



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Vývoj Javascript knihovny pro embedování vizualizací skrze Emplifi Public API

Milan Janoch



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Vývoj Javascript knihovny pro embedování vizualizací skrze Emplifi Public API

Milan Janoch

Vedoucí práce

Doc. Ing. Dalibor Fiala, Ph.D.

© Milan Janoch, 2023.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

JANOCH, Milan. *Vývoj Javascript knihovny pro embedování vizualizací skrze Emplifi Public API*. Plzeň, 2023. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Doc. Ing. Dalibor Fiala, Ph.D.

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Milan JANOCH**
Osobní číslo: **A21B0152P**
Adresa: **V Brance 20, Přeštice, 33401 Přeštice, Česká republika**
Téma práce: **Vývoj Javascript knihovny pro embedování vizualizací skrze Emplifi Public API**
Téma práce anglicky:
Jazyk práce: **Čeština**
Vedoucí práce: **Doc. Ing. Dalibor Fiala, Ph.D.**
Katedra informatiky a výpočetní techniky

Zásady pro vypracování:

1. Prostudujte problematiku integrování analytických grafů do aplikací třetích stran v kombinaci se zabezpečeným přístupem přes OAuth 2.
2. Navrhněte knihovnu pro komunikaci s public API od firmy Emplifi včetně zabezpečeného přístupu.
3. Implementujte navrženou knihovnu.
4. Řešení řádně otestujte a minimálně kritické části testujte i s využitím unit testů.
5. Vytvořené řešení kriticky zhodnoťte.

Seznam doporučené literatury:

dodá vedoucí bakalářské práce

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 15. prosince 2023

.....

Milan Janoch

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Tato bakalářská práce se zaměřuje na vývoj specializované JavaScript knihovny s cílem umožnit snadné embedování vizualizací do aplikací třetích stran. Hlavními cíli práce jsou návrh a implementace knihovny, návržení rozhraní pro efektivní komunikaci s Public API firmy Emplifi a zajištění bezpečného přístupu k datům pomocí OAuth 2 protokolu. V teoretické části práce je diskutována problematika spojená s embedováním vizualizací do externích aplikací, bezpečný přístup k datům třetích stran a jsou analyzována již existující řešení. Praktická část se zaměřuje na návrh a implementaci JavaScript knihovny, popisuje návržení rozhraní pro efektivní komunikaci s API a zabývá implementací bezpečného přístupu v souladu se standardem OAuth 2.

Abstract

This bachelor thesis focuses on the development of a specialized JavaScript library to enable easy embedding of visualizations into third-party applications. The main goals of the thesis are to design and implement the library, design an interface to communicate efficiently with Emplifi's Public API and provide secure data access using the OAuth 2 protocol. The theoretical part of the thesis discusses the issues related to embedding visualizations in external applications, secure access to third-party data and analyzes existing solutions. The practical part focuses on the design and implementation of a JavaScript library, describes the design of interfaces for efficient communication with APIs and deals with the implementation of secure access in accordance with the OAuth 2 standard.

Klíčová slova

OAuth 2.0 • embedování • Emplifi Public API • token • JavaScript

Poděkování

Na tomto místě bych rád poděkoval všem, kteří přispěli k úspěšnému dokončení této bakalářské práce. Velké díky patří především

Bc. Ondřeji Altmanovi za trpělivost, cenné rady a vstřícnost při implementaci praktické části

Doc. Ing. Daliborovi Fialovi Ph.D. za jeho podporu a ochotu při tvorbě teoretické části

a také rodině za podporu během celého studia.

Obsah

1	Úvod	3
2	Embedded analytics a zabezpečení	5
2.1	Princip	5
2.2	Zabezpečení	6
2.3	Existující řešení	8
3	Návrh knihovny	11
3.1	Technologie	11
3.1.1	JavaScript a React	11
3.1.2	API	12
3.1.3	OAuth 2.0	12
3.2	Funkcionality	12
3.2.1	Komunikace s API	12
3.2.2	Playground	12
3.2.3	Obnova OAuth 2.0 tokenů	12
3.2.4	PreJSON	13
4	Závěr	15
	Bibliografie	17
	Seznam obrázků	19
	Seznam tabulek	21
	Seznam výpisů	23

V současné době internetu tvoří data významnou a důležitou součást každodenního života. Problematikou dat je nejen efektivně ukládat, ale také je umět rychle a efektivně interpretovat. Vizualizace dat není pouhým trendem, ale stala se klíčovým nástrojem v mnoha odvětvích. Od oblasti logistiky, kde pomáhá monitorovat a řídit tok zásob a logistické operace, až po oblast sociálního marketingu, kde je využívána na personalizovaných reklamách či sledování aktivit zákazníků. Pro snadné integrování vizualizací do aplikací třetích stran se využívá koncept *embedded analytics* – ten umožňuje uživatelům rychlý a efektivní přístup datům bez potřeby přecházet mezi různými aplikacemi.

