Projektowanie aplikacji webowych

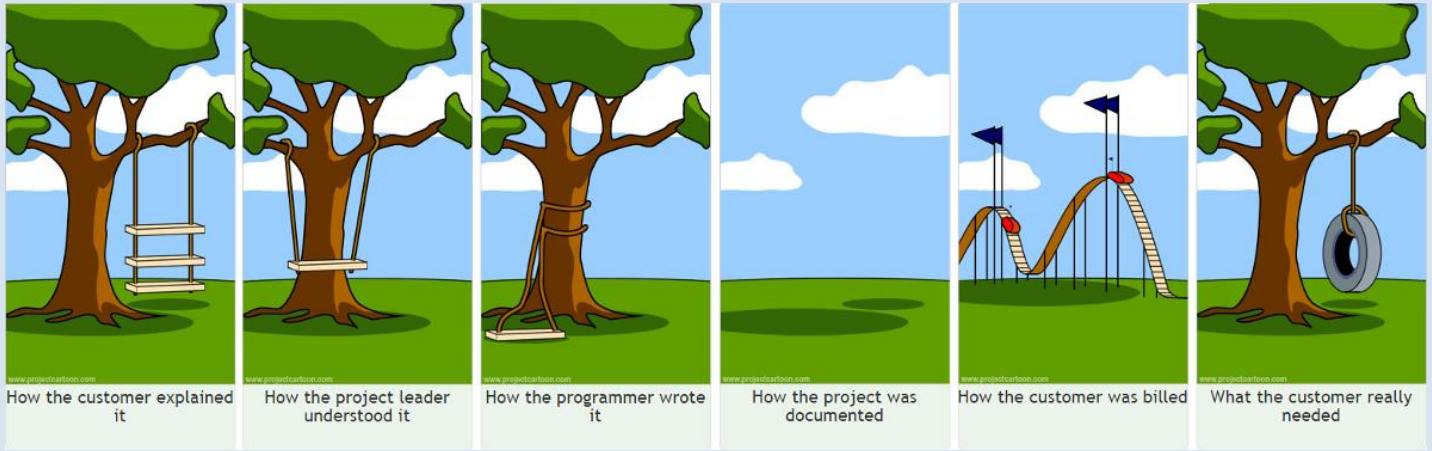
Sebastian Słomian

Sprawy organizacyjne

- Zaliczenie – test jednokrotnego wyboru, dostępny na SAKE od 13.05.2024 8.00 do 09.06.2024 23:59, 10 pytań 30 minut – 50% na zaliczenie.
- Nagrania zajęć zamieszczone będą na stronie SAKE maksymalnie do tygodnia od zakończenia zajęć
- Główna przerwa 13.00 – 14.00

Memory

Aplikacja do nauki słówek



01

Zbieranie wymagań

Cel projektu

Stworzenie aplikacji wspomagającej naukę języka obcego poprzez wielokrotne powtarzanie słów.

Aktorzy

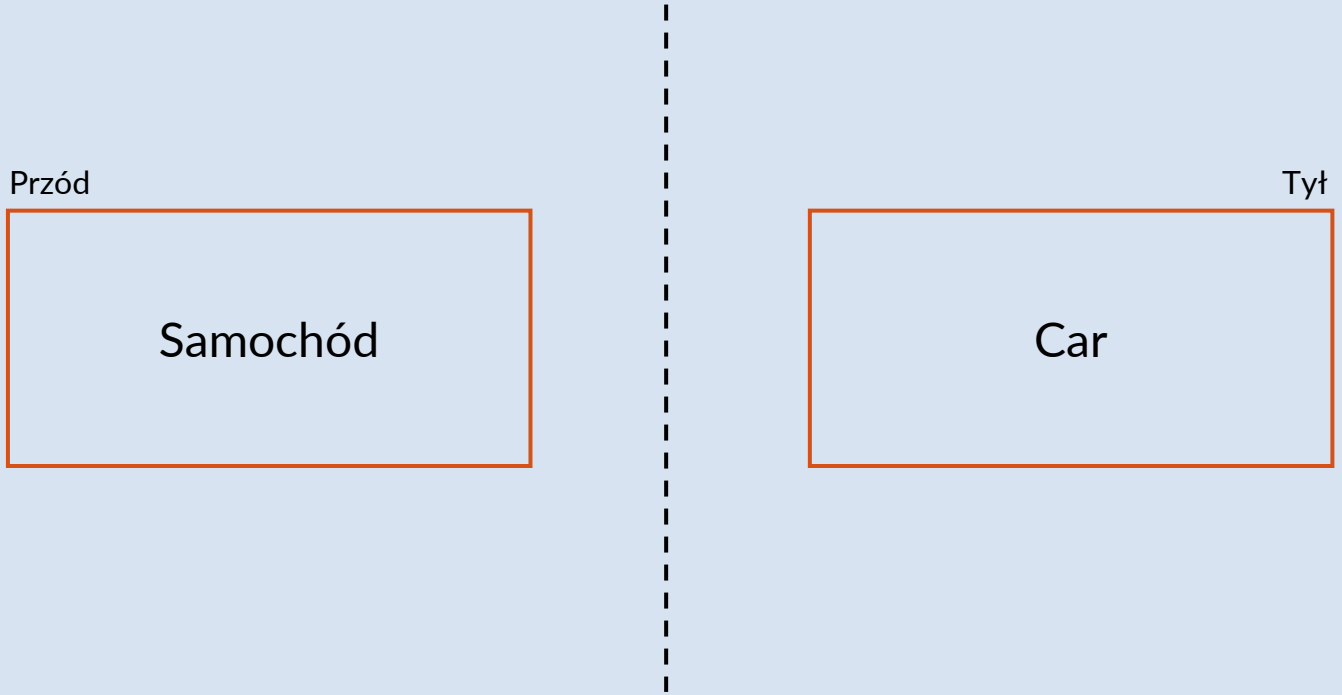
Student – osoba korzystająca z aplikacji, ucząca się języka obcego

Wymagania

- Jako Student, chce mieć możliwość dodawania oraz usuwania kategorii fiszek
- Jako Student, chce mieć możliwość przeglądania kategorii fiszek
- Jako Student, chcę mieć możliwość dodawania i usuwania fiszek do/z kategorii
- Jako Student, chcę mieć możliwość przeglądania fiszek z kategorii
- Jako Student, chce mieć możliwość wykonania sesji nauki słówek poprzez wyświetlanie pojęcia z fiszki, a następnie wyświetlenie definicji z fiszki i wykonanie samooceny czy pojęcie jest znane lub nie
- Jako Student, chcę znać rezultat po wykonanej każdej sesji nauki

