Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Porovnanie API tokenov

Bakalárska práca

2023
Jitka Muravská

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Porovnanie API tokenov

Bakalárska práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	RNDr. Richard Ostertág, PhD.

Bratislava, 2023
Jitka Muravská

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Jitka Muravská
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Porovnanie API tokenov
Comparison of API Tokens

Anotácia: REST aplikačné rozhrania (API) často nebývajú čisto verejné. V takom prípade treba pri implementácii API riešiť aj jeho bezpečnosť. Väčšina schém zabezpečenia API používa token, ktorý je súčasťou jednotlivých požiadaviek. Tieto tokeny sú nejakým spôsobom spojené s identitou a autorizáciou používateľa. Aplikačné rozhranie prevezme požiadavku, extrahuje token, a podľa pravidiel prístupu rozhodne ako pokračovať.

Cieľom práce je porovnanie rôznych API tokenov (napríklad: OAuth 2.0, JWT, PASETO, Protobuf Tokens, Authenticated Requests, Macaroons, Biscuits, …). Prvým krokom bude zozbieranie a popísanie rôznych v praxi používaných API tokenov. Následne sa vykonajú porovnania ich výhod a nevýhod (napríklad rýchlosť, jednoduchosť použitia) na jednoduchej základnej aplikácii.

Vedúci: RNDr. Richard Ostertág, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 12.10.2022

Dátum schválenia: 13.10.2022 doc. RNDr. Dana Pardubská, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Zoznam obrázkov

Zoznam tabuliek

Kapitola 1

Ukážky užitočných príkazov v systéme LaTeX

V tejto kapitole si ukážeme príklady niektorých užitočných príkazov, ako napríklad správne používanie tabuliek a obrázkov, číslovanie matematických výrazov a podobne. Konkrétne príkazy použité v tejto kapitole nájdete v zdrojovom súbore `latex.tex`. Všimnite si, že pre potreby obsahu a hlavičky stránky je v zdrojovom súbore uvedený aj skrátený názov tejto kapitoly. Ďalšie užitočné príkazy nájdete aj v kapitole **??**, na ktorú sme sa na tomto mieste odvolali príkazom `\ref`.

1.1 Obrázky

Vašu prácu ilustrujte vhodnými obrázkami. Pri použití programu pdflatex je potrebné pripraviť obrázky vo formáte pdf, jpg alebo png. Vektorové obrázky (napr. eps, svg) je najvhodnejšie skonvertovať do formátu pdf, napríklad programom Inkscape.

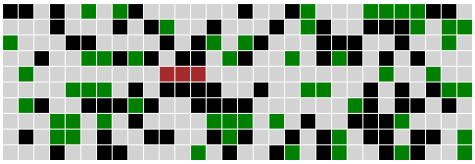
Na vkladanie obrázkov použite prostredie `figure`, ktoré obrázok umiestni na vhodné miesto, väčšinou na vrch alebo spodok stránky a tiež sa stará o automatické číslovanie obrázkov. Na každý obrázok sa treba v hlavnom texte odvolať. Napríklad ilustráciu hry Červík vidíme na obrázku 1.1. Pri odvolávaní sa na číslo obrázku používame príkaz `\ref`. Pri vložení alebo zmazaní obrázku tak nemusíme ručne všetky ostatné obrázky prečíslovať.

Podobne tabuľky vkladajte pomocou prostredia `table`, pričom samotnú tabuľku vytvoríte príkazom `tabular`. Každú tabuľku potom spomeňte aj v hlavnom texte. Napríklad v tabuľke 1.1 vidíme porovnanie časov niekoľkých fiktívnych programov.

V texte môžete tiež potrebovať dlhšie matematické výrazy, ako napríklad tento

$$\sum_{k=0}^{n} q^k = \frac{q^{n+1} - 1}{q - 1}. \tag{1.1}$$

Použitím prostredia `equation` bol tento výraz zarovnaný na stred na zvláštnom riadku



Obr. 1.1: Ukážka hry Červík. Červík je znázornený červenou farbou, voľné políčka sivou, jedlo zelenou a steny čiernou. Hoci tento popis obrázku je dlhší, v zdrojovom texte je aj kratšia verzia, ktorá sa zobrazí v zozname obrázkov.

Tabuľka 1.1: Doba výpočtu a operačná pamäť potrebná na spracovanie vstupu XYZ. V tomto popise môžeme vysvetliť detaily potrebné pre pochopenie údajov v tabuľke.

Meno programu	Čas (s)	Pamäť (MB)
Môj super program	25.6	120
Speedy 3.1	32.1	100
VeryOld	244.1	200

a očíslovaný. Na toto číslo sa tiež môžeme odvolať príkazom `\ref`. Napríklad rovnica (1.1) predstavuje súčet geometrickej postupnosti.

