

Rozwój narzędzia rozszerzającego usługę Google Forms

(Development of an extension to Google Forms service)

Aleksandra Rozkrut

Praca inżynierska

Promotor: dr hab. Jan Otop, prof. UWr

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

Wrocław 2022

Streszczenie

Praca jest kontynuacją pracy Agnieszki Pawickiej „Implementacja narzędzia rozszerzającego usługę Google Forms”. Zawiera implementację oraz opis rozwijanego rozszerzenia do usługi Google Forms. Rozszerzenie to pozwala na automatyczne generowanie formularzy z odpowiedniego pliku w formacie JSON, konwertowanie wstawek matematycznych napisanych w L^AT_EX do odpowiednich symboli oraz zarządzanie niektórymi własnościami utworzonych uprzednio formularzy. W pracy znajduje się również omówienie wykorzystanych technologii.

Spis treści

1. Wprowadzenie	7
1.1. Wprowadzone usprawnienia	8
2. Środowisko	9
2.1. Interfejs użytkownika	9
2.1.1. React	9
2.1.2. Material UI	9
2.1.3. axios	10
2.1.4. date-fns	10
2.1.5. react-code-blocks	10
2.1.6.	10
2.2. Serwer	10
2.3. Vagrant i Ansible	10
2.4. Google Forms API oraz Imgur API	10
3. Instrukcja użytkownika	11
3.1. Początki pracy z narzędziem	11
3.1.1. Projekt na Google Cloud	11
3.1.2. Imgur API	12
3.1.3. Repozytorium	13
3.1.4. Instalacja z Vagrant	13
3.1.5. Instalacja bez Vagrant	13
3.2. Praca z narzędziem	14

3.2.1. Schemat pliku kodującego (JSON)	14
3.2.2. Automatyczne ocenianie	16
3.2.3. Obsługa narzędzia	16
3.2.4. Przeprowadzanie egzaminów	18

Rozdział 1.

Wprowadzenie

Wraz z rozpoczęciem pandemii problem zdalnego sprawdzania wiedzy i umiejętności zrobił się popularny. Miejsca, w których było to szczególnie uciążliwe to m. in. szkoły, uczelnie, czy procesy rekrutacyjne w różnych firmach. Dobre narzędzie do przeprowadzania zdalnych testów powinno spełniać pewne wymagania, takie jak:

- przejrzystość formularzy,
- łatwość tworzenia nowych testów,
- możliwość skutecznego weryfikowania tożsamości osoby piszącej test.

Dodatkowo dobrze jest, gdy takie narzędzie posiada również takie funkcje jak:

- automatyczne sprawdzanie pytań zamkniętych,
- kontrolowanie czasu potrzebnego poszczególnym osobom na zakończenie testu,
- możliwość losowych zmian kolejności pytań,
- możliwość zapisywania odpowiedzi w popularnym formacie.

Ponadto niektórzy egzaminujący cenią sobie możliwość wstawiania symboli matematycznych, wykresów czy zdjęć w ramach pytań.

We wrześniu 2021 roku Agnieszka Pawicka obroniła pracę inżynierską o tytule „Implementacja narzędzia rozszerzającego usługę Google Forms”, w której zaproponowała system umożliwiający generowanie testów Google Form z pliku w formacie JSON. Projekt umożliwiał także generowanie pytań zawierających wstawki w \LaTeX , które były automatycznie kompilowane i wstawiane do formularza.

Celem tej pracy jest usprawnienie działania wymienionego wyżej narzędzia.

1.1. Wprowadzone usprawnienia

W pracy został zaimplementowany szereg modyfikacji, które ułatwiają korzystanie z narzędzia zarówno przy tworzeniu formularzy, zarządzaniu nimi, jak i ich późniejszym sprawdzaniu. Te usprawnienia to:

- łatwiejszy sposób instalacji narzędzia,
- zmiana interfejsu użytkownika na bardziej intuicyjny,
- zmiana sposobu wgrywania zakodowanego formularza,
- możliwość edycji pliku JSON z zakodowanym formularzem już po wgraniu go,
- możliwość zobaczenia odpowiedzi oraz wyników w formacie JSON,
- możliwość pobrania danych o przesłanych odpowiedziach i wynikach sprawdzania automatycznego w formacie EXCEL,
- utworzone formularze są teraz przechowywane na Dysku Google użytkownika - nie jak w poprzedniej wersji na wspólnym dysku dla wszystkich użytkowników,
- szereg nowych opcji związanych z parametrami formularza tj.:
 - możliwość zmiany kolejności odpowiedzi wewnątrz pytań zamkniętych,
 - możliwość dodania opisu testu,
 - odpowiedzi do pytań mogą być teraz w formie symboli matematycznych (\LaTeX),
 - możliwe jest dodanie oceniania innego niż zerojedynekowe dla pytań wielokrotnego wyboru.

