



Każde **API**
ma dwa końce
- historia **dostawcy** i **klienta**



Adrian

Ruby Developer

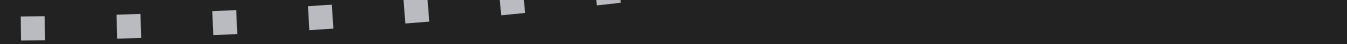
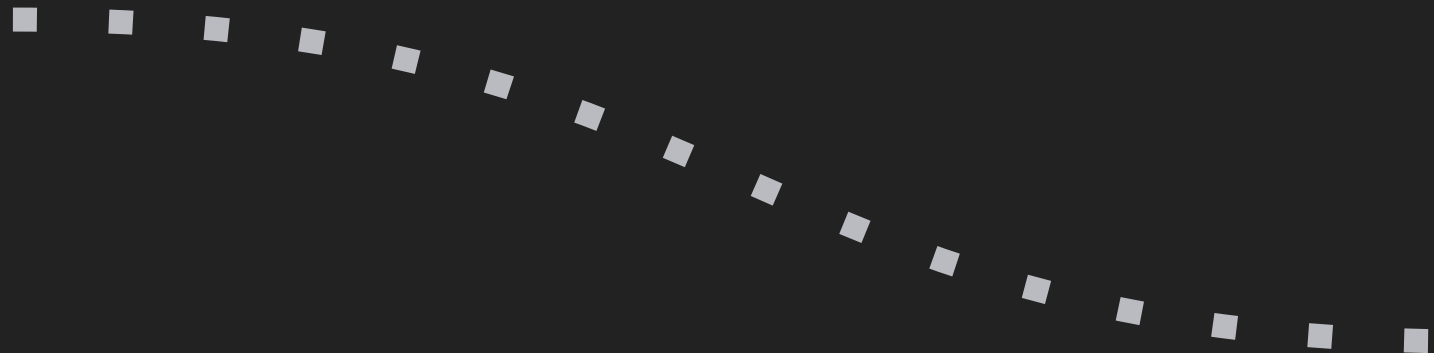
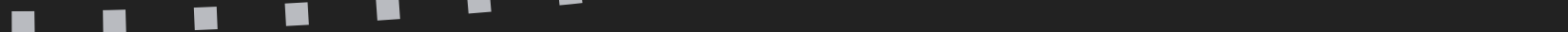
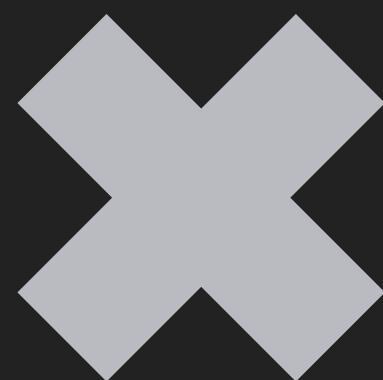
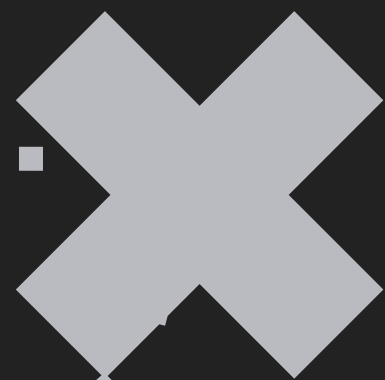
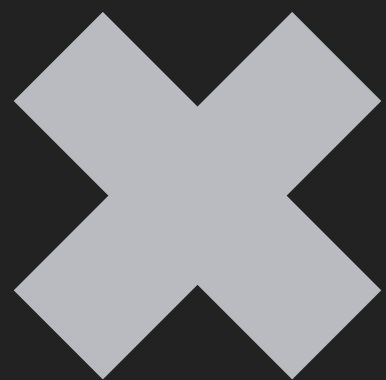
Od lat **rozwija** API inFakt