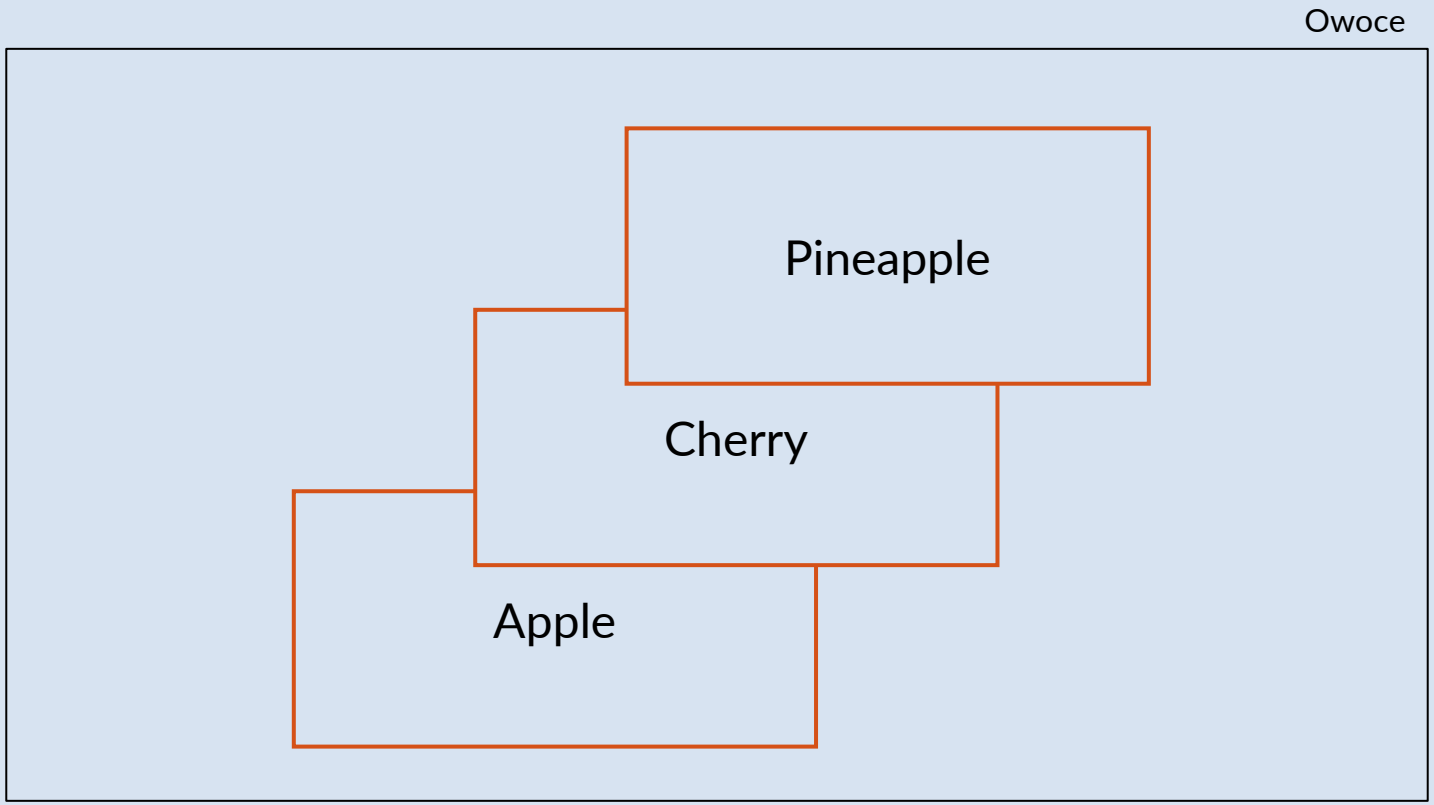
Fiszka (karta)

Pytanie wraz z odpowiedzią. Składa się z pojęcia oraz definicji



Kategoria

Zbiór fiszek należących do wspólnej grupy pod względem pewnej cechy



02

Planowanie

Wybór technologii

Backend

SQLAlchemy

Frontend

tailwindcss

Baza danych

Strona internetowa
vs
Aplikacja webowa

Strona internetowa

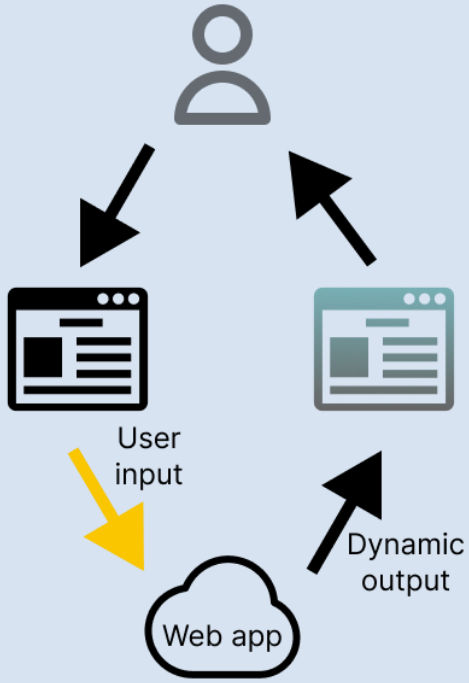
- podstawowy cel to wyświetlić informację
- zbiór usystematyzowanych i linkowanych między sobą dokumentów html, css i js znajdujących się na jednej domenie
- statyczne dane, każdy użytkownik widzi te same dane
- mogą zawierać elementy mini aplikacji webowych np. lokowanie na stronie map Google
- najczęściej są to blogi, strony z wydarzeniami, wizytówki firm

Aplikacja webowa

- podstawowy cel interakcja z użytkownikiem
- napisana przy użyciu języków programowania np. Java, Python, PHP oraz framework'ów
- prezentowana zawartość może zależeć od użytkownika
- składa się z klienta i części serwerowej
- łatwość aktualizacji w porównaniu z wersją desktopową

Website

Web App



Serwer aplikacji

Zapewnia podstawowe mechanizmy takie jak:

- zarządzanie wielowątkowością,
- dynamiczne ładowanie aplikacji,
- autoryzacja i uwierzytelnienie
- obsługę protokołu HTTP,
- szyfrowanie – SSL

Przykłady serwerów aplikacji:

Tomcat, Wildfly (JBoss), Glassfish, WebSphere, uWSGI, Gunicorn

Warstwa klienta

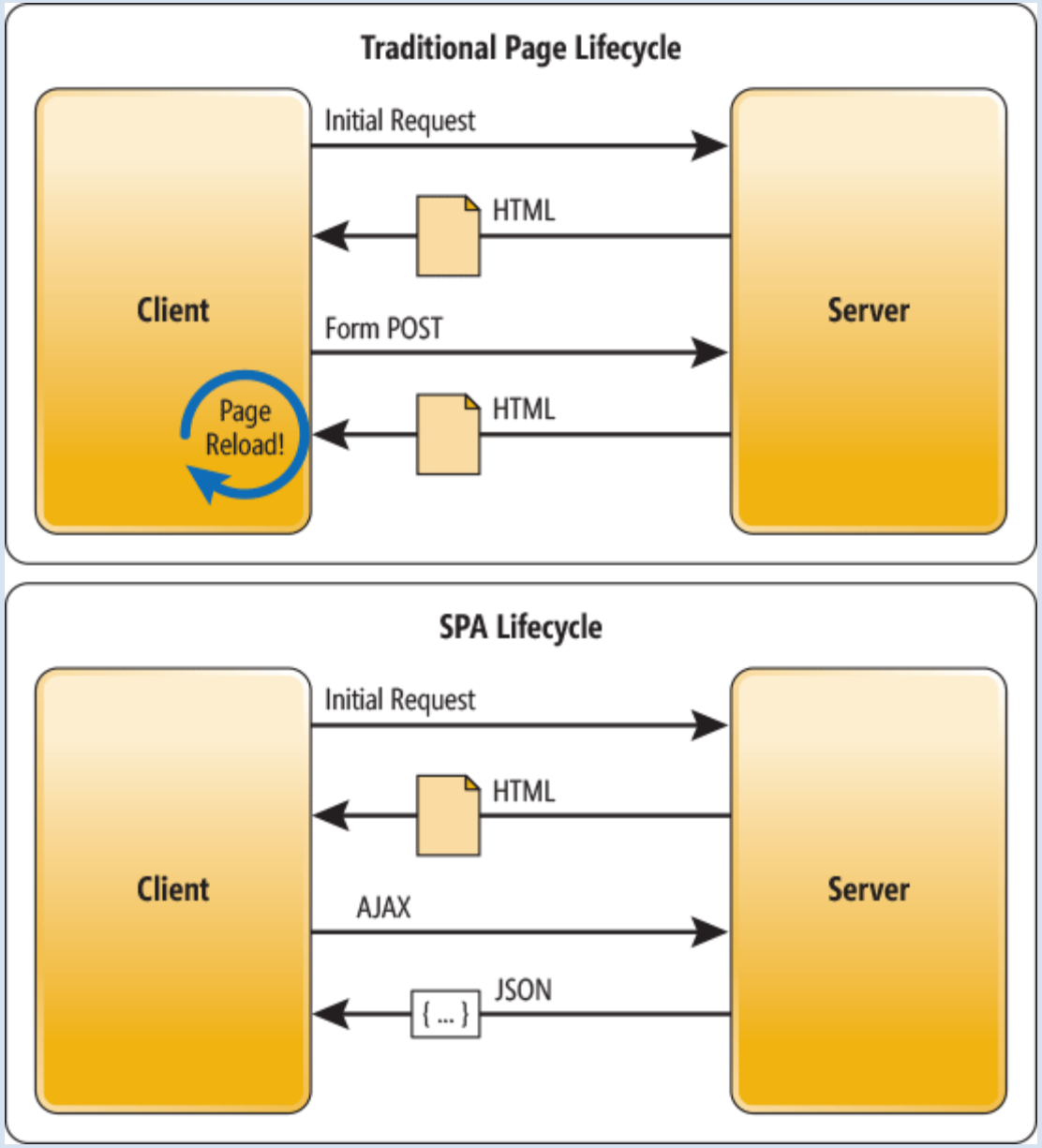
client side rendering
vs
server side rendering

Server side rendering

- działa w architekturze serwer MVC, dane są modelem, szablony widokiem a kontrolery przekazują dane do widoków
- konwersja szablonu HTML i uzupełnianie danymi po stronie serwera w wyniku czego otrzymujemy gotową stronę HTML
- przy każdej próbie przejścia na inna podstronę następuje ponowne żądanie do serwera o wygenerowanie kolejnej strony HTML
- minus - jedna strona jedno żądanie
- plus – wygodne pozycjonowanie w wyszukiwarkach internetowych, szybsze wczytywanie pojedynczej strony

Client side rendering

- zawartość strony jest generowana w przeglądarce za pomocą JavaScript
- dane z serwera pobierane są poprzez przygotowane punkty końcowe API (application programming interface)
- minus - ciężkie pozycjonowanie SEO, pierwsze wczytanie strony może zająć więcej czasu nić SSR
- plus - szybkość działania już po pierwszej inicjalizacji, szeroka gama interaktywności z użytkownikiem


Traditional Page Lifecycle
Client
Server
Initial Request
HTML
Form POST
HTML
Page Reload!
SPA Lifecycle
Client
Server
Initial Request
HTML
AJAX
JSON
{ ... }

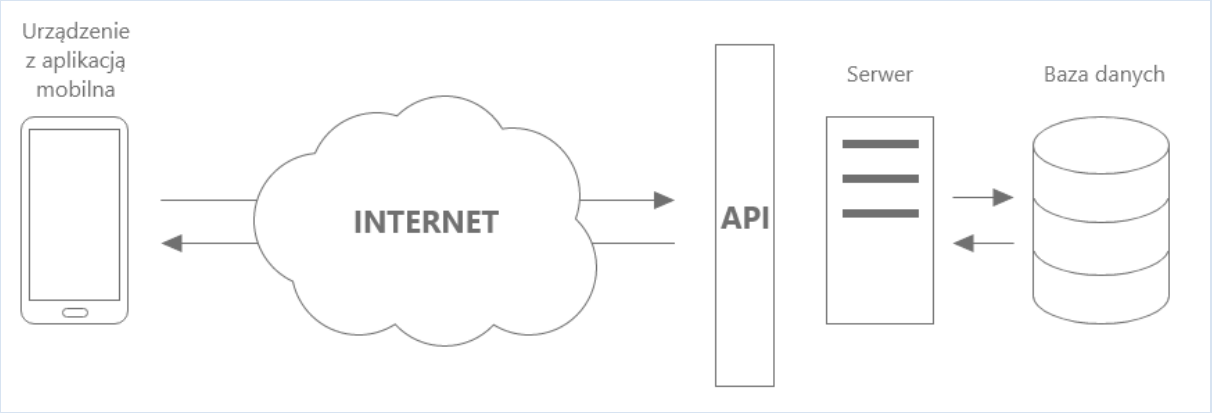
REST API

API

Application Programming Interface – za pomocą API integrujemy tworzone aplikacje z usługami innych dostawców.

Właściwości API:

- udostępniane funkcje,
- dostępność funkcji,
- ograniczenia techniczne tj. limity, sposób połączenia



HTTP

- Protokół przeznaczony do komunikacji w sieci WWW
- Najczęściej wykorzystywany do dwukierunkowej komunikacji pomiędzy przeglądarką internetową a serwerem WWW
- Komunikacja jest dwuetapowa: żądanie oraz odpowiedź na żądanie
- Bezstanowy protokół – każde żądanie jest niezależne i nie powiązane z żadnym innym
- Dla zwiększenia bezpieczeństwa stosuje się rozszerzenie protokołu – HTTPS

HTTP - zasoby

Lokalizacja zasobu składa się z następujących elementów:

- protokół – http / https,
- domena – nazwa, która identyfikuje jeden lub kilka adresów IP gdzie zlokalizowany zasób,
- ścieżka – określa lokalizację na serwerze,
- parametry – wykorzystywane do dodatkowej identyfikacji lub filtrowania.