Wszystkie wymienione zmiany wraz z użytymi narzędziami, instrukcją użytkownika oraz opisem środowisk i napotkanych problemów są opisane w dalszych rozdziałach niniejszej pracy.

Rozdział 2.

Środowisko

Praca jest rozszerzeniem innego projektu, wobec tego wiele użytych bibliotek zostało opisanych w rozdziale „Środowisko” pracy „Implementacja narzędzia rozszerzającego usługę Google Forms”. W niniejszym rozdziale skupię się więc na nowo dodanych elementach środowiska. W celu zapoznania się z całością polecam lekturę wymienionej pracy.

Kod aplikacji można podzielić na trzy główne części:

1. interfejs użytkownika,
2. serwer komunikujący się z zewnętrznymi API oraz bazą danych,
3. kod służący do instalacji zależności i uruchamiania aplikacji na maszynie wirtualnej - korzystający z technologii Vagrant i Ansible.

Dodatkowo aplikacja korzysta z Google Forms API oraz Imgur API.

2.1. Interfejs użytkownika

2.1.1. TypeScript

Aplikacja z interfejsem użytkownika została napisana w języku TypeScript i korzysta z bibliotek: types/node, typescript i web-vitals.

2.1.2. React

Technologia służąca do tworzenia interfejsów graficznych aplikacji internetowych. Aplikacja korzysta z bibliotek: emotion/react, @emotion/styled, types/react, types/react-dom, react, react-dom, react-scripts.

2.1.3. Material UI

Biblioteka, która udostępnia wiele gotowych komponentów, których można użyć w aplikacjach korzystających z Reacta.

2.1.4. axios

Biblioteka służąca do wysyłania zapytań http i pobierania danych z serwera.

2.1.5. date-fns

Biblioteka pomagająca zarządzać datami. Udostępnia wiele użytecznych funkcji.

2.1.6. react-code-blocks

Biblioteka z gotowymi komponentami służącymi do wyświetlania bloków kodu.

2.2. Serwer

2.3. Vagrant i Ansible

2.4. Google Forms API oraz Imgur API

Google Forms API

Imgur API

Rozdział 3.

Instrukcja użytkownika

3.1. Początki pracy z narzędziem

Zanim zaczniesz się pracować z narzędziem, należy zainstalować wszystkie wykorzystywane przez nie zależności. Instalacja została usprawniona wykorzystując Vagrant oraz Ansible, ale można ją przeprowadzić również bez tych technologii. Oba te sposoby oraz późniejsza praca z narzędziem, która zależy od wybranego sposobu instalacji są opisane poniżej.

3.1.1. Projekt na Google Cloud

Aplikacja używa Google Forms API do zarządzania formularzami na dysku Google użytkownika i z tego powodu potrzebuje danych uwierzytelniających do komunikacji z tym API. Można je uzyskać zakładając projekt na Google Cloud. Dokładne instrukcje można znaleźć w poradnikach w dokumentacji Google, a w tej pracy spróbuję krótko opisać ten proces:

- zaloguj się do konsoli Google Cloud,
- znajdź przycisk „Wybierz Projekt”,
- w nowym oknie naciśnij przycisk „Nowy Projekt”,
- wpisz nazwę projektu oraz lokalizację, a następnie naciśnij „Utwórz”,
- w menu po lewej stronie wybierz „Wyświetl Wszystkie Usługi”,
- w sekcji „Zarządzanie” wybierz „Interfejsy API i usługi”,
- wybierz swój projekt,
- kliknij „Interfejsy API i usługi”,

- wyszukaj „Google Forms API”,
- w szczegółach Google Forms API kliknij „Włącz”,
- w menu po lewej stronie wybierz „Dane logowania”,
- kliknij „Create Credentials”, a potem „Klucz interfejsu API”,
- zapisz klucz w pliku i zachowaj na później,
- ponownie kliknij „Create Credentials”, a potem „Identyfikator Klienta OAuth”,
- kliknij „Skonfiguruj Ekran Zgody”,
- jako typ użytkownika wybierz „Zewnętrzny” i kliknij „Utwórz”,
- wpisz nazwę aplikacji, adresy e-mail i kliknij „Zapisz i kontynuuj”,
- kliknij „Dodaj lub usuń zakres”,
- w „Filtruj” wpisz „Google Forms API”,
- zaznacz zakresy „forms.body” i „forms.responses.readonly”,
- kliknij „Zaktualizuj”, następnie „Zapisz i kontynuuj”,
- dodaj swój adres konta Google do użytkowników testowych, kliknij „Zapisz i kontynuuj” i na końcu „Powrót do panelu”,
- wybierz ponownie „Dane Logowania”, „Create Credentials” i „Identyfikator Klienta OAuth”
- jako typ aplikacji wybierz „Aplikacja internetowa”, dodaj URI `http://localhost:3000/oauth2callback` do sekcji „Autoryzowane identyfikatory URI przekierowania” i kliknij „Utwórz”,
- wybierz „Pobierz JSON”.