Darek

android / iOS / microservices

Od lat jest klientem API inFakt



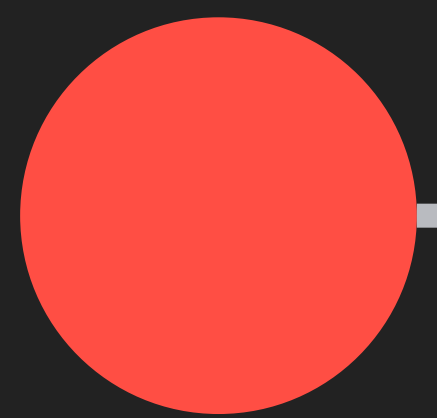




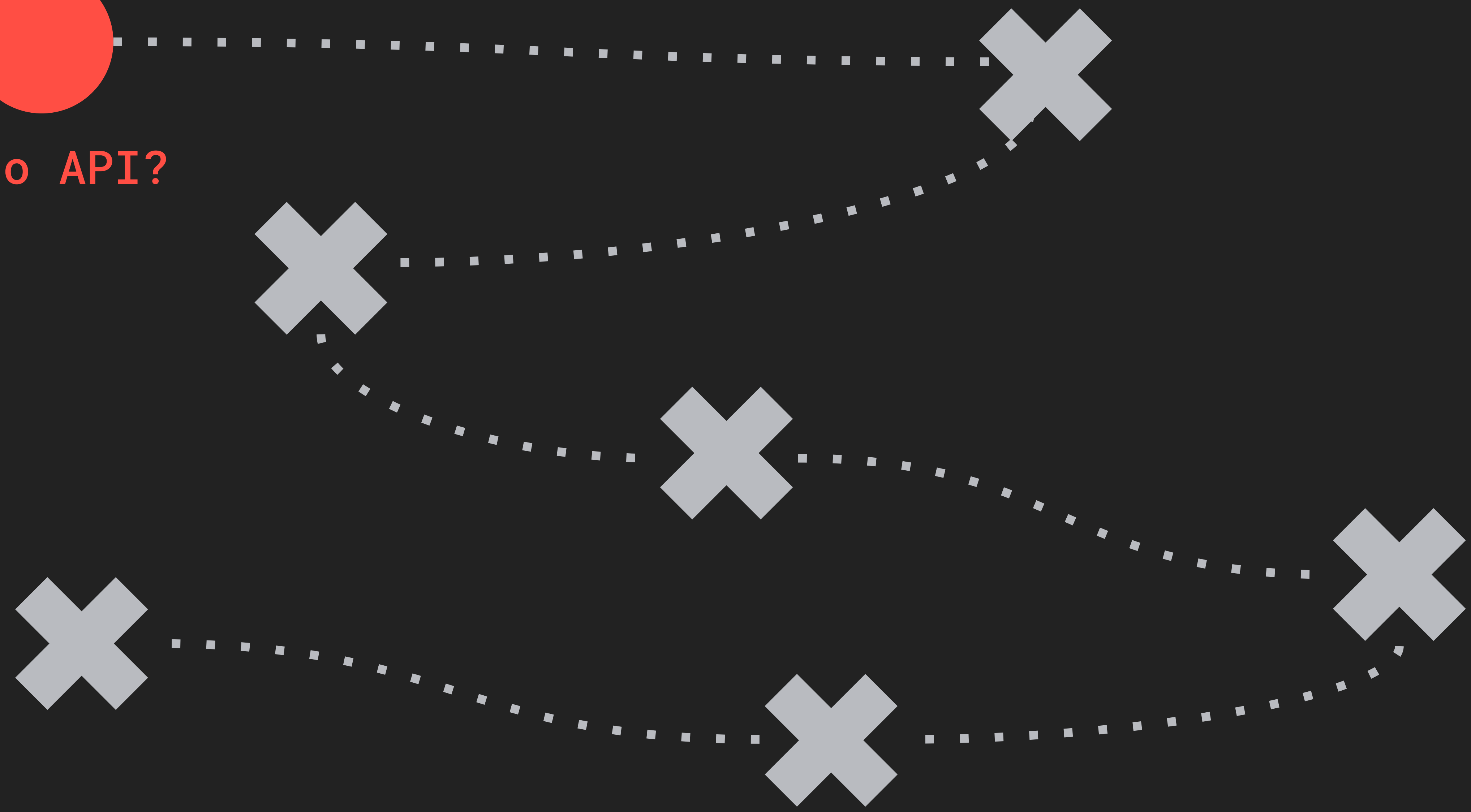
[dobra rada]



[zrealizuj kupon]



Po co API?



