

Informatyka, Aplikacje internetowe i mobilne, semestr 5

Projektowanie serwisów internetowych

Laboratorium nr 6

Serwis REST-API na przykładzie stripe.com

Budowa front-endu w PHP z wykorzystaniem REST-API Stripe

UWAGA!

Podczas wykonywania laboratorium należy wypełnić odpowiednimi zrzutami ekranu plik lab06_sprawozdanie.docx, które następnie należy zgodnie z terminem wysłać w systemie Sprawer.

1. Kilka słów REST API

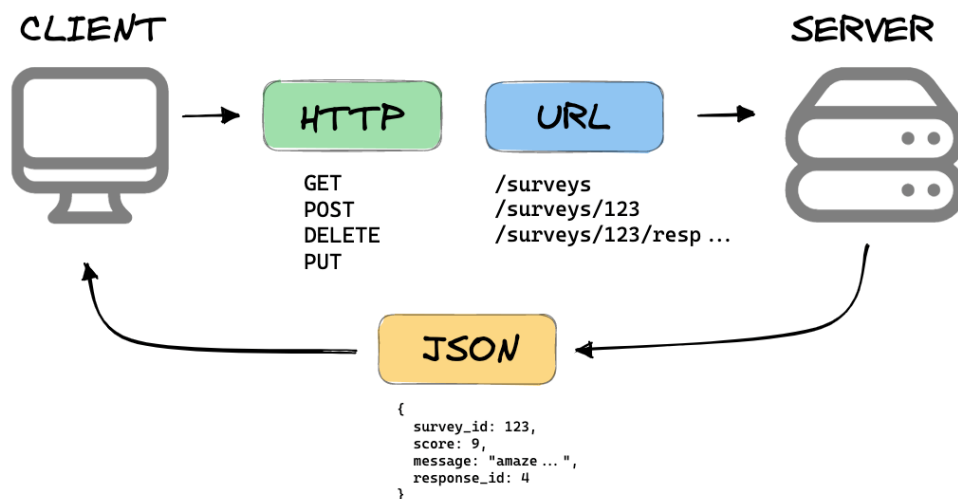
Co to jest REST API (czasami nazywany RESTful API)?

Interfejsy API są najpopularniejszym sposobem interakcji programów i urządzeń w nowoczesnych technologiach obliczeniowych. API to zestaw reguł opisujących, jak jeden program może się łączyć oraz komunikować z innym. Jak sama nazwa wskazuje, API REST przekazuje na każde żądanie stan każdej transakcji, co daje korzyści związane z opracowaniem, wydajnością i zasobami w porównaniu do innych metod. REST rozwija się w ciągu ponad dwóch dekad i jest bardzo powszechnym podejściem do architektur opartych na usługach i architektur rozproszonych.

Zasób jest podstawowym pojęciem dla API REST. Zasób jest obiektem, który ma typ, powiązane dane, relacje z innymi zasobami i zestaw metod, które na nim działają. Jest bardzo podobny do idei obiektów w programowaniu, chociaż wykorzystywanych jest najczęściej tylko kilka standardowych metod, typowych dla HTTP: **GET, POST, PUT i DELETE**. Zasoby mogą istnieć same lub w zbiorach, które same są zasobami.

Styl architektoniczny REST działa w architekturze klient-serwer gdzie klient, który potrzebuje zasobu, identyfikuje się i komunikuje z serwerem, który może go dostarczyć. W ten sposób zarządzany jest praktycznie cały ruch w chmurze, gdyż oferuje maksymalną elastyczność licznym klientom i pozwala na dostęp do licznych serwerów. Zasada ta sprawdza się również w przypadku tzw. architektur „bezserwerowych”, w których miejsce serwera znanego klientowi zajmuje broker usług. Ogólną koncepcję REST API ilustruje rysunek poniżej. Zgodnie z koncepcją REST zasoby są niezależne od reprezentacji odsyłanej klientowi w odpowiedzi, najczęściej wykorzystuje się JSON, HTML lub XML.

WHAT IS A REST API?



Więcej informacji na temat REST API można znaleźć na przykład w artykułach:

<https://devszczepaniak.pl/wprowadzenie-do-rest-api/>

<https://devszczepaniak.pl/projektowanie-rest-api/>

2. Wprowadzenie do serwisu stripe.com

Serwis Stripe.com zapewnia interfejsy API, których programiści mogą używać do integracji przetwarzania płatności w swoich aplikacjach internetowych i mobilnych. Stripe eliminuje niepotrzebną złożoność, dzięki czemu programiści mogą szybko rozpocząć pracę ze Stripe.

Stripe daje możliwość integracji np. ze sklepem internetowym lub innym serwisem. W celu skorzystania z usług serwisu Stripe należy utworzyć konto w tym serwisie i uzyskać klucz/e API. Stripe pozwala wybrać jedną z dostępnych opcji integracji, takich jak wtyczki dla popularnych platform e-commerce lub skorzystanie z **API** dostarczonego przez Stripe.

2.1. Utworzenie konta w serwisie Stripe

W kolejnym kroku należy wykonać czynności opisane w dokumencie:

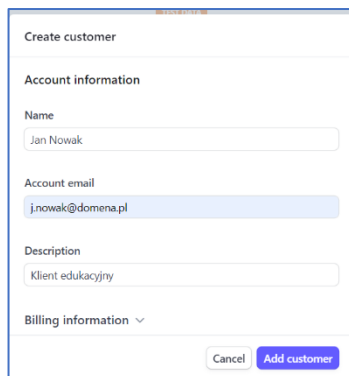
Aktywacja_konta_w_serwisie_stripe_com.pdf

w celu utworzenia konta w serwisie stripe.com, ale **bez określenia płatności**.

Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu.

2.2. Utworzenie nowych klientów w serwisie Stripe

Za pomocą przeglądarki utwórz w serwisie stripe trzech klientów (customers). W tym celu zaloguj się do stripe.com, wybierz z menu Customers, a następnie: Add a test customer



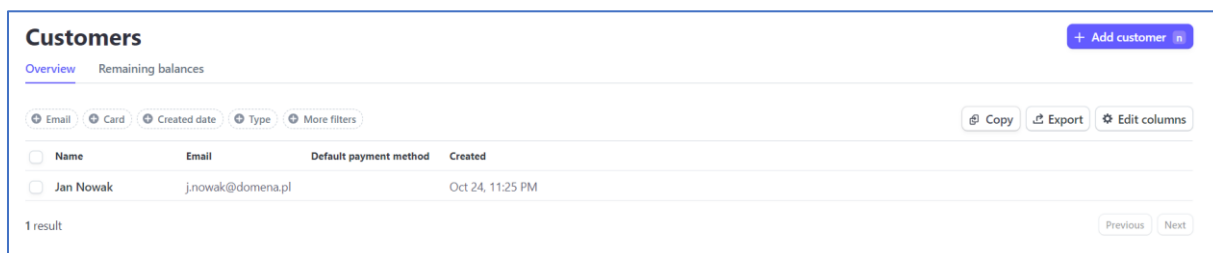
Dopisz dane klientów:

Jan Nowak, j.nowak@domena.pl, Klient edukacyjny (Description)

Adam Kowalski, a.kowalski@test.pl, Klient edukacyjny (Description)

Wanda Pokaz, w.pokaz@moja.pl, Klient edukacyjny (Description)

Nie definiuj dla ww. użytkowników: Billing Account!

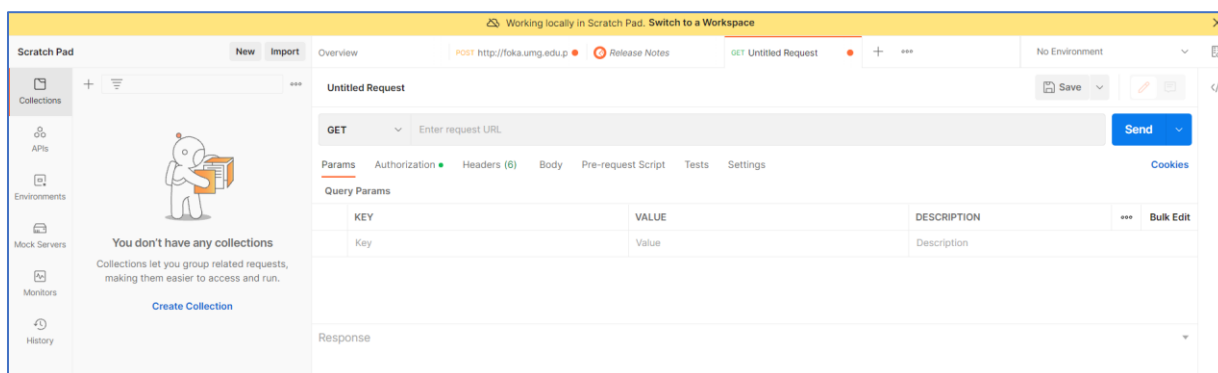


Name	Email	Default payment method	Created
Jan Nowak	j.nowak@domena.pl		Oct 24, 11:25 PM

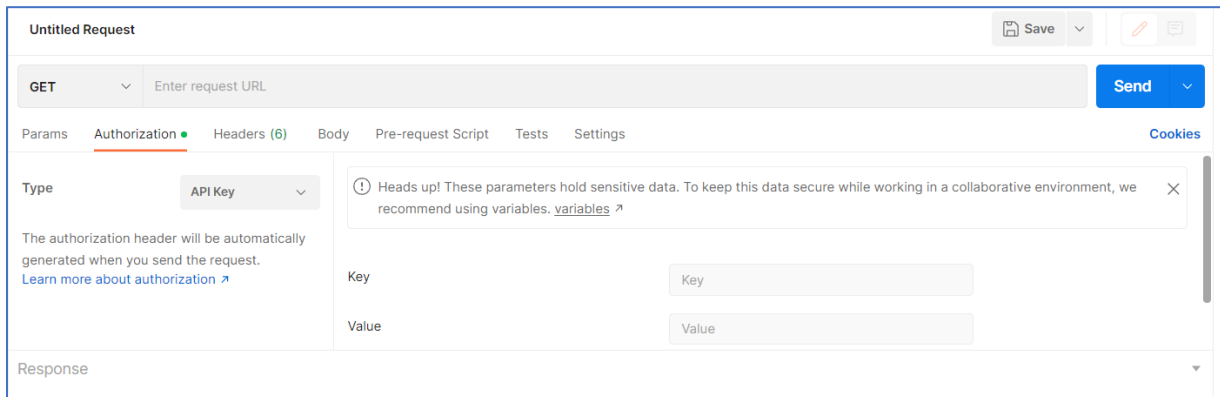
Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu.