Cílem této práce, zadanou společností Emplifi, která se specializuje na sociální marketing, je vytvořit praktickou knihovnu umožňující embedování analytických grafů do aplikací třetích stran. Velký důraz bude kladen na bezpečný přístup k datům skrze OAuth 2 protokol, který je klíčovým prvkem Emplifi Public API. Součástí bude také administrační rozhraní, jenž umožní snadné generování a obnovování OAuth 2 tokenů.

V rámci teoretické části bude podrobně diskutována problematika spojená s embedováním. Analyzovat se bude princip, výhody a nevýhody bezpečnostního protokolu OAuth 2 a již existující řešení *embedded analytics*.

Tímto způsobem práce nespojuje pouze technologickou inovaci s nutností bezpečnosti dat, ale také reaguje na aktuální potřeby v odvětví sociálního marketingu a digitálního prostoru.

Embedded analytics a zabezpečení

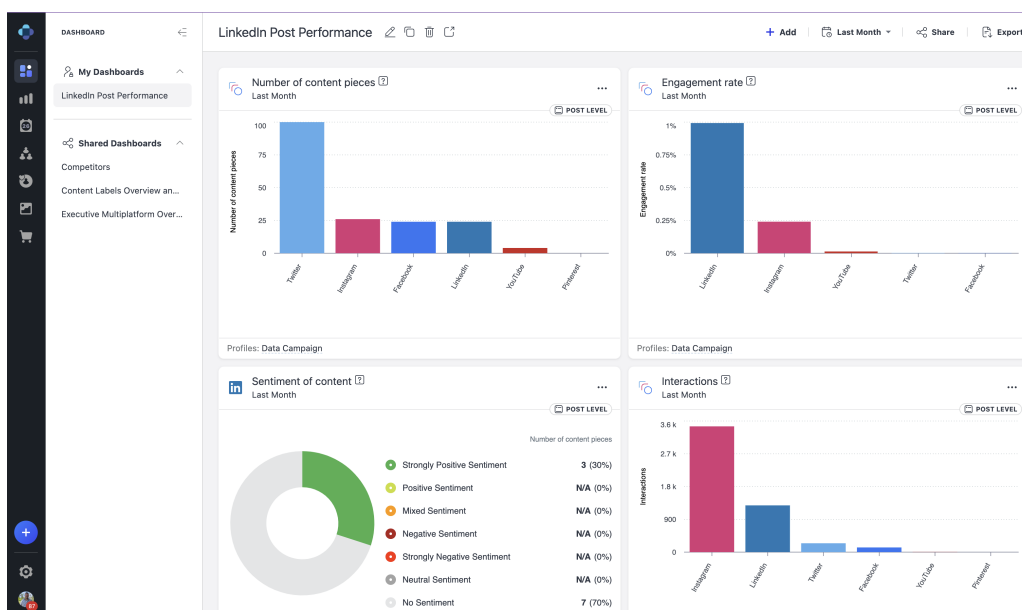
2

V této kapitole se zmíníme o principu embedded analytics, zabezpečení dat pomocí OAuth 2 tokenu a již existujících řešeních.

2.1 Princip

Embedded analytics transformuje data do grafů a dashboardů [1]. Dashboard je souhrn informací a nástrojů, který umožňuje rychle a jednoduše zobrazovat důležité metriky (např. počet komentářů pod příspěvkem) [2]. Emplifi dashboardy umožňují definovat, vytvářet a vizualizovat aktivity uživatelů na sociálních sítích [3], což umožňuje komplexní pohled na interakci s obsahem.