V práci tiež možno budete uvádzať úryvky kódu v niektorom programovacom jazyku. Môže vám pomôcť prostredie `lstlisting` z balíčka `listings`, v ktorom môžete nastaviť aj jazyk a kód bude krajšie sformátovaný. Ukážku nájdete ako Algoritmus 1.1.

Napokon, v texte nezabudnite citovať použitú literatúru pomocou príkazu `\cite` Napríklad ďalšie detaily o systéme LaTeX nájdete v knihe od Tobiasa Oetikera a kolektívu [6]. Pre ukážku citujeme aj článok z vedeckého časopisu [3] a článok z konferencie [2], technickú správu [4], knihu [1] a materiál z internetu [8].

Algoritmus 1.1: Algoritmus na výpočet faktoriálu v jazyku C

```
int factorial = 1;
for(int i = 1; i <= n; i++) {
    factorial *= i;
}
```

Úvod

Cieľom tejto práce je poskytnúť študentom posledného ročníka bakalárskeho štúdia informatiky kostru práce v systéme LaTeX a ukážku užitočných príkazov, ktoré pri písaní práce môžu potrebovať. Začneme stručnou charakteristikou úvodu práce podľa smernice o záverečných prácach [8], ktorú uvádzame ako doslovný citát.

> Úvod je prvou komplexnou informáciou o práci, jej cieli, obsahu a štruktúre. Úvod sa vzťahuje na spracovanú tému konkrétne, obsahuje stručný a výstižný opis problematiky, charakterizuje stav poznania alebo praxe v oblasti, ktorá je predmetom školského diela a oboznamuje s významom, cieľmi a zámermi školského diela. Autor v úvode zdôrazňuje, prečo je práca dôležitá a prečo sa rozhodol spracovať danú tému. Úvod ako názov kapitoly sa nečísluje a jeho rozsah je spravidla 1 až 2 strany.

V nasledujúcej kapitole nájdete ukážku členenia kapitoly na menšie časti a v kapitole 1 nájdete príkazy na prácu s tabuľkami, obrázkami a matematickými výrazmi. V kapitole **??** uvádzame klasický text Lorem Ipsum a na koniec sa budeme venovať záležitostiam záveru bakalárskej práce.

Kapitola 2

Prehľad a súvisiaca literatúra

V tejto kapitole sa zameriame na prehľad témy API tokenov a súvisiacu literatúru, ktorá sa venuje nami skúmanej problematike. Popíšeme, čomu sa nižšie uvedené zdroje venujú a ako sa líšia od cieľov našej práce.

2.1 Prehľad

Ako prvé si v krátkosti predstavíme tému API tokenov a vysvetlíme základné pojmy. API token (ďalej token) je identifikátor, ktorý sa používa na autorizáciu prípadne aj identifikáciu používateľa pri prístupe k API. Používateľom v tomto kontexte môže byť webová aplikácia, server alebo iný program, ďalej ich súhrnne budeme označovať ako klient.

Token typicky vydáva autentifikačný server danej API, voči ktorému sa klient autentifikuje, zväčša prihlasovacím menom a heslom. Následne autentifikačný server vytvorí token pre klienta a tento token bude používať klient pri ďalších požiadavkách na API. Token je typicky uložený v pamäti klienta a pri každej požiadavke na API je posielaný spolu s inými údajmi. Tokeny sa môžu vytvárať pomocou šifrovania, elektronických podpisov alebo hashovania. Napríklad šifrovanie v prípade, ak chceme chrániť citlivé údaje v tokene, elektronické podpisy sa využívajú v prípade, ak chceme overiť, že token bol vytvorený autentifikačným serverom a hešovanie je vhodné v prípade, ak chceme overiť, že token nebol modifikovaný.

2.2 Súvisiaca literatúra

K problematike rôznych typov tokenov nie sú dostupné odborné články alebo knihy a odborná literatúra spočíva najmä z príspevkov na blogoch a technických špecifikácií autentifikačných a autorizačných protokolov a konkrétnych tokenov. Zamerali sme sa na blog najširší záber tokenov.

2.3 Porovnanie do šírky

Tokeny sa môžu líšiť štrukturálne. To znamená, ktorou dátovou štruktúrou sú reprezentované a akú informáciu v sebe nesú. Obyčajne takto delíme tokeny na nepriehľadné (angl. opaque) a štruktúrované. Nepriehľadné tokeny predstavujú náhodné reťazce, ktoré neobsahujú (ani v zašifrovanej podobe) relevantnú informáciu, naopak štrukturované tokeny obsahujú navyše informáciu vhodnú pre API, napríklad napríklad môže obsahovať identifikátor používateľa, pre ktorého bol vydaný, čas vytvorenia alebo dátum expirácie.