inFakt APP

inFakt APP

Android

inFakt APP

Android

iOS

inFakt APP

Android

iOS

AI księgowe

inFakt APP

Android

iOS

AI księgowe

**Sklepy
internetowe**

inFakt APP

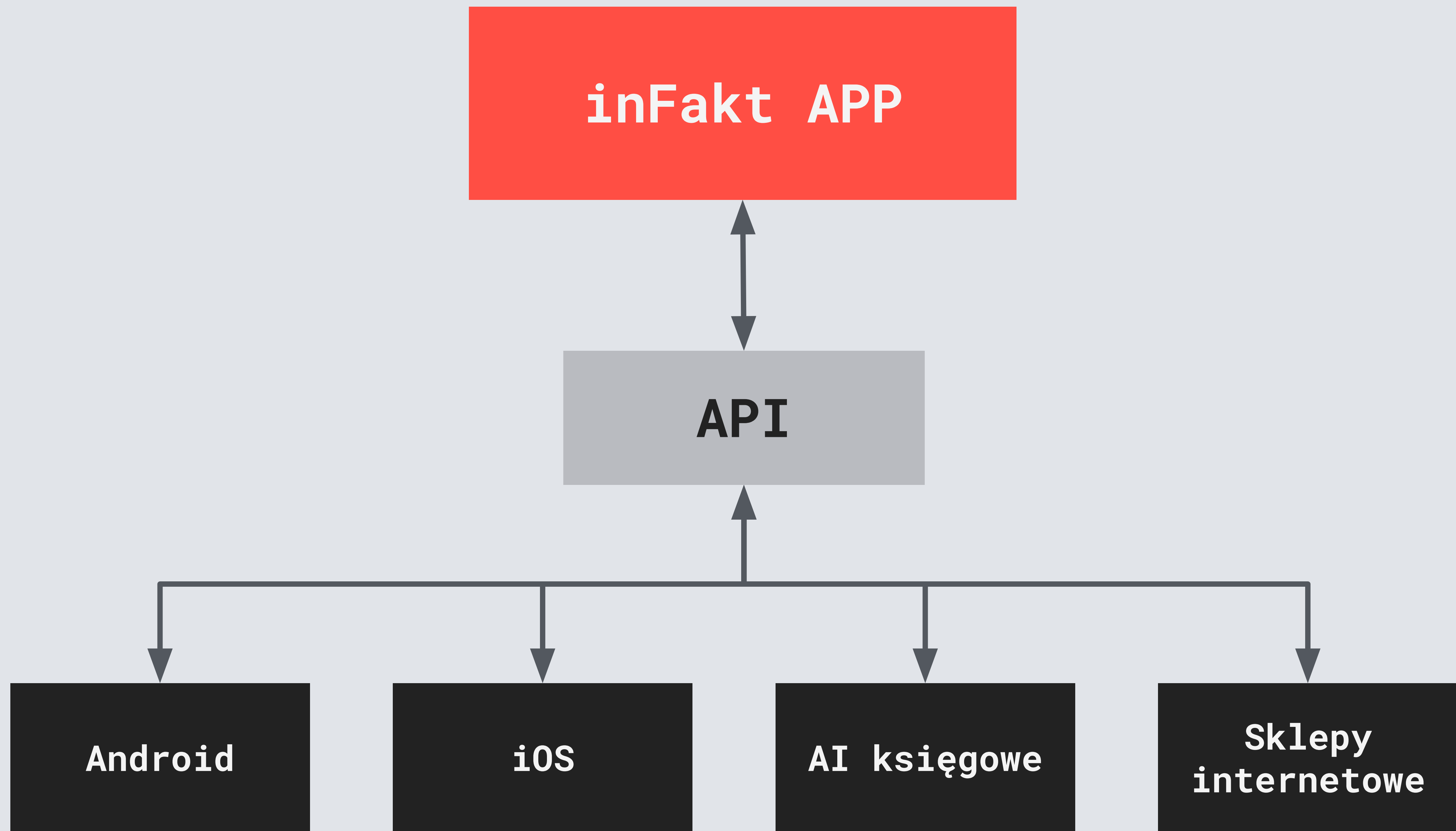
API

Android

iOS

AI księgowe

**Sklepy
internetowe**





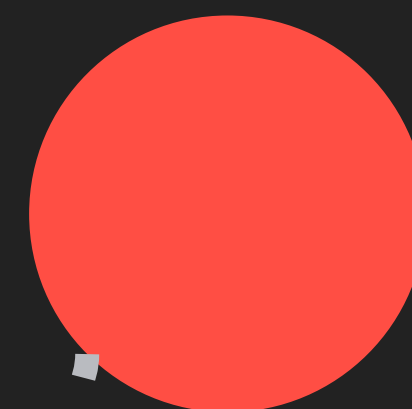
[API pozwala na
integrację z innymi
aplikacjami]



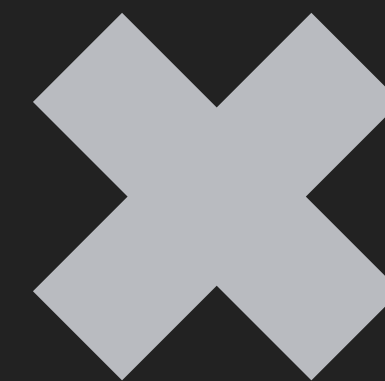
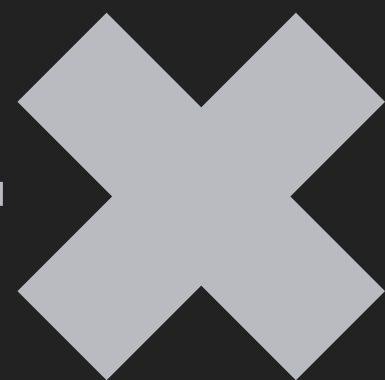
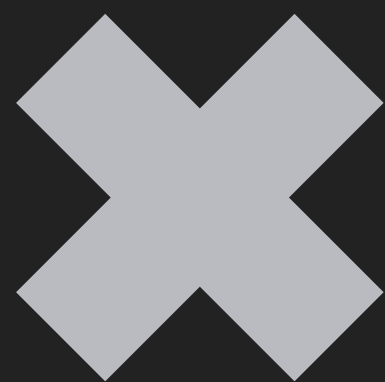
[warto mieć API]



Po co API?



Konwencja



konwencja API

- łatwa implementacja w RoR
- wsparcie na platformie



twórca

klient



konwencja API

- łatwa implementacja w RoR
- wydajne
- wsparcie na platformie
- niski narzut komunikacyjny



twórca

klient



konwencja API

- łatwa implementacja w RoR
- wydajne
- łatwe do testów
- wsparcie na platformie
- niski narzut komunikacyjny
- łatwe w prototypowaniu



twórca

klient



konwencja API

- łatwa implementacja w RoR
- wydajne
- łatwe do testów
- komunikacja w json
- wsparcie na platformie
- niski narzut komunikacyjny
- łatwe w prototypowaniu
- użyte znane standardy



twórca

klient



konwencja API

- łatwa implementacja w RoR
- wydajne
- łatwe do testów
- komunikacja w json
- uniwersalność (różni klienci)
- wsparcie na platformie
- niski narzut komunikacyjny
- łatwe w prototypowaniu
- użyte znane standardy
- niski próg wejścia



twórca

klient



konwencja API

- łatwa implementacja w RoR
- wydajne
- łatwe do testów
- komunikacja w json
- uniwersalność (różni klienci)
- wsparcie dla narzędzi
- wsparcie na platformie
- niski narzut komunikacyjny
- łatwe w prototypowaniu
- użyte znane standardy
- niski próg wejścia
- wsparcie dla narzędzi



twórca

klient



REST

Oparty o HTTP (GET, POST, PUT, PATCH, DELETE)