HTTP - czasowniki

Podstawowe metody protokołu HTTP:

GET
odczyt

POST
zapis

PUT
aktualizacja

DELETE
usuwanie

HTTP – kody odpowiedzi

Podstawowe metody protokołu HTTP:

- 1xx – informujące,
- 2xx – żądanie zakończone powodzeniem,
- 3xx – klient został przekierowany do innego zasobu,
- 4xx – żądanie zawiera błędy,
- 5xx – błąd po stronie serwera

REST

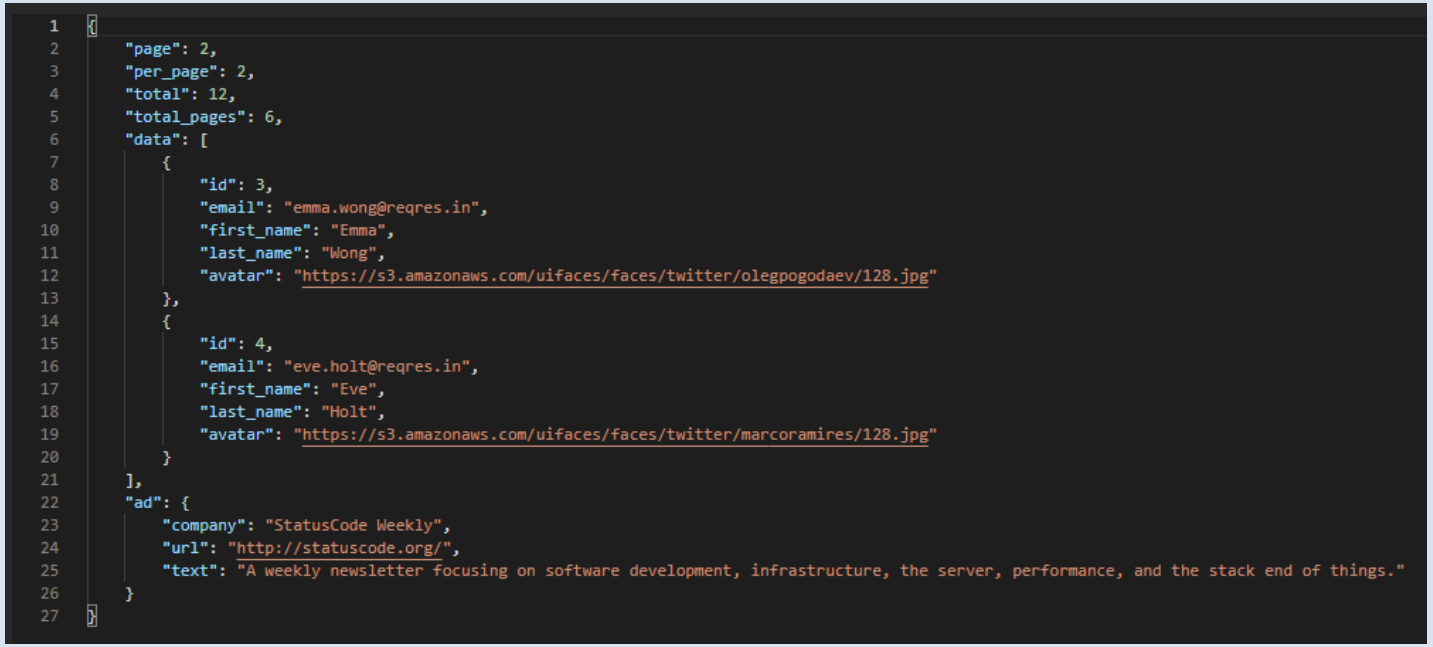
Representational State Transfer bazuje na API oraz HTTP. REST to styl architektury zgodny z takimi standardami sieciowymi jak czasowniki HTTP i identyfikatory URI.

Obowiązujące zasady:

- Wszystkie zasoby określa identyfikator URI.
- Każdy zasób może mieć liczne reprezentacje.
- Każdy zasób można pobrać zmodyfikować, utworzyć i usunąć standardowymi metodami HTTP.
- Na serwerze nie są przechowywane żadne informacje o stanie