3. Testowanie REST API za pomocą narzędzia Postman

Po dopisaniu trzech klientów uruchom narzędzie Postman z katalogu DevEnv/postman/Postman-win64-Setup.exe.

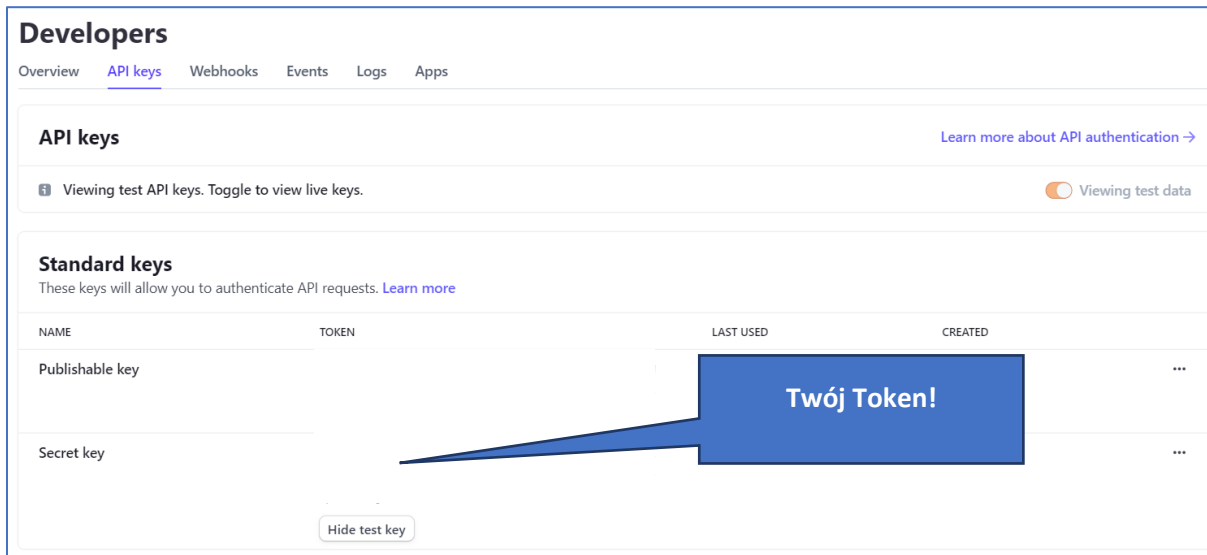


Przełącz się na zakładkę Authorization:



The screenshot shows the Postman interface with the 'Authorization' tab selected. The 'Type' is set to 'API Key'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. variables'. Below this, there are input fields for 'Key' and 'Value'. The 'Response' section is visible at the bottom.

W interfejsie Stripe wróć do zakładki Developers / API keys i odczytaj wartość klucza Secret key.



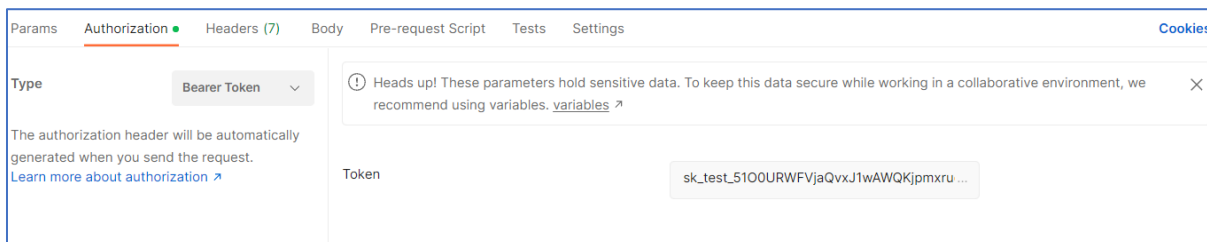
The screenshot shows the Stripe 'Developers' page, specifically the 'API keys' section. It indicates 'Viewing test API keys'. A table lists two types of keys: 'Publishable key' and 'Secret key'. A blue callout box with the text 'Twój Token!' points to the 'Secret key' row. A 'Hide test key' button is at the bottom.

NAME	TOKEN	LAST USED	CREATED
Publishable key			
Secret key			

3.1. Ustawienie autoryzacji w postman

Narzędzie Postman autoryzuje się w serwisie Stripe za pomocą tokena.

Wróć do aplikacji Postman i w ustawieniach Authorization wybierz Typ autoryzacji: **Bearer Token**, a w polu Token wklej skopiowaną wartość Tokena ze Stripe dla wiersza Secret Key:

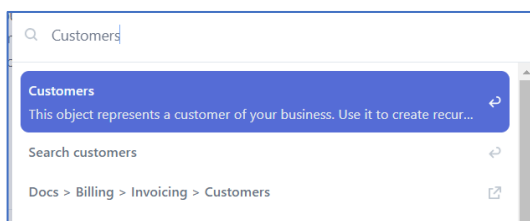


The screenshot shows the Postman 'Authorization' tab with the 'Type' set to 'Bearer Token'. The 'Token' field contains the value 'sk_test_5100URWFVjaQvxJ1wAWQKjpmxru...'. The same warning message about sensitive data is present.