Każdy zasób ma swój url

Bezstanowy

Wsparcie dla każdej platformy/języka

Łatwe testowanie (np z konsoli: curl)

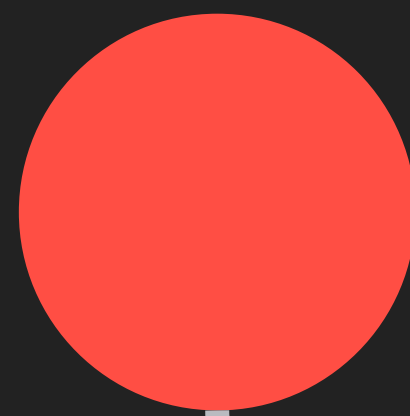


[w API
stawiaj na REST]

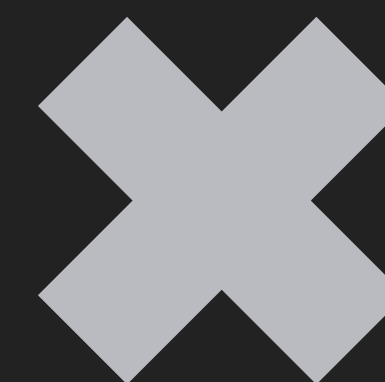
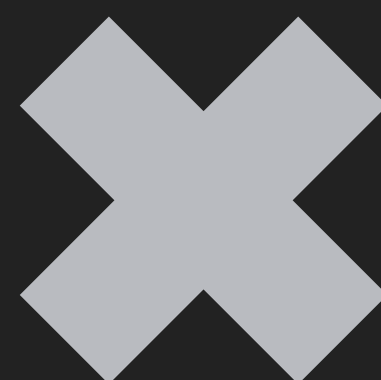


Konwencja

Po co API?



Autoryzacja



autoryzacja

- bezpieczna
- pozwala na logowanie danymi logowania z WWW



twórca

klient



autoryzacja

- bezpieczna
- możliwe dodawanie/usuwanie wielu integracji
- pozwala na logowanie danymi logowania z WWW
- możliwa w implementacji



twórca

klient



autoryzacja

- bezpieczna
- możliwe dodawanie/usuwanie wielu integracji
- każdy request jest autoryzowany
- pozwala na logowanie danymi logowania z WWW
- możliwa w implementacji
- oparta na standardzie



twórca

klient



autoryzacja

- bezpieczna
- możliwe dodawanie/usuwanie wielu integracji
- każdy request jest autoryzowany
- pozwala na logowanie danymi logowania z WWW
- możliwa w implementacji
- oparta na standardzie
- dostępne biblioteki



twórca

klient



API KEY

API KEY

GENEROWANIE

API

Infakt API umożliwia dostęp do zasobów serwisu inFakt.pl za pomocą interfejsu JSON.

Szczegółowa dokumentacja dla API znajduje się pod adresem <https://www.infakt.pl/developers/>.

Wygeneruj nowy klucz

Opis klucza

test

✓ Wygeneruj

Lista aplikacji

Aplikacja iOS (systemowy)

[Usuń](#)

Ostatnio używany

Grudzień 28, 2017

Wygenerowany klucz API

*** [Wygeneruj ponownie](#)

API KEY

PRZEKAZYWANIE

Do każdego zapytania

```
GET /invoices HTTP/1.1
```

```
HEADERS:
```

```
X-API-Key: abcdef12345...
```

PRYWATNE ZASOBY

KONTROLA DOSTĘPU

TOTP

(Time-based One-time Password)

TOTP

GENEROWANIE

generowanie TOTP

- wyliczany po stronie klienta API na podstawie **współdzielonego klucza**
- wyliczany i weryfikowany po stronie dostawcy API
- ważny 30 sekund

TOTP

PRZEKAZYWANIE

Do każdego zapytania

```
GET /invoices HTTP/1.1
```

```
HEADERS:
```

```
X-API-Totp: 163523
```