Na obrázku 2.1 vidíme příklad dashboardu - skládá se z několika grafů často



Obrázek 2.1: Ukázka dashboardu v produktu firmy Emplifi

nazývané jako widgety [3], jejichž obsah lze modifikovat pomocí přepínačů v liště. Obsah může být modifikován v časovém období (posledních 30 dní, poslední rok, konkrétní časový rozsah, ...), ale lze ho modifikovat také na základě metrik jako je např. sociální síť (Facebook, Instagram, LinkedIn), druh interakcí (komentáře, liky, sdílení) apod.

Embedded analytics se nechová a nevypadá jako samostatná aplikace, ale je integrován do jiného softwaru nebo webového portálu. Koncoví uživatelé často ani nepoznají, že se jedná integrovanou analytiku a vnímají software s touto integrovanou analytikou jako jeden nástroj [1]. To umožňuje softwarovým společnostem získat a plně integrovat analytickou platformu s jejich vlastním SaaS produktem bez nutnosti velkých investic do vývoje vlastního řešení.

SaaS (Software as a Service) je licenční model, v němž přístup je poskytován na základě předplatného, přičemž software je umístěn na externích serverech, nikoli na firemních serverech [4]. K těmto službám se běžně přistupuje prostřednictvím webového prohlížeče s přihlašovacím jménem a heslem. Výhodou je, že místo toho, aby mělo každé zařízení ve firmě nainstalované tento program, stačí se do programu přihlásit přes internet. Pro firmu to znamená úsporu financí, jelikož nemusí investovat do nového hardwaru, na němž by dané programy běžely. Nevýhodou SaaS je bezpečnost dat a rychlost jejich doručování. Protože jsou data umístěna na externích serverech, je třeba zajistit rychlé a spolehlivé internetové připojení a vyloučit přístup neoprávněných uživatelů.

Existuje hned několik způsobů, jak data embedovat. Mezi nejpopulárnější iframe a HTML, webová komponenta či React SDK s voláním API. Zatímco nejjednodušší a nejrychlejší metoda je používáním HTML iframu, pokročilejší techniky embedování využívají React SDK a API.

SDK (Software development kit) je označení sady nástrojů pro tvorbu softwaru [5], které umožňuje vývojářům vytvářet rychleji a standartizovaně nové aplikace. API (Application Programming Interface) je rozhraní usnadňující komunikaci mezi dvěma platformami. Uživatel specifikuje požadavek na data, rozhraní API provede volání na webový server, který následně tento požadavek zpracuje a odešle odpověď na specifikované volání (může se jednat např. o data ve formátu JSON). Díky tomu se krátí vývojový cyklus (automatizace), efektivně se poskytují nové služby uživatelům a může zlepšit reputaci důvěryhodnost značky.

2.2 Zabezpečení

Zabezpečení API je důležitou součástí v moderním inženýrství zajišťující bezpečnou a důvěryhodnou komunikaci mezi různými aplikacemi. S rostoucím významem API pro výměnu dat mezi systémy je nezbytné věnovat zvláštní pozornost implementaci efektivních bezpečnostních opatření. API authentication (autentizace) je řešením

pro ověřování uživatelů - to umožňuje vlastníkovvi daného API ochranu před neoprávněným přístupem ze strany uživatelů, kteří nemohou ověřit svou totožnost [6].

Řešení autentizace jsou obvykle nastavena tak, aby zablokovala přístup do API, pokud se při volání zjistí něco nesprávného/nevalidního s uživatelem. S tím se pojí druhá bezpečnostní složka, kterou je autorizace (authorization). Zatímco autentizace ověřuje identitu uživatele, autorizace se zabývá tím, jaké akce má daný uživatel povoleny [7]. Mezi nejpoužívanější metody patří HTTP authentication, API keys a OAuth 2.0.

HTTP authentication omezuje přístup k serveru pomocí Basic HTTP schématu [8]. Uživatel, který chce ověřit svou identitu, tak může učinit vložím hlavičky Authorization s bezpečnostními údaji - nejčastěji tím bývá uživatelské jméno a heslo.

API klíč (key) je jedinečný alfanumerický řetězec, který slouží k jednoznačné identifikaci klientů/aplikací [9], jenž API využívají. Narozdíl od tokenů nejsou časově omezené - to může znamenat bezpečnostní problém v případě, že při zadávání požadavku API by přenášený klíč mohl být zachycen [7].

OAuth (Open Authorization) protokol byl vyvinut jako řešení pro udělování přístupu na předem definovanou dobu bez sdílené uživatelských jmen a hesel [10]. OAuth 2.0 je nejlepší volbou pro identifikaci osobních uživatelských účtů a udělování patřičných oprávnění [7]. Proces autentizace je vidět na obrázku 2.2.