3.2. Pobranie klientów z serwisu Stripe za pomocą Postman.

Aby pobrać za pomocą postmana listę klientów z serwisu Stripe należy użyć odpowiedniego URL z metodą REST API oraz skorzystać z odpowiedzi zawartej w dokumentacji Stripe API. W tym celu w nowej zakładce przeglądarki (tej samej sesji gdzie jesteś zalogowany do Stripe) wprowadź adres URL: <https://stripe.com/docs/api>

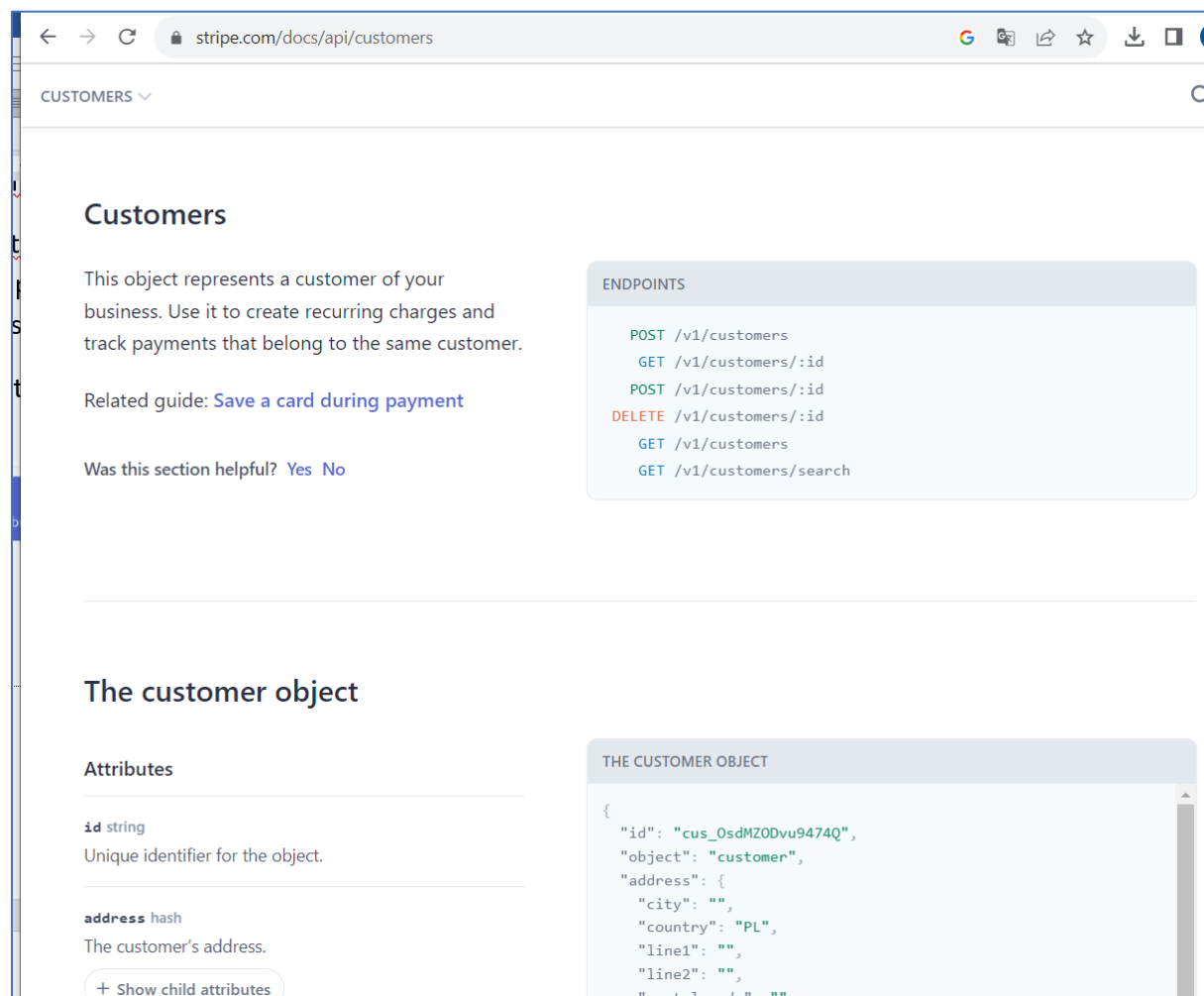
Z tak otrzymanej dokumentacji dowiadujesz się, że podstawowy adres API (Basic API URL):



<https://api.stripe.com>

W obszarze API Docs Stripe wybierz Ctrl-F i znajdź pomoc na temat: Customers

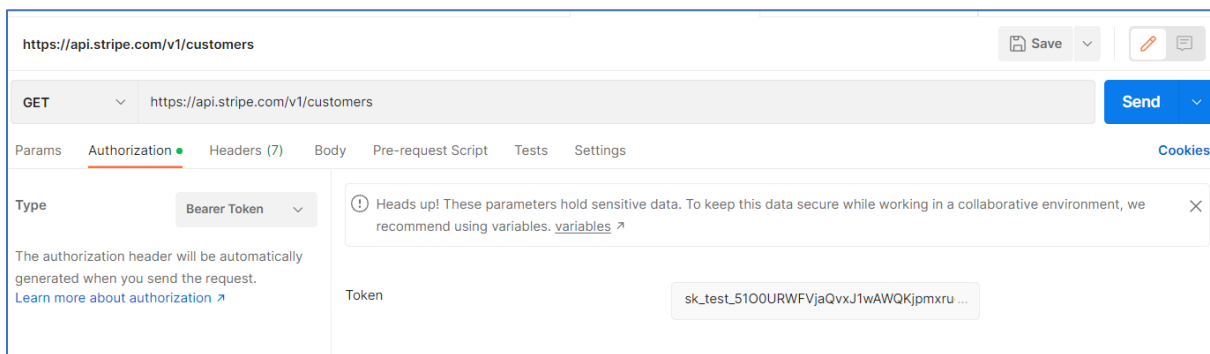
W efekcie otrzymasz obszerną odpowiedź na temat metod GET, POST, DELETE