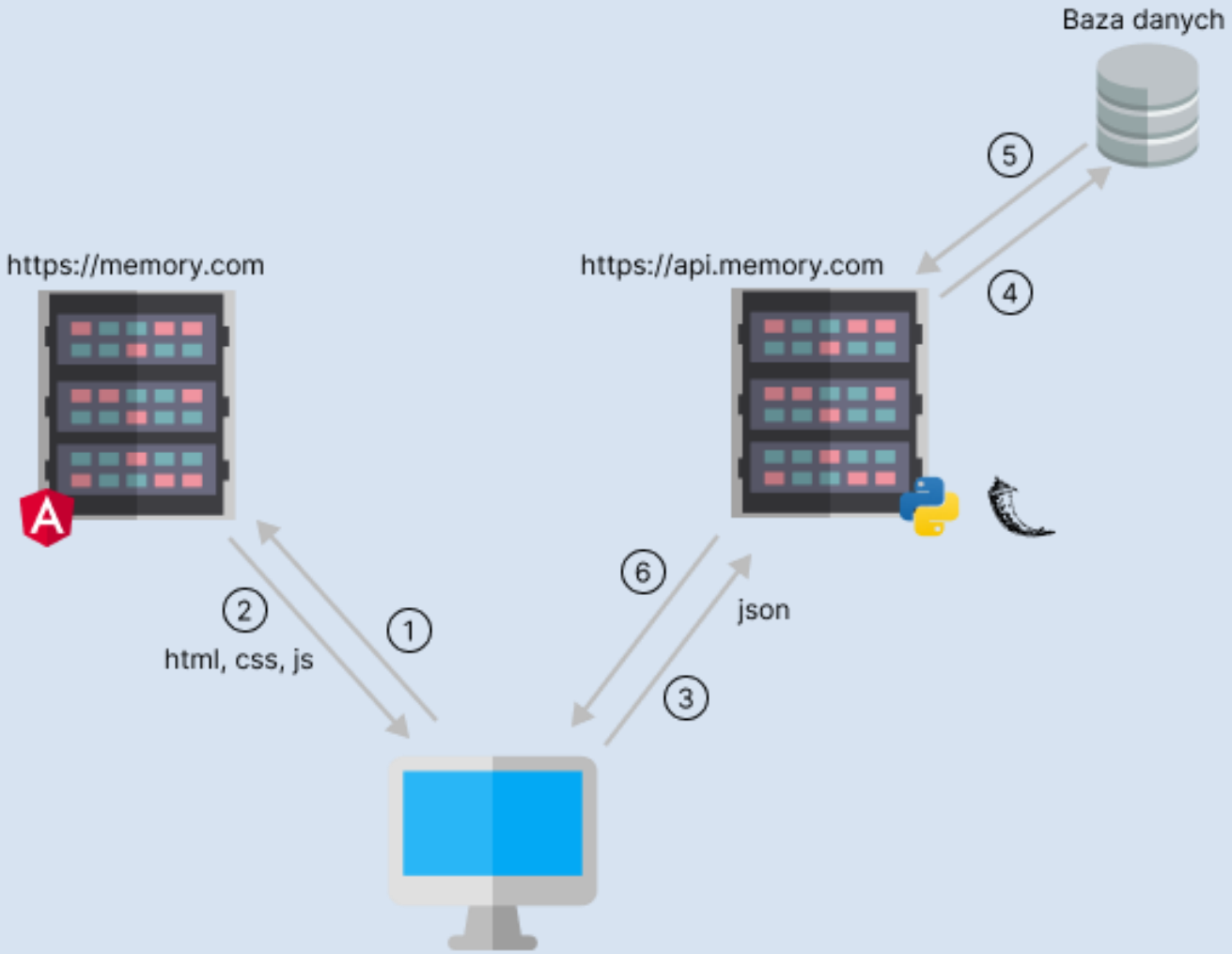
REST - format

W architekturze REST komunikacja przebiega przy użyciu formatów JSON, XML lub HTML, najczęściej jest wykorzystywany format JSON (JavaScript Object Notation).




```
{
    "page": 2,
    "per_page": 2,
    "total": 12,
    "total_pages": 6,
    "data": [
        {
            "id": 3,
            "email": "emma.wong@reqres.in",
            "first_name": "Emma",
            "last_name": "Wong",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/olegpogodaev/128.jpg"
        },
        {
            "id": 4,
            "email": "eve.holt@reqres.in",
            "first_name": "Eve",
            "last_name": "Holt",
            "avatar": "https://s3.amazonaws.com/uifaces/faces/twitter/marcoramires/128.jpg"
        }
    ],
    "ad": {
        "company": "StatusCode Weekly",
        "url": "http://statuscode.org/",
        "text": "A weekly newsletter focusing on software development, infrastructure, the server, performance, and the stack end of things."
    }
}
```

Architektura rozwiązania



03

Projektowanie

Rzeczowniki i ich atrybuty

Kategoria

Nazwa

Fiszka

Pojęcie

Definicja

UI

User Interface

Kolory

violet-800
#5b21b6

violet-300
#c4b5fd

stone-200
#e7e5e4

black
#000000

white
#ffffff

Memory

Cześć,

czym zajmiesz się dzisiaj?