Nejprve se uživatel přihlásí do systému - nejčastěji formou uživatelského jména a hesla, nebo client_id a client_secret. Tím pošle HTTP požadavek na endpoint. Autorizace uživateli následně vrátí token. Pokud má uživatel již platný token, může přistupovat k datům. Naopak pokud mu token již expiroval nebo není platný, volání do API mu vrátí chybové hlášení.

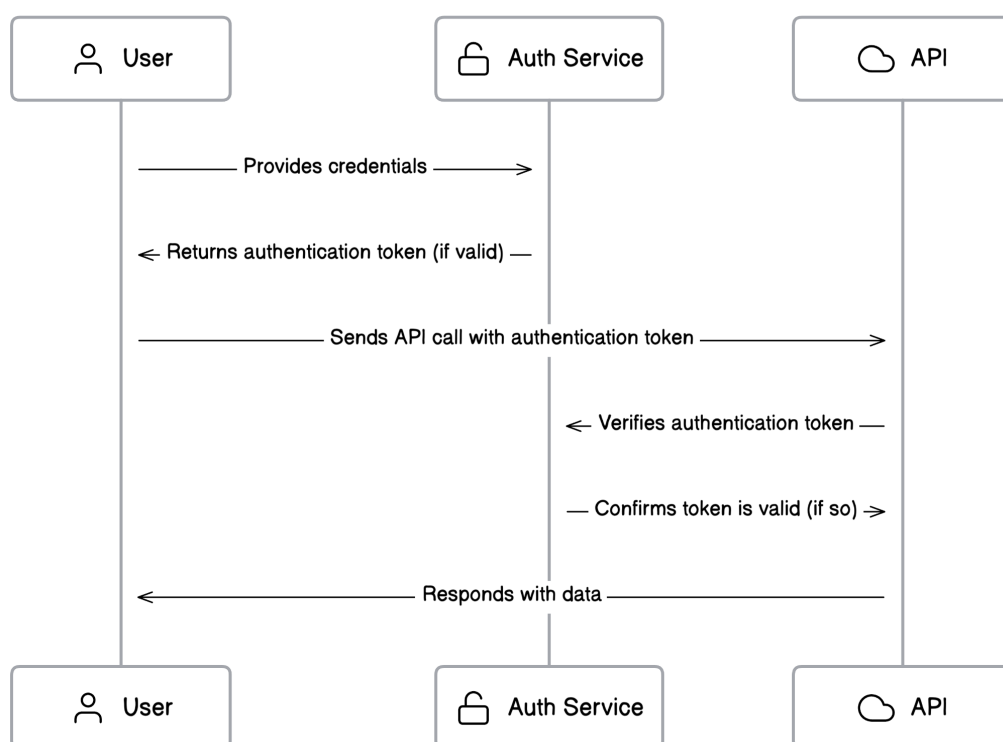
Součástí autorizační odpovědi bývá i několik dalších údajů. Odpověď může vypadat např. jako na výpisu 2.1.

Zdrojový kód 2.1: Ukázková odpověď autorizačního serveru

```

1 {
2     "access_token": "ABCDEF1234567890GHIJKL",
3     "token_type": "Bearer",
4     "expires_in": 3600,
5     "refresh_token": "
        ZYXWVU0987654321DCBANMLKJIHGFEDCBA",
6     "scope": "create"
7 }
```

Odpověď bývá ve formátu JSON a obsahuje informace o daném tokenu. Kromě přístupového tokenu obsahuje i refresh_token, který slouží k obnově tokenu po vypršení jeho platnosti (expires_in - platnost je v sekundách). To umožňuje uživateli



Obrázek 2.2: Autentizační diagram protokolu OAuth 2.0

používat stále jeden token bez nutnosti generování nového. Dále obsahuje atribut `scope`, který specifikuje práva uživatele a atribut `token_type` [11].

2.3 Existující řešení

Embedded analytics problematikou se zabývá mnoho firem a každá z nich disponuje jinými způsoby řešení. Zde budou zmíněny firmy, které se objevují často na vrchních příčkách internetových recenzí či firmy, které byly pro inspiraci doporučeny externím zadavatelem. Pro aktuálnost jsou brány informace ze zdrojů z roků 2022/2023.