A screenshot of the Stripe API documentation page for 'Customers'. The page title is 'CUSTOMERS'. The main heading is 'Customers'. Below it, a description states: 'This object represents a customer of your business. Use it to create recurring charges and track payments that belong to the same customer.' There is a link to a 'Related guide: Save a card during payment'. A feedback question asks 'Was this section helpful?' with 'Yes' and 'No' options. On the right side, there is a section titled 'ENDPOINTS' listing several API methods: POST /v1/customers, GET /v1/customers/:id, POST /v1/customers/:id, DELETE /v1/customers/:id, GET /v1/customers, and GET /v1/customers/search. Below this, there is a section titled 'The customer object' with a subsection 'Attributes'. It lists 'id' as a string and 'address' as a hash. At the bottom, there is a button to '+ Show child attributes'. On the right side of this section, there is a box titled 'THE CUSTOMER OBJECT' containing a JSON snippet: { "id": "cus_OsdMZ0Dvu9474Q", "object": "customer", "address": { "city": "", "country": "PL", "line1": "", "line2": "", "postal_code": "" } }

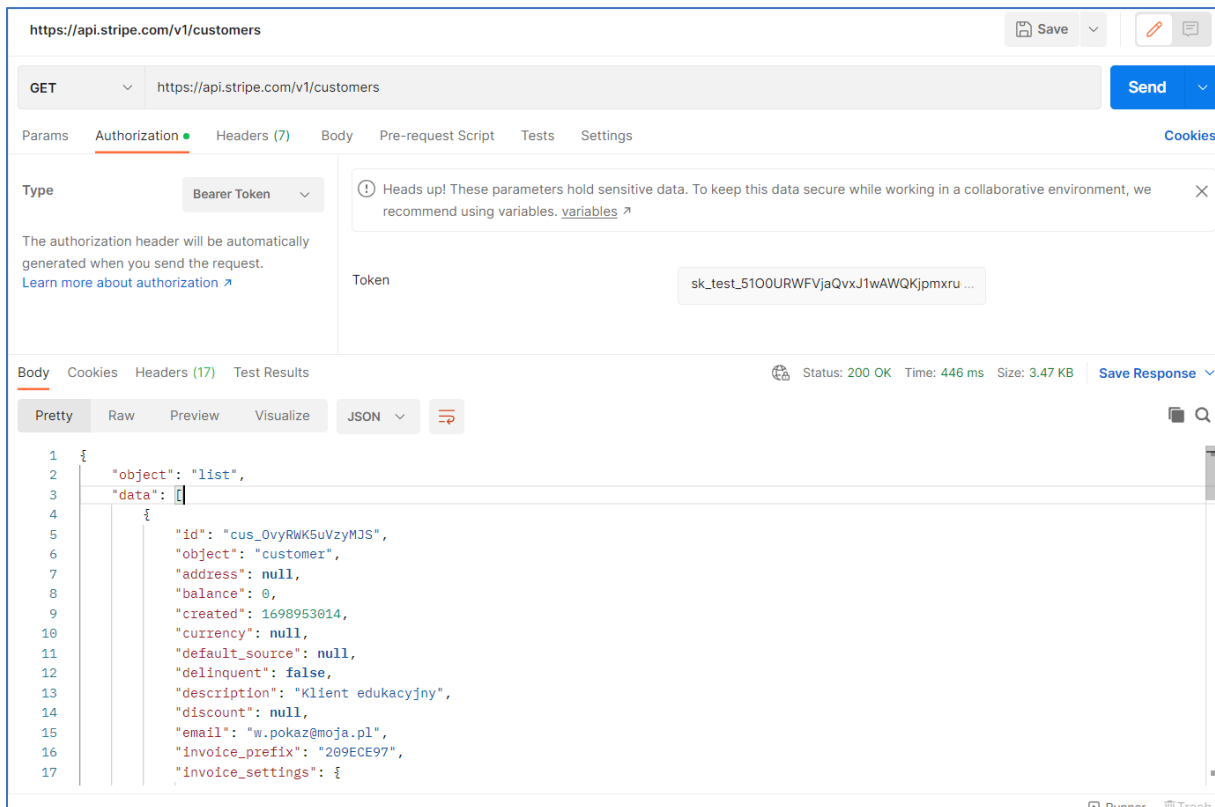
W tak otrzymanej podpowiedzi kliknij na wiersz GET /v1/customers



Po dopisaniu do ww. składni Basic API URL powstaje URL w postaci <https://api.stripe.com/v1/customers>. Skopiuj go i wklej do pola GET w Postmanie (rysunek poniżej). Następnie wybierz Send.



Otrzymamy listę trzech klientów w formacie JSON (sprawdź odpowiedź w Postmanie).