Nauka

Opanuj nowe słówka

Baza słówek

Zarządzaj zbiorem swoich fiszek

Memory

← Powrót

Nie posiadasz żadnej kategorii

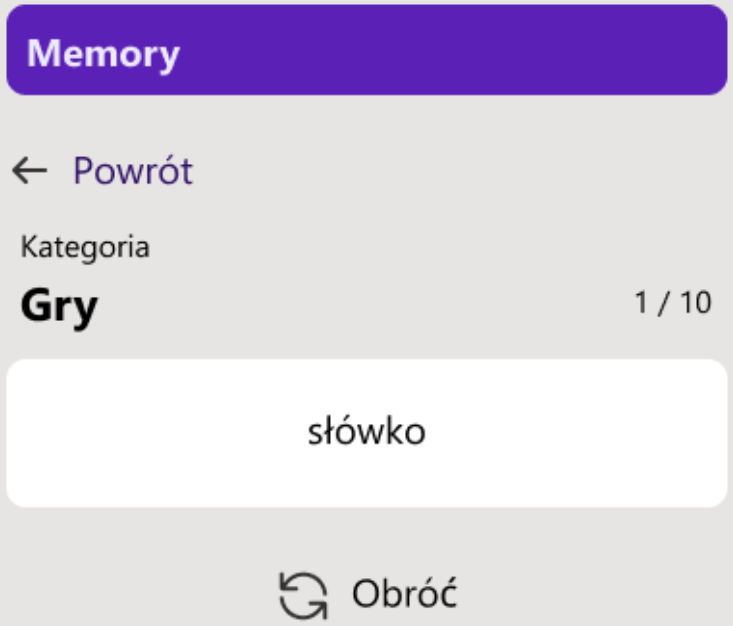
Memory

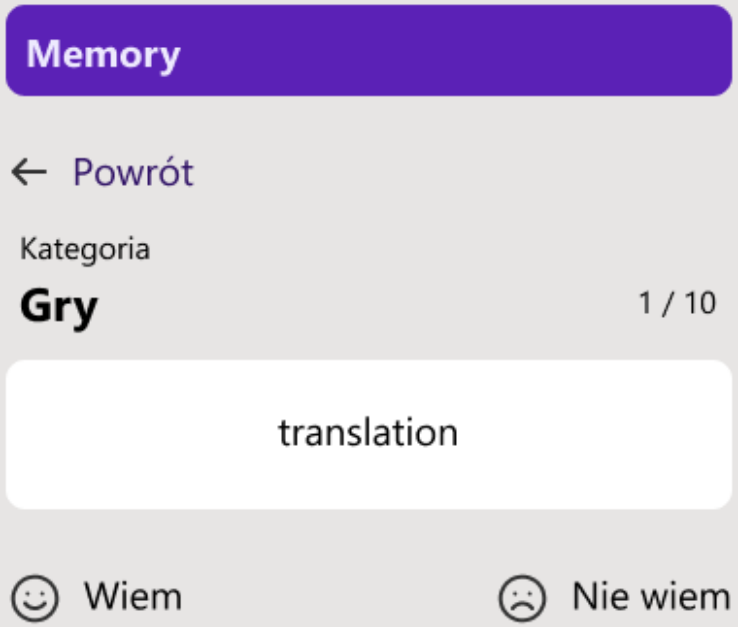
← Powrót

Wybierz kategorię słówek do ćwiczeń

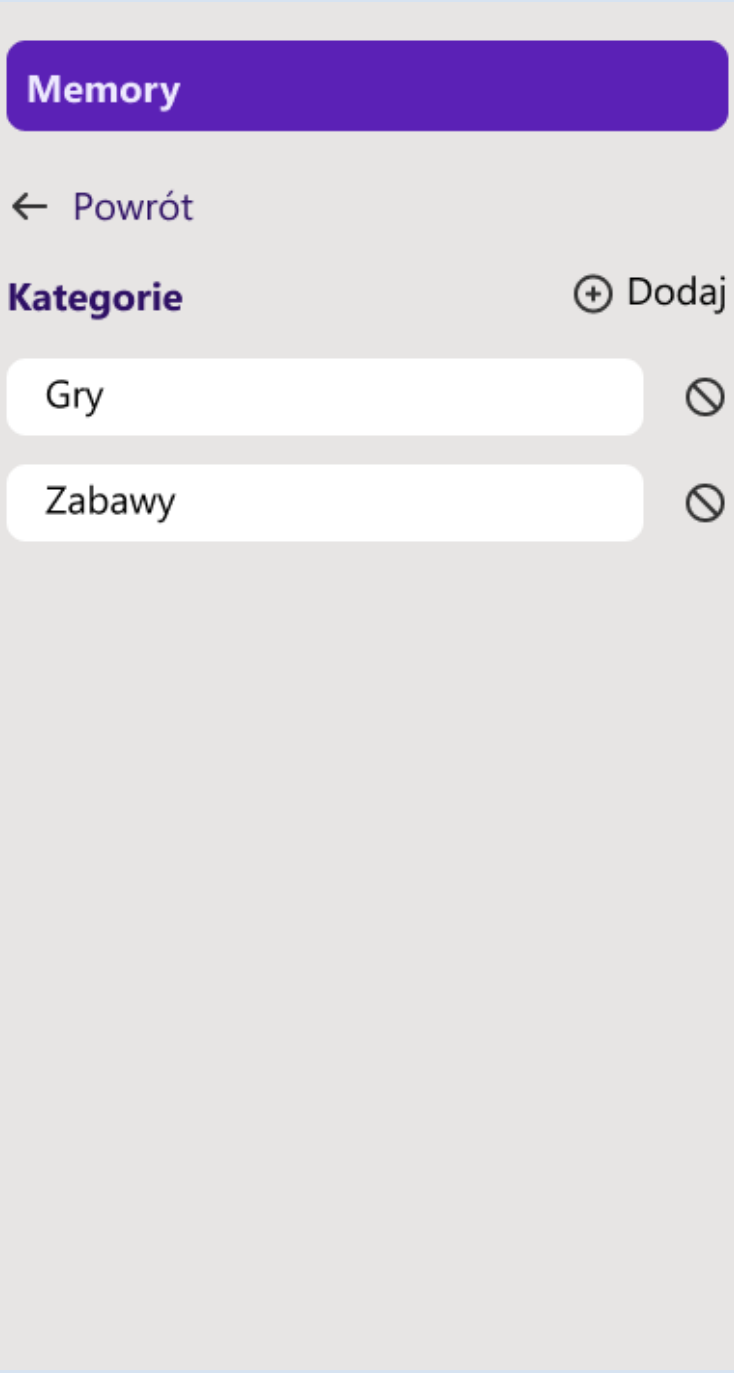
Gry

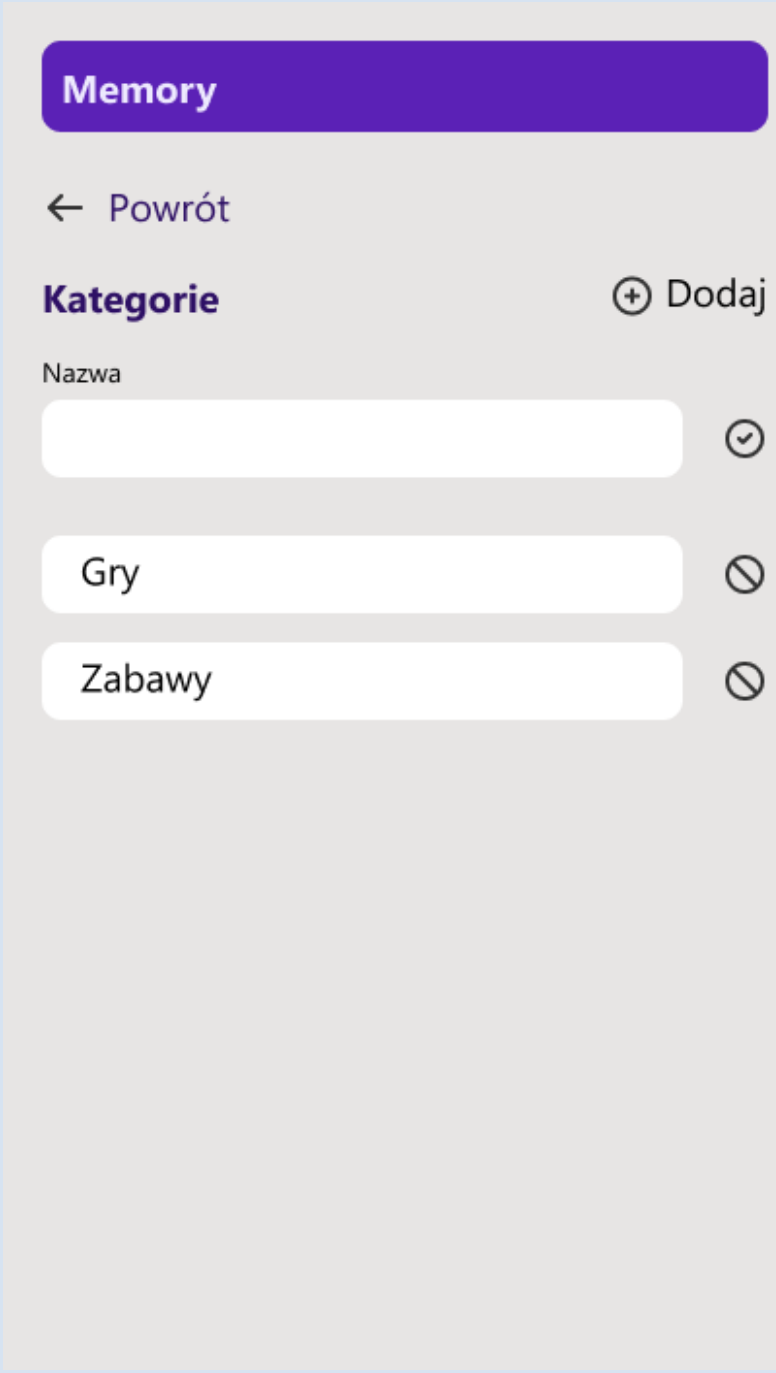
Zabawy


Memory
Powrót
Kategoria
Gry
1 / 10
słówko
Obróć

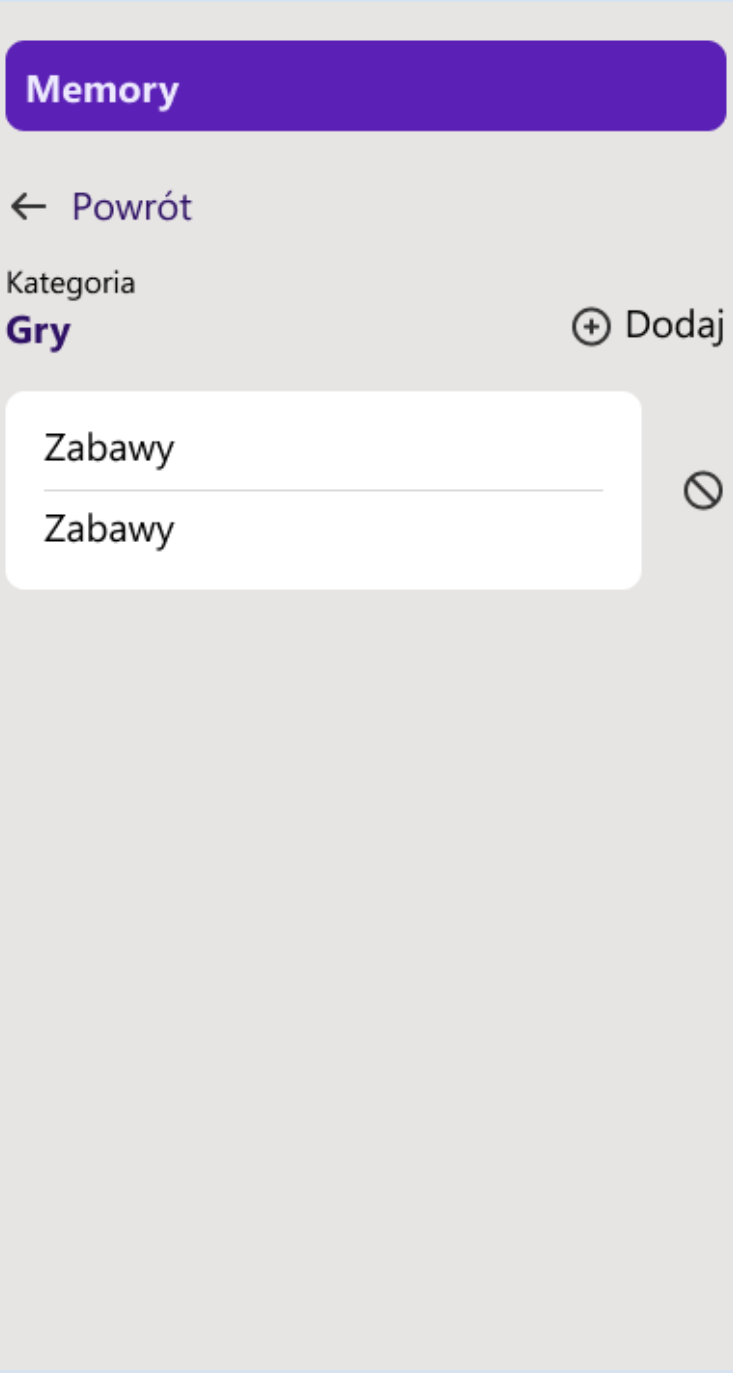

Memory
Powrót
Kategoria
Gry
1 / 10
translation
Wiem