Firma GoodData byla zmíněna zadavatelem jako jedna z nejvýraznějších na trhu. Dodává embedded analytics systémy firmám jako je VISA, Zalando či Bentley [12]. Embedování provádí třemi zmíněnými způsoby - HTML iFramem, webovou komponentou či React SDK. Zákazníkům nabízí přizpůsobení dashboardů tak, aby odpovídaly zákaznicko značce. GoodData získaly v roce 2023 hned několik ocenění firmou TrustRadius, důvěryhodnou platformou v oblasti B2B (business to business) [13]. Firma GoodData obdržela ocenění Best Value for Price, Best Relationship či Top Rated 2023. Recenze zákazníků firmy GoodData si chválí hlavně snadné používání, zákaznickou podporu, širokou škálu podporovaných databází či uživatelsky

přívětivé rozhraní [13]. Naopak se zmiňují, že dokumentace může být nedostačující či některé endpointy v API mají neintuitivní parametry. I tak ale 92% recenzentů uvádí, že by si produkt zakoupili znovu.

Mezi často zmiňovaným produktem je Microsoft Power BI, který je ve svém oboru nejlepší v oblasti bezpečnosti a zabezpečení [14], jelikož Microsoft zaměstnává více než 3500 bezpečnostních expertů. V produktu nabízí také tzv. playground, který umožňuje uživatelům testovat funkce a vlastnosti daných dashboardů před úplným nasazením. Je také podporována integrace s cloudovou službou Microsoft Azure. Na webu TrustRadius je k Microsoft Power BI podstatně méně recenzí než k produktu GoodData (5x méně). Uživatelé zmiňují dlouhou dobu při zpracování většího objemu dat či vysokou provozní cenu [15].

V oblasti modelování data vyniká platforma Powered by Looker [14]. Jedná se o produkt služby Google Cloud, který poskytuje způsob, jak sledovat změny a prohlížet jejich historii v databázi. Disponuje modelovacím jazykem LookML založeným na SQL, který slouží k vytváření sémantických datových modelů. Pomocí něj lze popisovat jednotlivé dimenze, agregáty, výpočty či datové vztahy v databázi [16]. Z toho vyplývá, že datové modely jsou rozšiřitelné, opakovatelně použitelné a konzistentní a tudíž efektivní. V roce 2022 byl tento produkt oceněn Top Rated a Most Loved společností TrustRadius [17]. Uživatelé zmiňují, že produkt může být poma-
lejší a že nástroje na přizpůsobení dashboardů by mohly být vylepšeny a rozšířeny.

3.1 Technologie

3.1.1 JavaScript a React

Pro vývoj knihovny bude využit JavaScript s knihovnou React. Důvodem zvolení těchto technologií je požadavek ze strany zadavatele. React a JavaScript jsou dnes široce používané technologie ve vývoji webových aplikací.

React je snadný na naučení, obsahuje málo konceptů, které je třeba se naučit [18]. Jeho instalace je snadná - stačí pouze nainstalovat jeho knihovnu. Využívá speciální JSX syntaxe, která sice vypadá jako HTML, ale ve skutečnosti je tato syntaxe převáděna na HTML. Jelikož je React hojně využíván v Facebook aplikaci či na Instagramu, dostává se mu velké podpory právě i z tohoto odvětví - čtyři největší přispěvatelé knihovny React jsou zaměstnanci Facebooku [18]. Kromě Facebooku využívají React značky jako jsou např. Netflix, airbnb, BBC News či PayPal [19]. Díky virtualizaci a uchovávaním DOM poskytuje React velmi rychlé vykreslování, přičemž všechny změny se snadno promítají do virtuálního DOM.

DOM (Document Object Model) je strukturovaná reprezentace jazyka HTML, které reprezentuje celé uživatelské rozhraní jakos stromovou datovou strukturu [20]. Každý prvek uživatelského rozhraní tvoří v DOM stromu právě jeden uzel. Dojde-li ke změně uživatelského rozhraní, DOM se aktualizuje a při každé změně se vykresluje znovu, což výrazně ovlivňuje výkon aplikace. Toto lze vyřešit použitím virtuálního DOM. Při přidávání nových věcí do aplikace se vytvoří virtuální DOM, která je reprezentována jako strom. Novější virtuální DOM se porovnává se starším, aby zaznamenal změny. Poté zjistí, jak je možné tyto změny provést pomocí skutečného DOM a aktualizované prvky se následně vykreslí.