Širokospektrálnemu prehľadu hlavne štrukturovaných typov tokenov sa venuje na svojom blogu [7] Thomas Ptacek. V texte najprv uvádza najjendnoduchšii typ tokenu a to nepriehľadný token pozostávajúci z náhodného reťazca a označuje ho ako náhodný token. Náhodný token je výhodný, lebo na jeho vydanie ani overenie netreba počítať zložité kryptografické funkcie. Jednoducho sa vygeneruje náhodný reťazec. Problém náhodných tokenov je, že si API potrebuje udržiavať zoznam aktívnych tokenov a im zodpovedajúcim metadát ako napríklad používateľa, pre ktorého bol token vydaný, ku ktorým volaniam má prístup, či časovú pečiatku dokedy je daný token platný.

Neskôr prechádza postupne viacero typov tokenov ako JSON web token (JWT), PASETO, CAT (z anglického Crypto Auth Token) a makaróny (angl. macaroons). O každom sa dozvieme ako je daný token reprezentovaný a ďalej sa autor venuje hlavne kryptografickej bezpečnosti. Autor pomyselne rozdelí tokeny na 'podobné JWT' a 'Anti-JWT'.

2.3.1 podobné JWT

Tu sa zameriava na porovnanie JWT a POSETO. Oba sú štrukturované a kým JWT, ako z názvu vyplýva, je reprezentovaný štruktúrou JSON, PASETO môže byť v niektorých verziách zachytený aj štruktúrou CBOR (z anglického Concise Binary Object Representation). Ptacek hlavne rozvýja problémy JWT, ktorý označuje za "kryptografický neporiadok". JWT dovoľuje programátorovi si vybrať kryptografické zabezpečenie z veľa rôznych symetrických aj asymetrických riešení. Týmto sa otvárajú dvere bezpečtnostým zraniteľnostiam. Napríklad problém známy ako RSAtoHMAC [5] využíva fakt, že pri verifikovaní nechávame útočníka si vybrať, ktorá metóda sa používa na verifikáciu tokenu.

Problémy, ktoré má JWT sa snaží vyriešiť PASETO. Verzionuju celý protokol a nepridáva nové možnosti pre programátora vrámci jednej verzie, čím sa snaží vyvarovať spomínanému 'kryptografickému neporiadku'. Najväčší problém PASETO, ktorý autor vypichuje je viacero aktuálnych verzii. PASETO má spolu 8 verzii, z ktorých až 4 sú označené za aktuálne.

Záverom jeho pozorovaní je, že je lepšie použiť PASETO, ale stále to nie je ideálne riešenie a silno naklonený k použitiu 'Anti-JWT' tokenov.

2.3.2 Anti-JWT

Anti-JWT tokeny sú tokeny, ktoré odstraňujú problémy JWT, lebo nepoužívajú podpisy ani zdieľané tajomstvo medzi API, autentifikačným serverom ani klientom. Autor uvádza CAT a makaróny. Obe tieto riešenia fungujú na základe 'zlatého kľuča', ktorý pozná len autentifikačný server. Z neho odvádza overovací kľúč pre API, z ktorého ďalej vytvorí generovací kľúč pre klienta. Klient si následne pomocou svojho kľúča vytvorí token, ktorý použije na prístup k API. API vie tento token overiť, lebo je vytvorený známym spôsobom z overovacieho kľúča, ktorý pozná.

Pri týchto protokoloch najviac vyzdvihuje flexibilnosť, jednoduchosť a najmä 'kryptografický poriadok', ktorý prinášajú. Kedy je jasné aký algortimus na krtypografické zabezpečenie.

2.3.3 V čom je naše práca iná

Ptacek sa vo svojom blogu venuje hlavne kryptografickej bezpečnosti tokenov. A celkovo robí prehľad najmä do šírky a nevenuje sa detailom jednotlivých protokolov. V našej práci chceme porovnávať tokeny podľa viacerých kritérií, teda spraviť komplexnejšie porovnanie. Tokeny budeme samozrejme porovnávať aj podľa bezpečnoti, keďže to je ich dôležitý aspekt. No navyše porovnáme aj ich flexibilnosť, škálovateľnosť, jednoduchosť, rýchlosť a ďalšie kritériá.

Kapitola 3

Využitie API tokenov

V tejto kapitole si priblížime, kde sa vyuzívajú API tokeny a aké sú rozdiely pri použití tokenov v autentifikačnej a autorizačnej schéme oproti iným známym prístupom ako API kľúče, či využitie prihlasovacieho mena a hesla.

Kapitola 4

Typy API tokenov

V tejto kapitole predstavíme v praxi využívané typy API tokenov a uvedieme ich základané charakteristiky.