Nie wiem

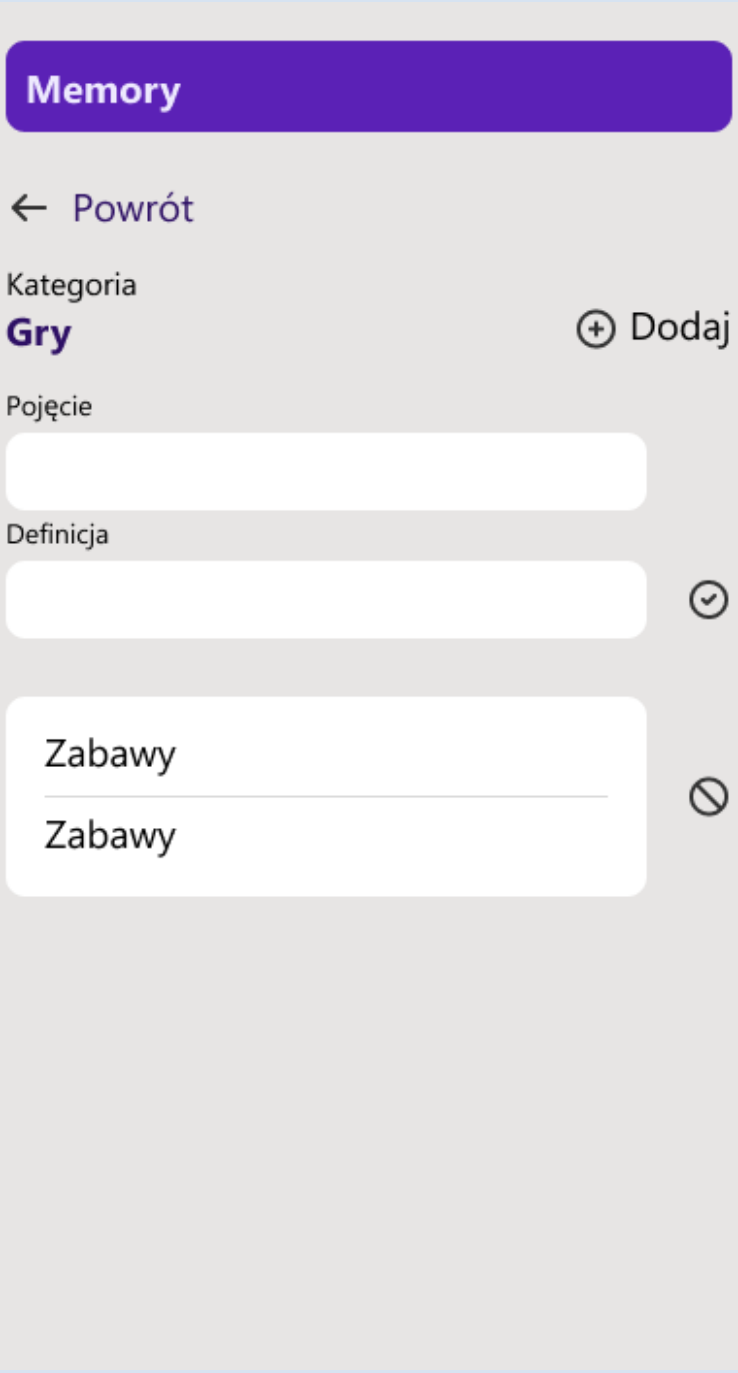

Memory
Powrót
Kategoria
Gry
10 / 10
Podsumowanie
5
Wiem
Nie wiem
1
Powtórz
Zakończ


Memory
Powrót
Kategorie
Dodaj
Gry
Zabawy


Memory
Powrót
Kategorie
Dodaj
Nazwa
Gry
Zabawy


Memory
Powrót
Nie posiadasz żadnej kategorii


Memory
Powrót
Kategoria
Gry
Dodaj
Zabawy
Zabawy


Memory
Powrót
Kategoria
Gry
Dodaj
Pojęcie
Definicja
Zabawy
Zabawy


Memory
Powrót
Kategoria
Gry
Dodaj
Nie posiadasz jeszcze słówek

Memory

Cześć,

czym zajmiesz się dzisiaj?