Na popularitě Reactu přidává také použití knihovny Redux, která umožňuje uchování dat jako jeden objekt (stav). K tomuto objektu se dá pak jednoduše přistupovat, což značně usnadňuje práci s daty. Je-li tento stav změněn, aplikace se překreslí a zobrazení se stále synchronizuje se souvisejícími daty [19].

React se kvůli těmto vlastnostem doporučuje využívat u:

1. Obsáhlých uživatelských rozhraní
2. Rozsáhlých aplikací
3. Aplikací náročné na výkon
4. Multi-platformních aplikací

3.1.2 API

Data, která budou následně vizualizována knihovnou, budou získávána prostřednictvím Emplifi Public API. Toto rozhraní poskytuje klientům snadný a oblíbený způsob, jak získat přístup k potřebným datům.

3.1.3 OAuth 2.0

Přístup k datům bude zabezpečen protokolem OAuth 2.0. To umožní nejen autentizaci uživatele, ale také umožné uživatelům udělovat patřičná omezení (např. které endpointy může běžný uživatel volat).

3.2 Funkcionality

3.2.1 Komunikace s API

Navrhovaná knihovna bude vybavená funkcionalitou pro provádění dotazů na Emplifi Public API, což umožní získávat potřebná data pro vizualizaci. Bude také umět reagovat na chybové odpovědi ze strany API (např. nevalidní token, špatný request, neexistující data apod.).

3.2.2 Playground

Vytvoření playgroundu umožní uživatelům náhled na vizualizace, které budou chtít embedovat v aplikacích třetích stran. To se může využít zejména pokud si budou chtít ověřit funkčnost či pozměnit design vizualizace.

3.2.3 Obnova OAuth 2.0 tokenů

K dispozici bude uživatelské rozhraní, které umožní snadné vytvoření či obnovení OAuth 2.0 tokenu. Jedná se o důležitou funkcionalitu, aby budoucí uživatelé nemuseli složitě získávat token, ale stačilo jim k tomu pouze jednoduché rozhraní.

3.2.4 PreJSON

Interní knihovna, která bude zveřejněna během příštího roku. Informace budou dopsány, jakmile bude zveřejněna (včetně vhodné ukázky). Slouží k tomu, aby se do schémat grafů daly jednoduše vkládat hodnoty. Např. zavoláním Emplifi Public API se vrátí schéma grafu, které bude ukazovat aktivitu uživatelů za posledních X dní. Díky PreJSON knihovně se mohou jednoduše do tohoto schématu vkládat hodnoty, na kterých je graf závislý (v tomto případě počet dní). Bude poskytnuta i dokumentace, až se tato knihovna zveřejní, takže poté odcituju a provede ukázku.

Aktuálně je praktická část bakalářské práce ve formě, kdy React aplikace dokáže vizualizovat grafy. Grafy lze vizualizovat pomocí playgroundu, kterému bude v budoucnu ještě upraven frontend. Zároveň aplikace dokáže obnovovat OAuth2 tokeny (jak token pro Emplifi Public Api, který vrací schémata widgetů, tak i token pro Omni Api, který vrací data do widgetů). Toto je udělané pouze formou dvou tlačítek. Ty jsou automaticky vypnuté, pokud jsou tokeny validní - momentálně se uchovávají v local storage webového prohlížeče. Externí zadavatel určí co dále. Jsou i vypisovány základní stavy tokenu (platný, neplatný, neexistující) a vizualizace v playground nelze embedovat, dokud tyto tokeny nejsou platné. Playground umožňuje uživatelům widgety i upravovat - například zadáním CSS stylu či zadáním šířky/-výšky. Pro debugování a kontrolu jsou umožněny i výpisy schémat daného widgetu - ty se ovládají jednoduchým přepínačem. Jsou ošetřeny základní chyby, které mohou nastat při vizualizaci (chybějící tokeny, nezadané parametry pro PreJSON apod.).

Nyní bych rád pokračoval v zefektivnění kódu, dopsáním dokumentačních komentářů, napsáním jednotkových testů pro kritické sekce programu, zlepšení frontendu pro playground a pro rozhraní pro obnovu OAuth 2.0 tokenů. Zároveň bude třeba ošetřit všechny výjimky, které mohou nastat. Dalším krokem bude převést React aplikaci na knihovnu a domluvit se se zadavatelem, jak pokračovat dále - například zda-li stačí vizualizovat jednotlivé widgety, nebo celé dashboardy. Jako poslední budou provedeny i funkcionální testy podle scénářů, které budou přiloženy ve finálním odevzdání bakalářské práce.