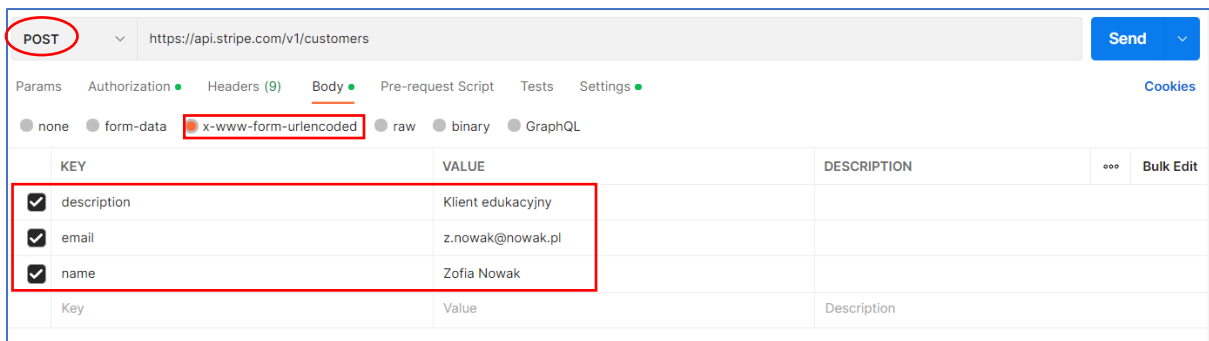


Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu.

3.3. Dopisanie nowego klienta w Stripe za pomocą Postman

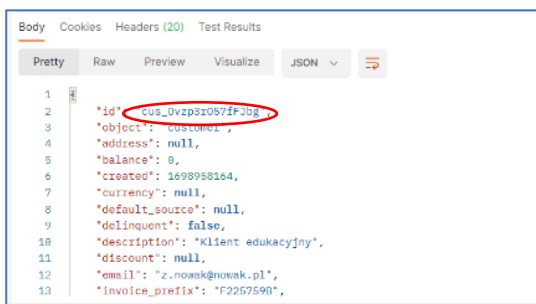
Teraz spróbujemy dopisać do serwisu Stripe nowego klienta edukacyjnego: **Zofia Nowak** o adresie e-mail: z.nowak@nowak.pl. Sprawdź w dokumentacji Stripe API, że należy do tego użyć metody POST z adresem URL API: <https://api.stripe.com/v1/customers>.

Ustaw w Postmanie metodę POST, wybierz Body i przełącz kodowanie na x-www-form-urlencoded. Dodaj odpowiednie klucze: description, email, name z ww. wartościami.



KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> description	Klient edukacyjny	
<input checked="" type="checkbox"/> email	z.nowak@nowak.pl	
<input checked="" type="checkbox"/> name	Zofia Nowak	

Wybierz Send.



```
1 {
2   "id": "cus_Ovzp3r057fFJbg",
3   "object": "customer",
4   "address": null,
5   "balance": 0,
6   "created": 1698958164,
7   "currency": null,
8   "default_source": null,
9   "delinquent": false,
10  "description": "Klient edukacyjny",
11  "discount": null,
12  "email": "z.nowak@nowak.pl",
13  "invoice_prefix": "F2257590",
```

Poniżej w oknie Body / Pretty można zobaczyć utworzone dane nowego klienta (w formacie JSON). Na rysunku obok w czerwonej obwódce wskazano Id nowego klienta (dla każdego żądania inna wartość)

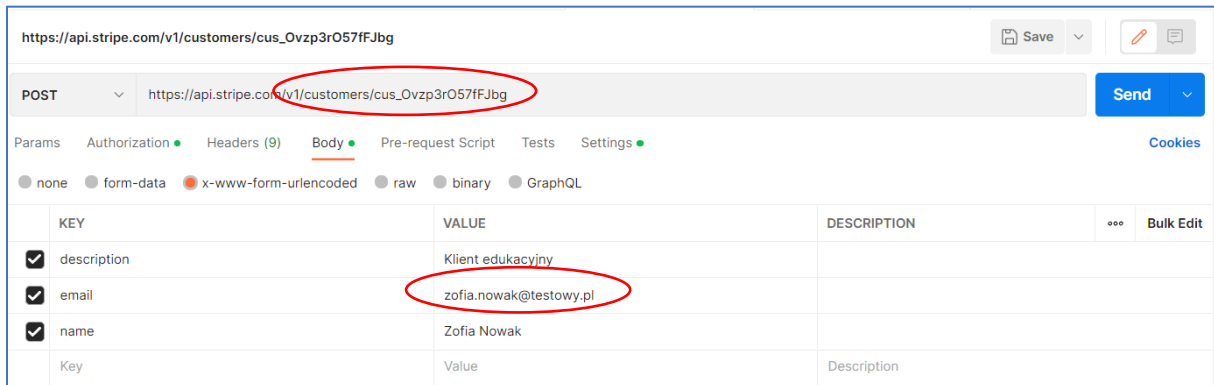
Przełącz się na interfejs usługi Stripe i sprawdź, czy na liście znajduje się nowy klient: **Zofia Nowak**

Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu – potwierdzenie dodania nowego klienta.

3.4. Aktualizacja danych w Postman Zofii Nowak

Odczytaj id klientki Zofia Nowak i przygotuj w Postmanie żądanie zmiany adresu email z z.nowak@nowak.pl na adres email: zofia.nowak@testowy.pl. W tym celu wybierz metodę POST, a w adresie URL wprowadź: <https://api.stripe.com/v1/customers/id>, gdzie w miejsce **id** trzeba wstawić odczytaną wartość id dla klientki: Zofia Nowak. Na przykład:

`https://api.stripe.com/v1/customers/cus_Ovzp3r057fFJbg`



https://api.stripe.com/v1/customers/cus_Ovzp3rO57fFJbg

POST

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> description	Klient edukacyjny			
<input checked="" type="checkbox"/> email	zofia.nowak@testowy.pl			
<input checked="" type="checkbox"/> name	Zofia Nowak			
Key	Value	Description		

Wybierz Send i sprawdź w Stripe, czy nastąpiła aktualizacja adresu email Zofii Nowak.

Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu – aktualizacja danych Zofii Nowak.

3.5. Usunięcie danych w postman Zofii Nowak

Zmień metodę POST na DELETE i wykonaj usunięcie danych Zofii Nowak używając wskazania Customer ID, na przykład: https://api.stripe.com/v1/customers/cus_Ovzp3rO57fFJbg.
Sprawdź listę klientów w Stripe.

Pamiętaj o umieszczeniu zrzutu ekranu w sprawozdaniu – potwierdzenie usunięcia danych Zofii Nowak.

4. Ćwiczenia z REST API

W tym ćwiczeniu skorzystamy z „fakeowej” wersji REST API służącej np. do ćwiczeń.

W nowej zakładce przeglądarki sprawdź zawartość strony:

<https://dummy.restapiexample.com/>:

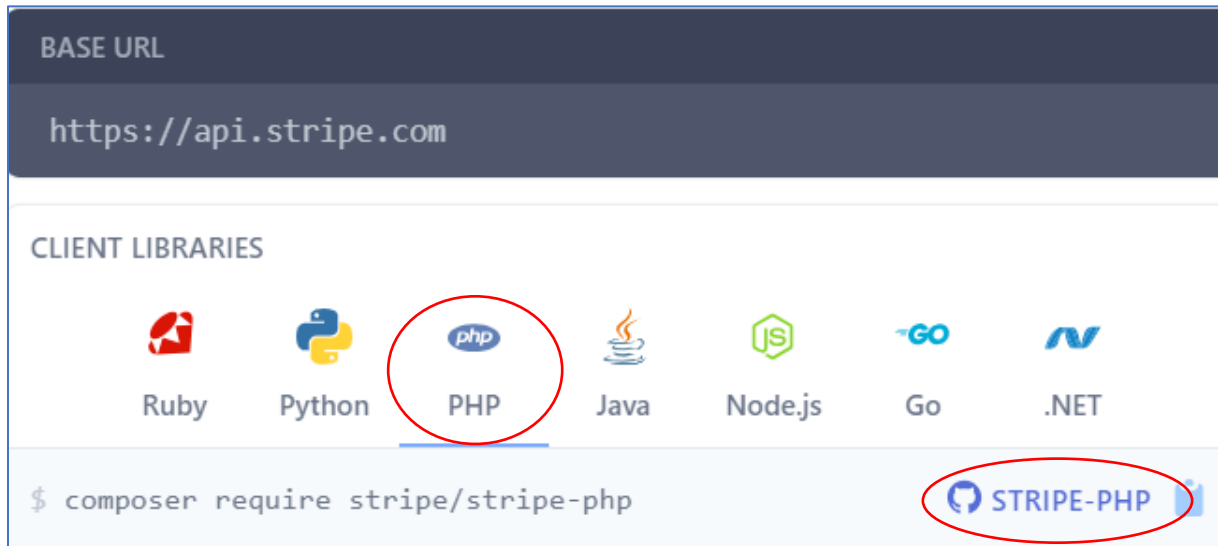
#	Route	Method	Type	Full route	Description	Details
1	/employee	GET	JSON	https://dummy.restapiexample.com/api/v1/employees	Get all employee data	Details
2	/employee/{id}	GET	JSON	https://dummy.restapiexample.com/api/v1/employee/1	Get a single employee data	Details
3	/create	POST	JSON	https://dummy.restapiexample.com/api/v1/create	Create new record in database	Details
4	/update/{id}	PUT	JSON	https://dummy.restapiexample.com/api/v1/update/21	Update an employee record	Details
5	/delete/{id}	DELETE	JSON	https://dummy.restapiexample.com/api/v1/delete/2	Delete an employee record	Details

Przygotuj w Postmanie odpowiednie wywołania GET, POST, PUT i DELETE do tej strony. Najpierw sprawdź każdy adres FULL route w przeglądarce. Obejrzyj zawartość Detail dla poszczególnych wierszy z przykładami.

Dla każdego realizowanego przykładu (numer od 1 do 5) wykonaj zrzut ekranu z ustawień i uzyskanych odpowiedzi w Postmanie. Tak przygotowane zrzuty ekranu umieść w sprawozdaniu (plik Sprawozdanie_Lab6.docx).

5. Dostęp do serwisu Stripe z poziomu PHP

W serwisie Stripe przejdź w zakładce z dokumentacją Stripe API do strony startowej Introduction, adres: <https://stripe.com/docs/api>. Jako język wybierz PHP:

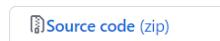


Możesz skorzystać z composera albo pobrać wersję SRC: Stripe PHP za pomocą linku: [STRIPE-PHP](https://github.com/stripe/stripe-php).

Wybierz z menu github latest, np.:

<https://github.com/stripe/stripe-php/releases/tag/v13.0.0>

Pobierz source:



Wybierz link ze źródłem i pobierz plik.

Zależności: Stripe PHP wymaga, aby w PHP były dostępne:

- [curl](#) – dzięki temu można użyć cURL w kliencie PHP,
- [json](#) – format danych JSON,
- [mbstring](#) – Multibyte String.