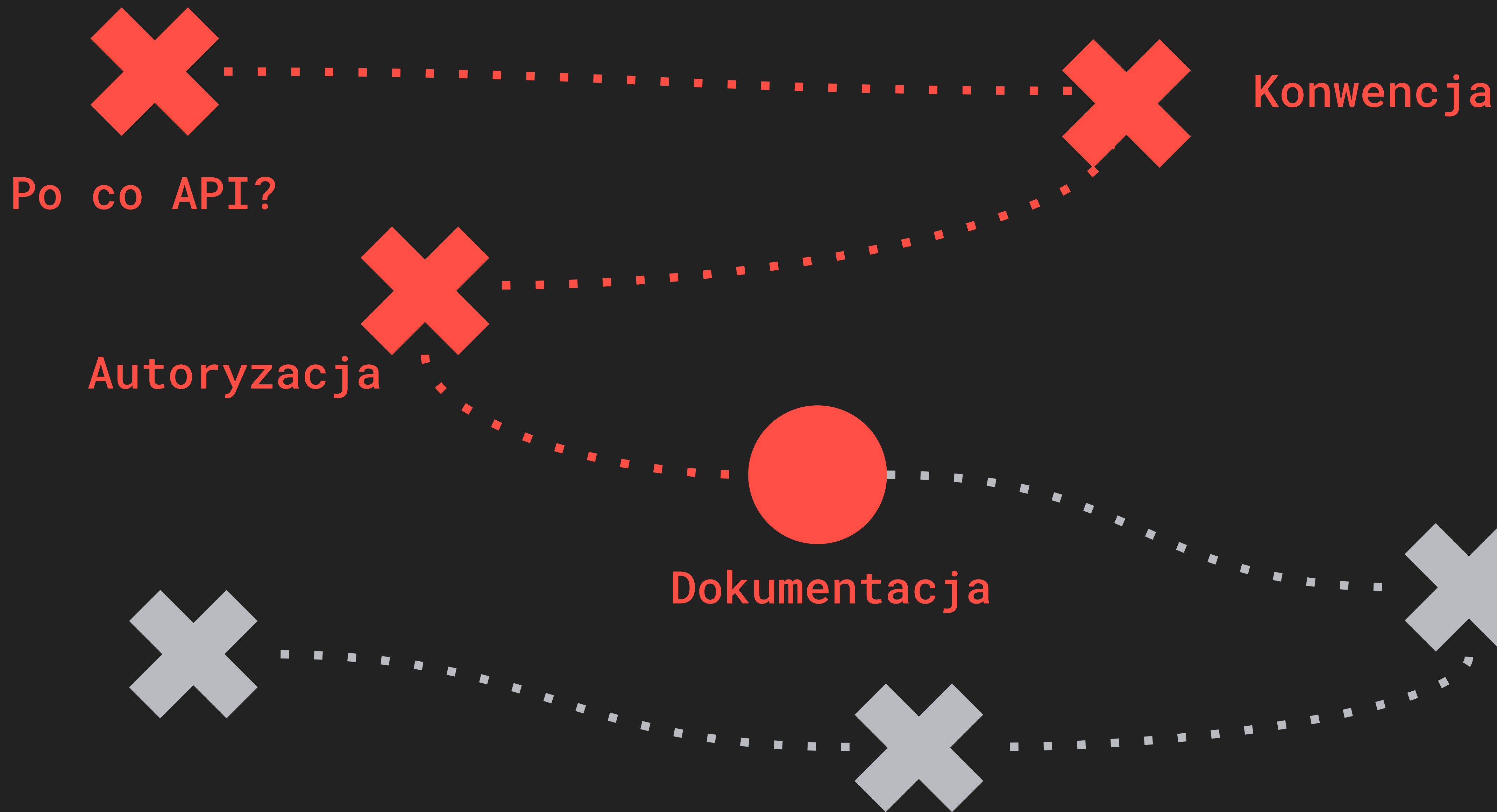
[autoryzacja w API
działa inaczej niż na
www]



[autoryzacja przez klucz
API jest wygodna i
uniwersalna]



[weryfikacja TOTP
pozwala na dostęp do
prywatnych zasobów]



dokumentacja

- po co? :)

- niezbędna!



twórca

klient



dokumentacja

- po co? :)
- kod sam się dokumentuje
- niezbędna!
- zrozumiała



twórca

klient



dokumentacja

- po co? :)
- kod sam się dokumentuje
- minimalny narzut na pracę
- niezbędna!
- zrozumiała
- aktualna



twórca

klient



dokumentacja

- po co? :)
- kod sam się dokumentuje
- minimalny narzut na pracę
- wygodna w aktualizacji
- niezbędna!
- zrozumiała
- aktualna
- umożliwiająca testy



twórca

klient



AUTOMATYCZNE GENEROWANIE I PLAYGROUND

apiary.io

apiary.io

generowana dokumentacja

playground

wystawienie mocka API na podstawie odpowiedzi

testowanie API

apiary.io

1. piszemy specyfikację w markdown

apiary.io

1. piszemy specyfikację w **markdown**
2. dodajemy na **github**



<> Edit file

Preview changes

Spaces



4



Soft wrap



12

13 **# Group Messages**

14 Group of all messages-related resources.

15

16 **## My Message [./message]**

17

18 **### Retrieve a Message [GET]**

19 This action has ****two**** responses defined: One returning plain text and the
20 other a JSON representation of our resource. Both have the same HTTP status
21 code. Also both responses bear additional information in the form of a custom
22 HTTP header. Note that both responses have set the `Content-Type` HTTP header
23 just by specifying `(text/plain)` or `(application/json)` in their respective
24 signatures.

25

26 + Response 200 (text/plain)

27

28 + Headers

29

30 X-My-Message-Header: 42

31

32 + Body

33

34 Hello World!

35

36 + Response 200 (application/json)

37

38 + Headers

39

40 X-My-Message-Header: 42

41

42 + Body

43

44 { "message": "Hello World!" }

45

apiary.io

1. piszemy specyfikację w **markdown**
2. dodajemy na **github**
3. podpinamy **apiary**

INTRODUCTION

API Blueprint

REFERENCE

Messages

My Message

Messages

Group of all messages-related resources.

My Message

Retrieve a Message

This action has **two** responses defined: One returning plain text and the other a JSON representation of our resource. Both have the same HTTP status code. Also both responses bear additional information in the form of a custom HTTP header. Note that both responses have set the `Content-Type` HTTP header just by specifying `(text/plain)` or `(application/json)` in their respective signatures.

Update a Message



Switch to Console

Messages / My Message / Retrieve a Message

GET `https://private-2274c1-agrofoto.apiary-mock.com/message`

Request

Mock Server

Raw

Try

Response

200

HEADERS

`Content-Type: text/plain`
`X-My-Message-Header: 42`

BODY

`01 Hello World!`

Response

200

HEADERS

INTRODUCTION

API Blueprint

REFERENCE

Messages

My Message

Messages

Group of all messages-related resources.