Kapitola 5

Teoretické parametre API tokenov

V tejto kapitole porovnáme teoretické parametre rôznych typov API tokenov ako bezpečnosť, škálovateľnosť a flexibilita.

Kapitola 6

Praktické parametre API tokenov

V tejto kapitole porovnáme praktické parametre rôznych typov API tokenov ako jednoduchosť implementácie a rýchlosť.

Kapitola 7

Porovnanie na jednoduchom rozhraní

V tejto kapitole navrhneme a implementujeme jednoduché rozhranie s použitím viacerých typov API tokenov. Výsledkom budú naše pozorovania pri implementacii a použití rozhrania s jednotlivými typmi API tokenov.

Záver

Na záver už len odporúčania k samotnej kapitole Záver v bakalárskej práci podľa smernice [8]: „V závere je potrebné v stručnosti zhrnúť dosiahnuté výsledky vo vzťahu k stanoveným cieľom. Rozsah záveru je minimálne dve strany. Záver ako kapitola sa nečísluje.“

Všimnite si správne písanie slovenských úvodzoviek okolo predchádzajúceho citátu, ktoré sme dosiahli príkazom `\uv`.

V informatických prácach niekedy býva záver kratší ako dve strany, ale stále by to mal byť rozumne dlhý text, v rozsahu aspoň jednej strany. Okrem dosiahnutých cieľov sa zvyknú rozoberať aj otvorené problémy a námety na ďalšiu prácu v oblasti.

Abstrakt, úvod a záver práce obsahujú podobné informácie. Abstrakt je kratší text, ktorý má pomôcť čitateľovi sa rozhodnúť, či vôbec prácu chce čítať. Úvod má umožniť zorientovať sa v práci skôr než ju začne čítať a záver sumarizuje najdôležitejšie veci po tom, ako prácu prečítal, môže sa teda viac zamerať na detaily a využívať pojmy zavedené v práci.

Literatúra

[1] X. Autor1 and Y. Autor2. *Názov knihy.* Vydavateľstvo, 1900.

[2] X. Autor1 and Y. Autor2. Názov článku (väčšinou z konferencie). In *Názov zborníka (väčšinou názov konferencie spolu s ročníkom)*, pages 1–100. Vydavateľstvo, 1900.

[3] X. Autor1 and Y. Autor2. Názov článku z časopisu. *Názov časopisu, ktorý článok uverejnil*, 4(3):1–100, 1900.

[4] X. Autor1 and Y. Autor2. Názov technickej správy. Technical Report TR123/1999, Inštitút vydávajúci správu, June 1999.

[5] Tim McLean. Critical vulnerabilities in json web token libraries, August 2020. `https://auth0.com/blog/critical-vulnerabilities-in-json-web-token-libraries/`.

[6] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *Nie príliš stručný úvod do systému LaTeX2e.* 2002. Preklad Ján Buša ml. a st.

[7] Thomas Ptacek. Api tokens: A tedious survey, August 2021. `https://fly.io/blog/api-tokens-a-tedious-survey/`.

[8] Univerzita Komenského v Bratislave. Vnútorný predpis č. 7/2018, Úplné znenie vnútorného predpisu č. 12/2013 Smernice rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigoróznych prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave v znení dodatku č. 1 a dodatku č. 2 smernica rektora Univerzity Komenského v Bratislave o základných náležitostiach záverečných prác, rigoróznych prác a habilitačných prác, kontrole ich originality, uchovávaní a sprístupňovaní na Univerzite Komenského v Bratislave, 2013. [Citované 2020-10-19] Dostupné z `https://uniba.sk/fileadmin/ruk/legislativa/2018/Vp_2018_07.pdf`.

Príloha A: obsah elektronickej prílohy

V elektronickej prílohe priloženej k práci sa nachádza zdrojový kód programu a súbory s výsledkami experimentov. Zdrojový kód je zverejnený aj na stránke `http://mojadresa.com/`.

Ak uznáte za vhodné, môžete tu aj podrobnejšie rozpísať obsah tejto prílohy, prípadne poskytnúť návod na inštaláciu programu. Alternatívou je tieto informácie zahrnúť do samotnej prílohy, alebo ich uviesť na obidvoch miestach.

Príloha B: Používateľská príručka

V tejto prílohe uvádzame používateľskú príručku k nášmu softvéru. Tu by ďalej pokračoval text príručky. V práci nie je potrebné uvádzať používateľskú príručku, pokiaľ je používanie softvéru intuitívne alebo ak výsledkom práce nie je ucelený softvér určený pre používateľov.

V prílohách môžete uviesť aj ďalšie materiály, ktoré by mohli pôsobiť rušivo v hlavnom texte, ako napríklad rozsiahle tabuľky a podobne. Materiály, ktoré sú príliš dlhé na ich tlač, odovzdajte len v electronickej prílohe.