidíme 3 tlačítka pro ukládání a načítání templeatu a pole s informacemi pro uživatele v pravém rohu
Níže je výběrové pole pro typ requestu (POST nebo GET). Uprostřed je místo pro URL. Program přijímá i url bez udání protokolu.
Tlačítko SEND pak odesílá request.
Levá spodní část je vstupní a pravá výstupní.

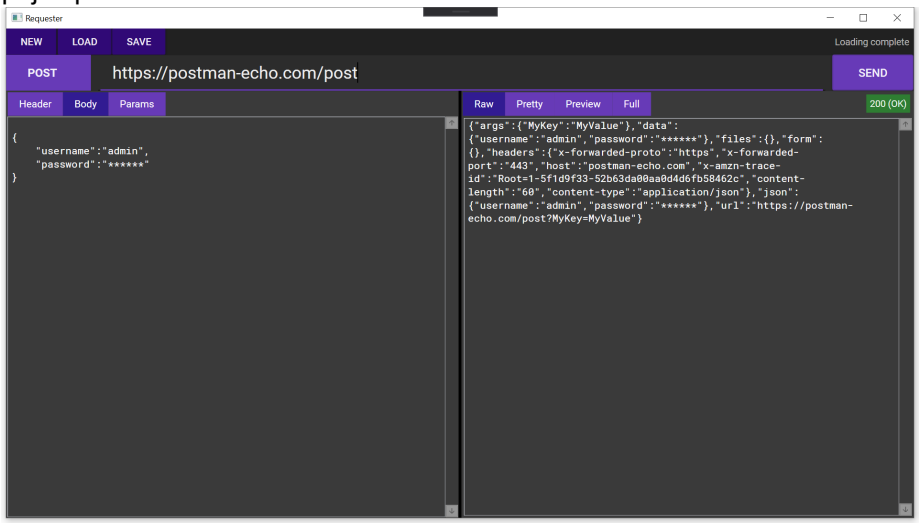
Vlevo vidíme tabulku, kde každý řádek je jedna položka v hlaviččce. Řádky mažeme pomocí označení a klávesy Del. Nové řádky se přidávají automaticky.

Vpravo vidíme výsledek requestu, tak jak jej automat sparsoval a kód přijatého requestu. (200 OK)

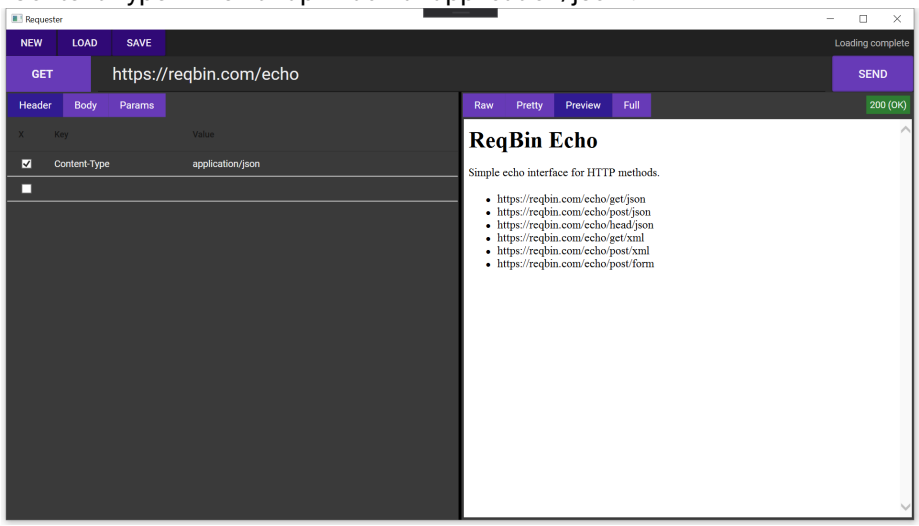


Vlevo vidíme podobnout tabulku jako pro zadávání hlavičky.

Vpravo je zobrazen celý výsledek requestu ...Stavový kód, Hlavička, čas pro přijetí požadavku a tělo.



Tělo requestu (tkz. Body) je čistě text a tak se i zpracovává. Pokud chtete odesílat JSON nebo XML, je třeba změnit datový typ v hlavičce. Přesněji položku ”Content-Type” změnit například na ”application/json”.



V případě potřeby je možné zobrazit odpověď i tak jak ji vidí uživatel v prohlížeči a to v záložce Preview.

4.1.1 Ukládání Templateu

Pro uložení Templatu stačí zmáčnout tlačítko SAVE a vybrat cílový adresář.
Pro načtení slouží tlačitko LOAD, templaty se vždy ukládají s koncovkou JSON.

4.1.2 Klávesové zkratky

- Ctrl + S → Uložení templeatu do souboru

- Ctrl + O → Načtení templeatu ze souboru
- Ctrl + Enter → Odeslání Requestu

4.2 Rozhraní CLI

Ukázka použití:

```
> .\bin\Debug\Requester.exe -t '.\sample requests\GET IP.json'
200 (OK)

Connection: keep-alive
Vary: Origin
Date: Sun, 26 Jul 2020 15:48:41 GMT
Server: Cowboy
Via: 1.1 vegur

530ms

93.99.225.88
```

Argumenty:

-o output file

-t template

-m method

-u url

-c content

-h header

V případě uvedení argumentu -o se ukládá pouze tělo odpovědi, nikolov celý záznam (hlavička atd.).