My Message

Retrieve a Message

This action has **two** responses defined: One returning plain text and the other a JSON representation of our resource. Both have the same HTTP status code. Also both responses bear additional information in the form of a custom HTTP header. Note that both responses have set the `Content-Type` HTTP header just by specifying `(text/plain)` or `(application/json)` in their respective signatures.

Update a Message



Switch to Console

Messages / My Message / Retrieve a Message

GET `https://private-2274c1-agrofoto.apiary-mock.com/message`

Request

Mock Server

Raw

Try

Response

200

HEADERS

`Content-Type: text/plain`
`X-My-Message-Header: 42`

BODY

`01 Hello World!`

Response

200

HEADERS

INTRODUCTION

API Blueprint

REFERENCE

Messages

My Message

Messages

Group of all messages-related resources.

My Message

Retrieve a Message

This action has **two** responses defined: One returning plain text and the other a JSON representation of our resource. Both have the same HTTP status code. Also both responses bear additional information in the form of a custom HTTP header. Note that both responses have set the `Content-Type` HTTP header just by specifying `(text/plain)` or `(application/json)` in their respective signatures.

Update a Message



Switch to Console

Messages / My Message / Retrieve a Message

GET `https://private-2274c1-agrofoto.apiary-mock.com/message`

Request

Mock Server

Raw

Try

Response

200

HEADERS

`Content-Type: text/plain`
`X-My-Message-Header: 42`

BODY

01 Hello World!

Response

200

HEADERS



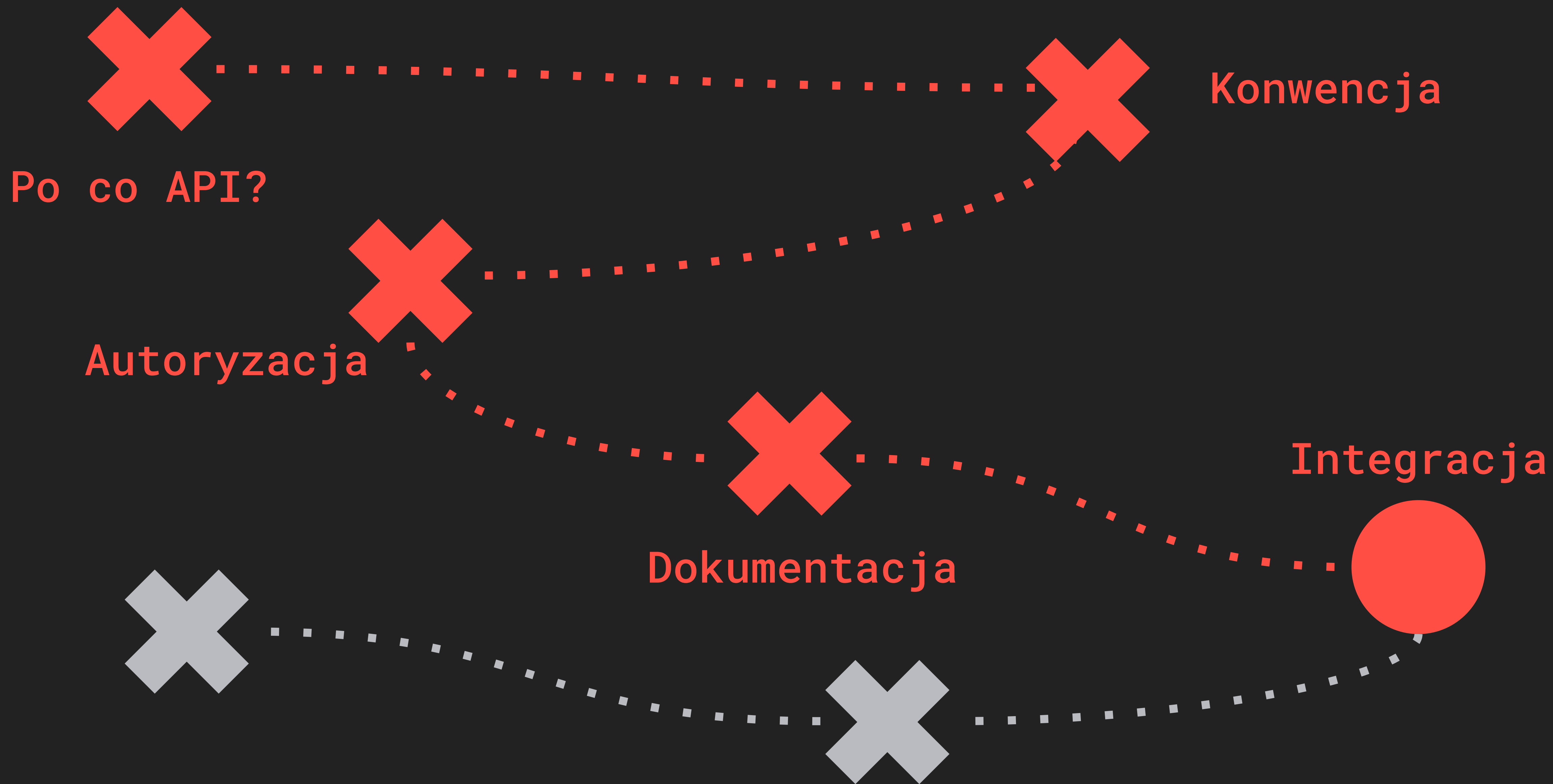
[klient poznaje API
dzięki dokumentacji]



[miej aktualną,
automatycznie generowaną
dokumentację]