Bibliografie

1. SEIDELOVA, Tereza. *Embedded Analytics: All You Need To Know*. GoodData, 2023-10-31. Dostupné také z: <https://www.gooddata.com/blog/what-embedded-analytics/>.
2. BRUK, Vojtěch. *Co je Dashboard?* 2023-02-22. Dostupné také z: <https://vojtechbruk.cz/pojem/dashboard/>.
3. *Dashboard*. Emplifi, 2023-12. Dostupné také z: <https://docs.emplifi.io/platform/latest/home/dashboard>. [Online: accessed 9.Dec.2023].
4. *Software as a Service (SaaS): Definition and Examples*. Investopedia, 2022-12-15. Dostupné také z: <https://www.investopedia.com/terms/s/software-as-a-service-saas.asp>.
5. *SDK vs. API: What's the Difference?* IBM, 2021-07-13. Dostupné také z: <https://www.ibm.com/blog/sdk-vs-api/>.
6. *What is API Authentication?* Noname Security, 2023-06-08. Dostupné také z: <https://nonamesecurity.com/learn/what-is-api-authentication/>.
7. LEVIN, Guy. *4 Most Used REST API Authentication Methods*. 2019-07-26. Dostupné také z: <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/>.
8. *HTTP authentication*. 2023-12. Dostupné také z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication>. [Online: accessed 9.Dec.2023].
9. *What is an API key?* Amazon Web Services, 2023-12. Dostupné také z: <https://aws.amazon.com/what-is/api-key/>. [Online: accessed 9.Dec.2023].
10. STARREVELD, Albert. *Understanding OAuth2*. 2023-08-02. Dostupné také z: <https://medium.com/web-security/understanding-oauth2-a50f29f0fbf7>.
11. *Access Token Response*. OAuth, 2023-12. Dostupné také z: <https://www.oauth.com/oauth2-servers/access-tokens/access-token-response/>. [Online: accessed 9.Dec.2023].
12. *GoodData embedded analytics platform*. GoodData, 2023-12. Dostupné také z: <https://www.gooddata.com/embedded-analytics/>.

13. *What is GoodData?* TrustRadius, 2023-12. Dostupné také z: <https://www.trustradius.com/products/gooddata/reviews?qs=pros-and-cons#reviews>. [Online: accessed 9.Dec.2023].
14. PORRITT, Stephen. *Best Embedded Analytics Tools (2023)*. 2023-01-30. Dostupné také z: <https://technologyadvice.com/blog/information-technology/embedded-analytics-software/>.
15. *What is Microsoft Power BI Embedded?* TrustRadius, 2023-12. Dostupné také z: <https://www.trustradius.com/products/microsoft-power-bi-embedded/reviews?qs=pros-and-cons>. [Online: accessed 9.Dec.2023].
16. *Introduction to LookML*. Google Cloud, 2023-12. Dostupné také z: <https://cloud.google.com/looker/docs/what-is-lookml>. [Online: accessed 9.Dec.2023].
17. *What is Looker?* TrustRadius, 2023-12. Dostupné také z: <https://www.trustradius.com/products/looker/reviews>. [Online: accessed 9.Dec.2023].
18. SURVE, Suraj. *Why You Should Use React.js For Web Development*. 2021-02-18. Dostupné také z: <https://www.freecodecamp.org/news/why-use-react-for-web-development/>.
19. HUTSULYAK, Oleksandr. *10 Key Reasons Why You Should Use React for Web Development*. 2023-03-12. Dostupné také z: <https://www.techmagic.co/blog/why-we-use-react-js-in-the-development/>.
20. *What is Dom in React?* Javatpoint, 2023-12. Dostupné také z: <https://www.javatpoint.com/what-is-dom-in-react>. [Online: accessed 9.Dec.2023].

Seznam obrázků

2.1	Ukázka dashboardu v produktu firmy Emplifi	5
2.2	Autentizační diagram protokolu OAuth 2.0	8

Seznam tabulek

Seznam výpisů

2.1	Ukázková odpověď autorizačního serveru	7
-----	--	---

101011000011100010 1100001
1010110001 10001



11010001101101001
01100001 1010101
111000101011101