3.1.2. Imgur API

Aplikacja używa Imgur API, żeby wygenerować linki zewnętrzne obrazków z formułami matematycznymi, z których pobiera je serwer Google. Do komunikacji z Imgur API również potrzebujemy danych uwierzytelniania:

- zaloguj się do Imgur,
- odwiedź stronę Imgur API
- znajdź „Register an application”,
- wypełnij dane, jako typ autoryzacji wybierz „OAuth2 authorization without a callback URL”,
- skopiuj Client ID i Client secret i zachowaj na później

3.1.3. Repozytorium

Teraz należy sklonować repozytorium projektu — `github.com/arozkrut/forms-app`. Następnie w pobranym repozytorium w katalogu `forms-app/src/credentials` stwórz pliki:

- `api_key.txt`: wklej tu klucz interfejsu API, tak żeby plik był postaci

```
key=<uzyskany klucz>
```

- `credentials.json`: wklej tu zawartość pobranego pliku JSON,
- `ingur_credentials.txt`: wklej tu Client ID i Client secret, tak żeby plik był postaci

```
ClientID: <Client ID>
ClientSecret: <Client secret>
```

3.1.4. Instalacja z Vagrant

Jeśli masz już zainstalowany Vagrant wystarczy, że w głównym katalogu projektu (tym z `Vagrantfile`) użyjesz polecenia `vagrant up`. Skrypt Ansible zainstaluje wszystkie potrzebne zależności i uruchomi wszystkie aplikacje. Ważne jest, że projekt potrzebuje portów 3000, 9090 i 9091. Jeśli któryś z nich jest już zajęty narzędzie nie uruchomi się. Instalacja może chwilę potrwać. Po jej zakończeniu wyświetli się wiadomość z linkiem prowadzącym do strony uwierzytelniania Google, gdzie trzeba wybrać konto Google, na którym będą przechowywane formularze oraz wyrazić zgodę na nadanie aplikacji dostępu do formularzy na tym koncie Google. Następnie w przeglądarce należy odwiedzić adres `http://localhost:9091`, gdzie działa interfejs użytkownika. Narzędzie jest gotowe do pracy.

3.1.5. Instalacja bez Vagrant

Jeśli użytkownik chce korzystać z narzędzia bez zainstalowanego Vagrant, musi on sam zainstalować wszystkie zależności: Python 3, L^AT_EX, poppler-utils, biblioteki do języka Python opisane w poprzednim rozdziale, nodejs, npm. Następnie w katalogach `forms-app` oraz `forms-app-ui` należy wydać polecenie `npm i`. Teraz należy włączyć serwer NodeJS poleceniem `node src/server.js` (koniecznie w katalogu `forms-app`). W przeglądarce powinna uruchomić się strona uwierzytelniania Google, gdzie trzeba wybrać konto Google, na którym będą przechowywane formularze oraz wyrazić zgodę na nadanie aplikacji dostępu do formularzy na tym koncie Google. Po tym w katalogu `src/forms-app-ui` należy włączyć serwer interfejsu użytkownika

poleceniem `npm start`, który uruchomi się pod adresem `http://localhost:9091`. Po odwiedzeniu tej strony w przeglądarce można zacząć pracować z narzędziem.

3.2. Praca z narzędziem

3.2.1. Schemat pliku kodującego (JSON)

Poniżej znajduje się rozpisany schemat kodowania:

```
1 {
2   "title": "string",
3   "description": "string",
4   "startDate": "string",
5   "endDate": "string",
6   "shuffleAnswers": "boolean",
7   "questions": [{
8     "type": "string",
9     "text": "string",
10    "tex" : "boolean",
11    "answers": [{
12      "text" : "string",
13      "tex" : "boolean",
14      "correct" : "boolean"
15    }],
16    "points": "number"
17    "pointsArray": "array"
18  }]
19 }
```

Jest to opis tych wartości, które mogą być użyte — nie wszystkie są jednak wymagane. Poniżej znajduje się szczegółowy opis pól i wartości:

- `title` – wartość tekstowa, odpowiada nagłówkowi formularza — **pole wymagane**,
- `description` – wartość tekstowa, krótki opis formularza, który pojawi się pod tytułem — **pole opcjonalne**,
- `startDate` – wartość tekstowa, data planowanego rozpoczęcia testu w formacie ISO. To pole służy tylko do odrzucania odpowiedzi, które zostały przesłane przed podaną datą podczas oceniania — **pole opcjonalne**,

- `endDate` – wartość tekstowa, data planowanego zakończenia testu w formacie ISO. To pole służy tylko do odrzucania odpowiedzi, które zostały przesłane po podanej dacie podczas oceniania — **pole opcjonalne**,
- `shuffleAnswers` – wartość boolowska, informacja o tym, czy odpowiedzi dla każdego pytania powinny zostać przetasowane dla każdego wyświetlenia formularza. Niestety na ten moment Google Forms API nie udostępnia opcji tasowania pytań, można ją jedynie włączyć ręcznie w ustawieniach formularza na stronie Google: settings -> presentation -> Shuffle question order — **pole opcjonalne**,
- `questions` – tablica, której każde pole zawiera informacje dotyczące jednego pytania — **pole wymagane**.

Poniżej znajdują się wartości kodujące pojedyncze pytanie:

- `type` – pole tekstowe dotyczące typu kodowanego pytania. Dopuszczalne wartości:
 - * „checkbox” – pytanie zamknięte wielokrotnego wyboru,
 - * „grid” – pytanie typu „prawda/fałsz”,
 - * „list” – zamknięte jednokrotnego wyboru,
 - * „text” – otwarte.

pole wymagane,

- `text` – zawiera treść pytania w formie tekstowej. Może zawierać wstawki z \LaTeX a. Należy jednak pamiętać, że JavaScript traktuje symbol „\” jako specjalny — wszystkie wystąpienia „\” należy więc zastąpić „\\” — **pole wymagane**,
- `tex` – wartość boolowska, jeśli **true** treść pytania (`text`) będzie konwertowana do obrazu z zachowaniem konwersji symboli matematycznych i innych wstawek z \LaTeX a z biblioteki standardowej — **pole wymagane**,
- `answers` – tablica, każde pole zawiera jedną z możliwych odpowiedzi w następującej formie:
 - * `text` – pole tekstowe, treść danej odpowiedzi,
 - * `tex` – wartość boolowska, jeśli **true** treść odpowiedzi (`text`) będzie konwertowana do obrazu z zachowaniem konwersji symboli matematycznych i innych wstawek z \LaTeX a z biblioteki standardowej, tylko dla pytań typu `list` i `checkbox`,
 - * `correct` – wartość boolowska wskazująca czy dana odpowiedź jest prawidłowa. Domyślna wartość: `false`.

— **pole opcjonalne**,

- `points` – wartość numeryczna, odpowiada liczbie punktów przyznawanej za poprawą odpowiedź na pytanie (tylko dla pytań typu `list`) — **pole opcjonalne**,

- `pointsArray` – tablica z wartościami numerycznymi, odpowiada liczbie punktów przyznawanej za poprawne odpowiedzi na pytanie (tylko dla pytań typu `checkBox` i `grid`), przykładowo wartość `[0, 1, 3]` wskazuje, że jeśli nie zostanie wybrana żadna dobra odpowiedź, zostanie przyznane 0 punktów, jeśli zostanie udzielona jedna dobra odpowiedź, zostanie przydzielony 1 punkt i jeśli zostaną udzielone dwie dobre odpowiedzi, zostaną przydzielone 3 punkty — **pole opcjonalne**

3.2.2. Automatyczne ocenianie

Pytania typu `list`, `checkBox` i `grid` mogą zostać automatycznie ocenione i wyniki mogą zostać pobrane w formacie JSON lub pliku Excel. Jeśli chcemy, żeby odpowiedzi zostały ocenione musimy ręcznie ustawić zbieranie adresów email w ustawieniach formularza. Google Forms API niestety na ten moment nie udostępnia możliwości zrobienia tego automatycznie. Sposób oceniania zależy od typu pytania:

- **list**: jeśli zostanie zaznaczona poprawna odpowiedź, odpowiadający uzyska całość punktów z pola `points`, w przeciwnym przypadku otrzyma 0 punktów,
- **checkBox**: zliczane są poprawnie zaznaczone odpowiedzi i od tej liczby odejmuje się liczbę niepoprawnie zaznaczonych odpowiedzi. Jeśli uzyskana liczba jest mniejsza od zera, podstawiamy za nią 0. Następnie z tablicy `pointsArray` z indeksu równego poprzednio uzyskanej liczbie odczytuje się liczbę uzyskanych punktów,
- **grid**: zliczane są poprawnie udzielone odpowiedzi i z indeksu równego tej liczbie odczytuje się liczbę uzyskanych punktów w tablicy `pointsArray`.