Nauka

Opanuj nowe słówka

Baza słówek

Zarządzaj kategoriami oraz słówkami

Memory

← Powrót

Wybierz kategorię słówek do ćwiczeń

- Gry
- Zabawy

Memory

← Powrót

Kategoria

Gry

1 / 1

słówko

Obróć

Memory

← Powrót

Kategoria

Gry

1 / 1

słówko

Wiem

Nie wiem

Memory

← Powrót

Kategoria
Gry

1 / 1

Podsumowanie

5 Wiem

Nie wiem 1

Powtórz

Zakończ

Memory

← Powrót

Kategorie

Dodaj

Nazwa

Gry

Zabawy

Memory

← Powrót

Kategoria

Gry

⊕ Dodaj

Zabawy

Zabawy

⊘

Memory

← Powrót

Kategoria

Gry

⊕ Dodaj

Pojęcie

Zabawy

Definicja

Zabawy

Zabawy

Zabawy

Operacje API

Kategoria

Lista kategorii

Pobranie pojedynczej kategorii

Tworzenie nowej kategorii

Usuwanie Kategorii

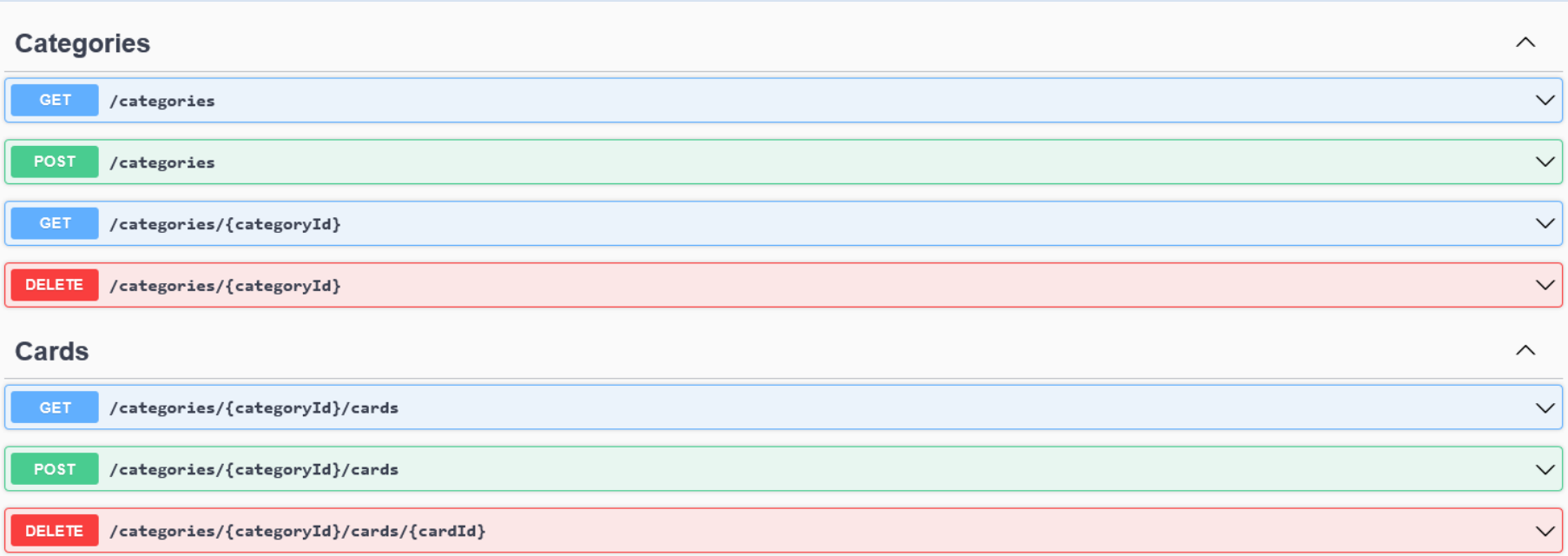
Fiszka

Lista fiszek w kategorii

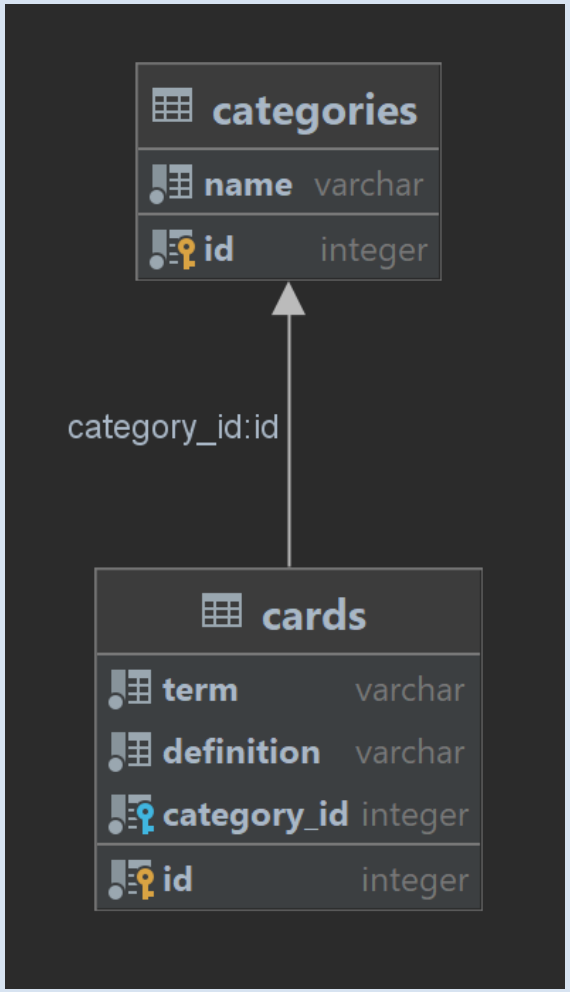
Tworzenie nowej fiszki w kategorii

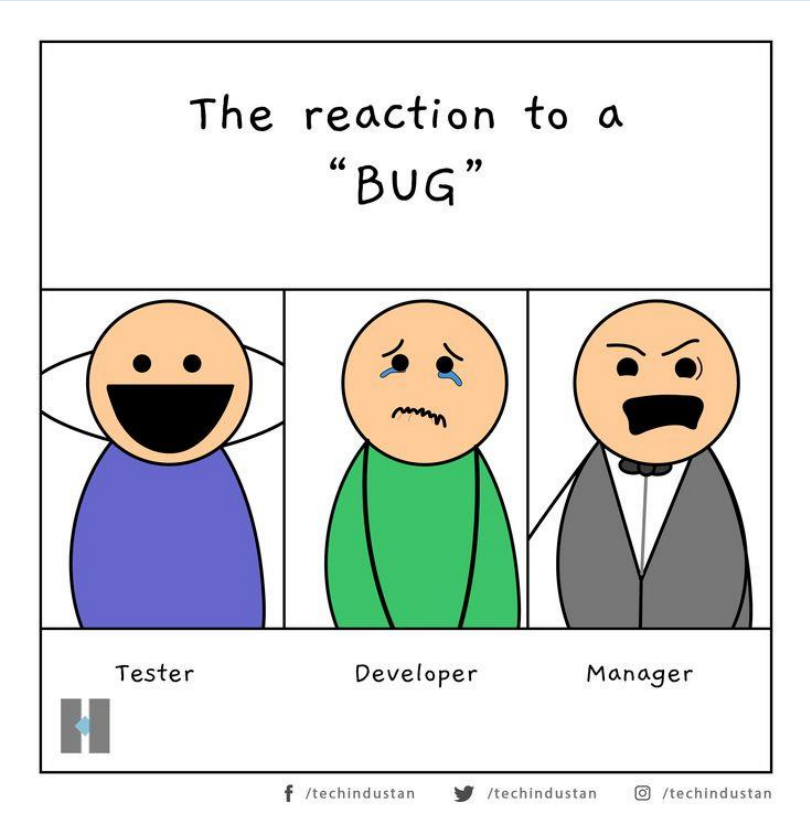
Usuwanie fiszki w kategorii

API endpoints



Schemat bazy danych





04

Development