5.1. Przygotowanie katalogu ze źródłami Stripe-PHP

Utwórz w swoim katalogu publikowania dokumentów html/PHP w XAMPP katalog: PSI_Lab6, a w nim umieść w katalogu strip-php rozpakowane źródło:

stripe-php-X.0.0.zip X: wersja stripe!

Następnie w katalogu PSI_Lab6 utwórz plik o nazwie customers.php.

5.1. Odczytanie listy 3 klientów z serwisu Stripe

Na początku pliku customers.php umieść wiersz:

```
require_once 'stripe-php/init.php';
```

Korzystając z dokumentacji Stripe API dla PHP w celu odczytania listy klientów uzyskujemy np. poniższy kod:

```
<?php
require_once 'stripe-php/init.php';

// połącz ze stripe używając klucza sk_test -> TEST_MODE
// kodzie zmień na swój klucz
$stripe = new \Stripe\StripeClient('sk_test_5100URWfVjaQvxJlwA

// odczytanie mx 3 customers
$max_3_customers = $stripe->customers->all(['limit' => 3]);

//print_r($max_3_customers);
//var_dump($max_3_customers);
// wersja json w przyjaznym formacie
header('Content-Type: application/json');
echo json_encode($max_3_customers, JSON_PRETTY_PRINT);

?>
```

Zauważ, że można różnie oglądać odczytane dane ze Stripe. Wstaw komentarz dla metody `json_encode` oraz `header` i usuń komentarz dla metody `print_r`, a potem zamiennie `var_dump`. Wszystkie próby weryfikuj za pomocą wywołania skryptu `customers.php` w przeglądarce internetowej korzystając z serwera WWW ze wsparciem PHP.

W sprawozdaniu wyślij plik z kodem.

5.2. Dodanie nowego klienta z wykorzystaniem klasy `Stripe\StripeClient`

Utwórz nowy plik o nazwie `add_customer.php` i umieść w nim kod umożliwiający dodanie nowego klienta do

Uwaga: Prawidłowo odczytaj swój Token `sk_test????` i umieść go w powyższym kodzie.

Poniżej przedstawiono przykładowy kod dodania nowego klienta:

```
<?php
require_once 'stripe-php/init.php';
$stripe = new \Stripe\StripeClient('sk_test...');
$customer = $stripe->customers->create([
    'description' => 'Klient edukacyjny',
```

```
'email' => 'r.lasko@domena.pl',  
'name' => 'Robert Laskowski',  
]);  
echo $customer;  
?>
```

W sprawozdaniu wyślij plik z kodem.

5.3. Usunięcie klienta z wykorzystaniem klasy Stripe\StripeClient

Przygotuj kod źródłowy dający możliwość usunięcia klienta **Robert Laskowski**

Sprawdź w swoim Stripe lub Postmanie jaki Customer ID ma Robert Laskowski.

W dokumentacji REST API dla PHP Stripe znajdziesz przykład:

Delete a customer

Permanently deletes a customer. It cannot be undone. Also immediately cancels any active subscriptions on the customer.

Parameters

No parameters.

Returns

Returns an object with a deleted parameter on success. If the customer ID does not exist, this call throws an [error](#).

Unlike other objects, deleted customers can still be retrieved through the API in order to be able to track their history. Deleting customers removes all credit card details and prevents any further operations to be performed (such as adding a new subscription).

```
DELETE /v1/customers/:id
```

```
$stripe = new \Stripe\StripeClient('sk_test_5100URWfVjaQxxJ1wAwQKjpmxruc6');  
$stripe->customers->delete(  
    'cus_OsdH2ODvu9474Q',  
    []  
);
```

RESPONSE

```
{  
  "id": "cus_OsdH2ODvu9474Q",  
  "object": "customer",  
  "deleted": true  
}
```

6. Zadanie - budowa front-endu w PHP z wykorzystaniem REST-API Stripe

Zbuduj z wykorzystaniem języka obiektowego PHP front-end dający możliwość zarządzania klientami serwisu Stripe. W tym celu:

- przygotuj własną klasę o nazwie `KlienciStripe` z odpowiednimi składowymi klasy oraz metodami: `listaklientow`, `dodajklienta`, `usunoklienta`, `aktualizujklienta`. Powyższe metody w klasie `KlienciStripe` powinny korzystać z odpowiednich klas i metod ze `stripe-php` -> tak jak to zostało przedstawione np. w punktach 5.1-5.3.

Tak przygotowaną klasę `KlienciStripe` wykorzystaj do budowy serwisu `html5` z menu z wykorzystaniem zewnętrznego pliku `CSS` oraz sesji `PHP`, tak aby menu pozwalała na:

- wyświetlenie listy klientów z serwisu Stripe,
- dodanie nowego klienta Stripe za pomocą formularza,
- aktualizację i usunięcie wskazanego klienta z listy,

Formularz umożliwiający dodanie i aktualizację klienta powinien posiadać odpowiednią walidację wymaganych pól: name, email, description. Wszystkie kody HTML i CSS muszą przechodzić walidację W3C.

Front-end umieść na swoim koncie na serwerze foka. W sprawozdaniu wyślij pliki .php i .css oraz adres front-endu z serwera foka. W pliku .docx ze sprawozdaniem umieść kody PHP, CSS itd., każdy plik na osobnej stronie.

Sprawozdanie

Przygotowane **pliki** i **adresy** wyślij w sprawozdaniu w systemie **Sprawer**.