[udostępnij playground]



MOBILE

iOS

iOS

SWIFT

iOS

SWIFT

Alamofire + ObjectMapper

ANDROID

ANDROID

JAVA/KOTLIN

ANDROID

JAVA/KOTLIN

RETROFIT

WYWOŁANIE METODY API

RETROFIT

```
public interface InFaktApi {  
    @GET("/cost_files")  
    PagedResponse<CostFile> getCostFiles();  
  
    @POST("/cost_files")  
    CostFile createCostFile(@Body CostFile costFile);  
  
    @PUT("/cost_files/{id}")  
    CostFile updateCostFile(@Body CostFile costFile, @Path("id") int id);  
  
    @DELETE("/cost_files/{id}")  
    Void deleteCostFile(@Path("id") int id);  
}
```

```
public interface InFaktApi {  
    @GET("/cost_files")  
    PagedResponse<CostFile> getCostFiles();  
  
    @POST("/cost_files")  
    CostFile createCostFile(@Body CostFile costFile);  
  
    @PUT("/cost_files/{id}")  
    CostFile updateCostFile(@Body CostFile costFile, @Path("id") int id);  
  
    @DELETE("/cost_files/{id}")  
    Void deleteCostFile(@Path("id") int id);  
}
```



```
public interface InFaktApi {  
    @GET("/cost_files")  
    PagedResponse<CostFile> getCostFiles();  
  
    @POST("/cost_files")  
    CostFile createCostFile(@Body CostFile costFile);  
  
    @PUT("/cost_files/{id}")  
    CostFile updateCostFile(@Body CostFile costFile, @Path("id") int id);  
  
    @DELETE("/cost_files/{id}")  
    Void deleteCostFile(@Path("id") int id);  
}
```

```
public interface InFaktApi {  
    @GET("/cost_files")  
    PagedResponse<CostFile> getCostFiles();  
  
    @POST("/cost_files")  
    CostFile createCostFile(@Body CostFile costFile);  
  
    @PUT("/cost_files/{id}")  
    CostFile updateCostFile(@Body CostFile costFile, @Path("id") int id);  
  
    @DELETE("/cost_files/{id}")  
    Void deleteCostFile(@Path("id") int id);  
}
```

OBIEKTY API

STATYCZNE TYPOWANIE

```
class CostFile : Entity() {  
    lateinit var filename: String  
  
    lateinit var status: Status  
  
    enum class Status constructor(val nameResId: Int, val colorResId: Int) {  
        not_registered(R.string.cost_file_status_pending, R.color.infakt_violet),  
        registered(R.string.cost_file_status_registered, R.color.infakt_green_2),  
        discarded(R.string.cost_file_status_discarded, R.color.infakt_error),  
        new(R.string.cost_file_status_new, R.color.infakt_gray)  
    }  
}
```

```
class CostFile: Entity() {  
    lateinit var filename: String  
  
    lateinit var status: Status  
  
    enum class Status constructor(val nameResId: Int, val colorResId: Int) {  
        not_registered(R.string.cost_file_status_pending, R.color.infakt_violet),  
        registered(R.string.cost_file_status_registered, R.color.infakt_green_2),  
        discarded(R.string.cost_file_status_discarded, R.color.infakt_error),  
        new(R.string.cost_file_status_new, R.color.infakt_gray)  
    }  
}
```

```
class CostFile: Entity() {  
    lateinit var filename: String  
  
    lateinit var status: Status  
  
    enum class Status constructor(val nameResId: Int, val colorResId: Int) {  
        not_registered(R.string.cost_file_status_pending, R.color.infakt_violet),  
        registered(R.string.cost_file_status_registered, R.color.infakt_green_2),  
        discarded(R.string.cost_file_status_discarded, R.color.infakt_error),  
        new(R.string.cost_file_status_new, R.color.infakt_gray)  
    }  
}
```

```
class CostFile: Entity() {  
    lateinit var filename: String  
    lateinit var status: Status
```

```
enum class Status constructor(val nameResId: Int, val colorResId: Int) {  
    not_registered(R.string.cost_file_status_pending, R.color.infakt_violet),  
    registered(R.string.cost_file_status_registered, R.color.infakt_green_2),  
    discarded(R.string.cost_file_status_discarded, R.color.infakt_error),  
    new(R.string.cost_file_status_new, R.color.infakt_gray)  
}
```

```
}
```



[logika klienta
mobilnego polega na API]