3.2.3. Obsługa narzędzia

Po uruchomieniu użytkownik widzi stronę w przeglądarce, jak na zdjęciu poniżej:

STWÓRZ NOWY DODAJ ISTNIEJĄCY

Shuffle Answers
Rozpoczyna się: nie ustawiono
Kończy się: nie ustawiono
OTWÓRZ EDYTUJ W GOOGLE EDYTUJ USUŃ ODPOWIEDZI OCENY JSON POBIERZ EXCEL
Rozwiń template formularza

Tex Form
Rozpoczyna się: nie ustawiono
Kończy się: nie ustawiono
OTWÓRZ EDYTUJ W GOOGLE EDYTUJ USUŃ ODPOWIEDZI OCENY JSON POBIERZ EXCEL
Rozwiń template formularza

Form Title
Form description can be looooooong
Rozpoczyna się: sobota, 10 września 2022 17:00
Kończy się: sobota, 10 września 2022 19:00
OTWÓRZ EDYTUJ W GOOGLE EDYTUJ USUŃ ODPOWIEDZI OCENY JSON POBIERZ EXCEL
Rozwiń template formularza

Another Form Title
Rozpoczyna się: nie ustawiono
Kończy się: nie ustawiono
OTWÓRZ EDYTUJ W GOOGLE EDYTUJ USUŃ ODPOWIEDZI OCENY JSON POBIERZ EXCEL
Rozwiń template formularza

Rysunek 3.1: Interfejs aplikacji

Przycisk „Stwórz nowy” otworzy dialog, w którym należy wprowadzić kodowanie formularza i ewentualnie czas rozpoczęcia i zakończenia testu. Po kliknięciu „Stwórz” na dysku Google zostanie stworzony nowy formularz i wpis o nim pojawi się na liście na głównej stronie aplikacji. Przycisk „Dodaj istniejący” otworzy dialog, gdzie możemy podać identyfikator istniejącego już formularza, który chcemy dodać do listy formularzy zarządzanych przez aplikację. Musimy również podać jego kodowanie. Po kliknięciu „Dodaj” formularz zostanie zedytowany i pojawi się na liście formularzy. Na liście formularzy każdy wpis odpowiada jednemu formularzowi na dysku Google użytkownika, jednak nie wszystkie formularze na dysku się tu pojawiają, jedynie te stworzone przez aplikację lub dodane używając przycisku „Dodaj istniejący”. W każdym wpisie możemy znaleźć przyciski:

- „Otwórz”: otwiera formularz na stronie Google w nowej karcie,
- „Edytuj w Google”: otwiera stronę edycji formularza na stronie Google w nowej karcie,
- „Edytuj”: otwiera dialog, gdzie można edytować kodowanie formularza,

- „Usuń”: usuwa formularz z listy formularzy zarządzanych przez aplikację i usuwa całą jego zawartość, jednak pusty formularz tylko z tytułem pozostanie na dysku Google użytkownika,
- „Odpowiedzi”: pokazuje wszystkie przesłane odpowiedzi, łącznie z tymi przesłanymi przed początkiem lub po końcu ustawionych w kodowaniu formularza,
- „Oceny JSON”: wyświetla ocenione odpowiedzi bez pytań otwartych w formacie JSON,
- „Pobierz Excel”: pobiera plik Excel z ocenionymi zamkniętymi pytaniami,
- „Rozwiń template formularza”: podgląd kodowania formularza,

3.2.4. Przeprowadzanie egzaminów

Jeśli narzędzie ma być użyte do przeprowadzenia egzaminu na początku należy stworzyć z jego pomocą pusty formularz tylko z tytułem, ewentualnie krótką informacją o czasie startu. Przed rozpoczęciem należy ręcznie włączyć w ustawieniach formularza na stronie Google zbieranie adresów email i ewentualnie tasowanie pytań. Następnie w momencie rozpoczęcia egzaminu należy go zaktualizować dodając do niego pytania. Po czasie zakończenia egzaminu można go ręcznie zamknąć, ale jeśli ma zostać użyte automatyczne ocenianie, nie wolno zmieniać jego kodowania w aplikacji. Narzędzie przefiltruje przesłane odpowiedzi i automatycznie je oceni po kliknięciu odpowiedniego przycisku.