MOBILE

MOBILE

RELEASE

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk /.ipa

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk /.ipa

UPLOAD
BINARKI DO
SKLEPU

5min

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

1h

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

1h

REVIEW

iOS App Store



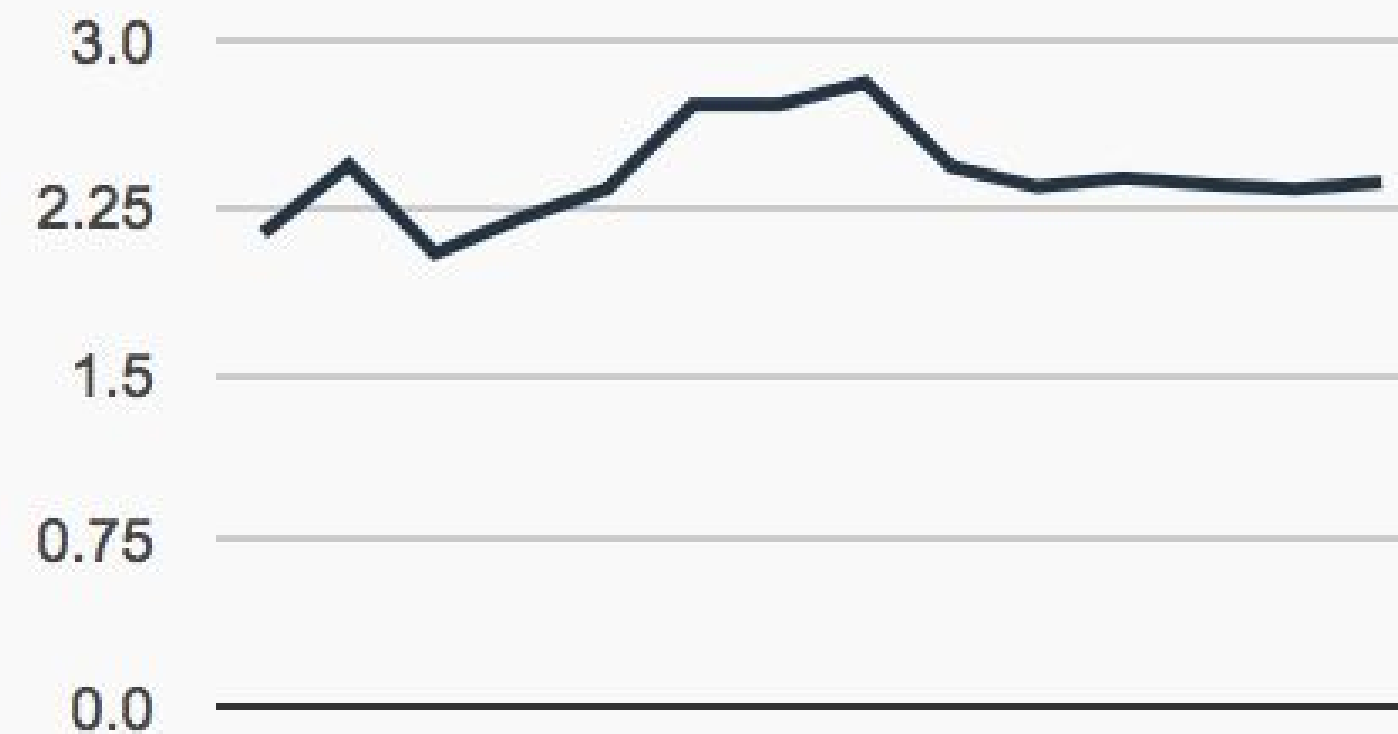
3 days

Based on 58 reviews in the last 14 days.

More Data

- ★ [Distribution Chart](#)
- ★ [Raw Data](#)
- ★ [Annual Trend Graph](#)

Rolling 14 Day Trend



ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

1h

REVIEW

3~5dni

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk / .ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

1h

REVIEW

3~5dni

AKTUALIZACJA
W URZĄDZENIU

ROBIMY RELEASE

BUDUJEMY
BINARKE
.apk /.ipa

5min

UPLOAD
BINARKI DO
SKLEPU

2min

PRZETWARZANIE

1h

REVIEW

3~5dni

AKTUALIZACJA
W URZĄDZENIU

?? dni



[dystribucja aplikacji
mobilnych trwa długo]



czas propagacji zmian

??

??



twórca



klient

czas propagacji zmian

30

min

??



twórca



klient

zmiany

- w dowolnym momencie
- wszyscy mają nową wersję
- zmiany na produkcji, można zapomnieć o starej wersji
- dokumentację się uzupełni :)
- po zapowiedzi :)
- czasem dwie wersje naraz
- wsparcie starej wersji
- kontrola nad wdrożeniem
- po aktualizacji dokumentacji



twórca

klient



ANALIZA ZMIAN

ZMIANA ZASOBU

```
class CostFile: Entity() {  
    lateinit var filename: String  
    lateinit var status: Status
```

```
enum class Status constructor(val nameResId: Int, val colorResId: Int) {  
    not_registered(R.string.cost_file_status_pending, R.color.infakt_violet),  
    registered(R.string.cost_file_status_registered, R.color.infakt_green_2),  
    discarded(R.string.cost_file_status_discarded, R.color.infakt_error),  
    new(R.string.cost_file_status_new, R.color.infakt_gray)  
}
```

```
}
```

zmiana zasobu

projektujemy zmianę

- aktualizacja dokumentacji
- implementacja
- utrzymanie starej wersji
- deploy
- analiza dokumentacji
- implementacja nowej wersji
- testy obu wersji
- release



twórca

klient



WERSJONOWANIE

FEATURE HEADER

do każdego zapytania

```
GET /invoices HTTP/1.1
```

```
HEADERS:
```

```
X-API-Features: 'new_status'
```

odpowiedź

X-API-Features: 'new_status'

```
"id": 51,  
"status": "not_registered",  
"assigned_to_month": "2017-10-01",  
"category": "company_cost",  
"cost_id": "",  
"filename": "test.pdf",  
"created_on": "2017-10-25",
```

```
"id": 51,  
"status": "waiting",  
"assigned_to_month": "2017-10-01",  
"category": "company_cost",  
"cost_id": "",  
"filename": "test.pdf",  
"created_on": "2017-10-25",
```


STARY I NOWY KLIENT

WERSJONOWANIE API

WERSJONOWANIE

ZUPEŁNIE NOWA WERSJA

każde zapytanie

```
GET v2/invoices HTTP/1.1
```



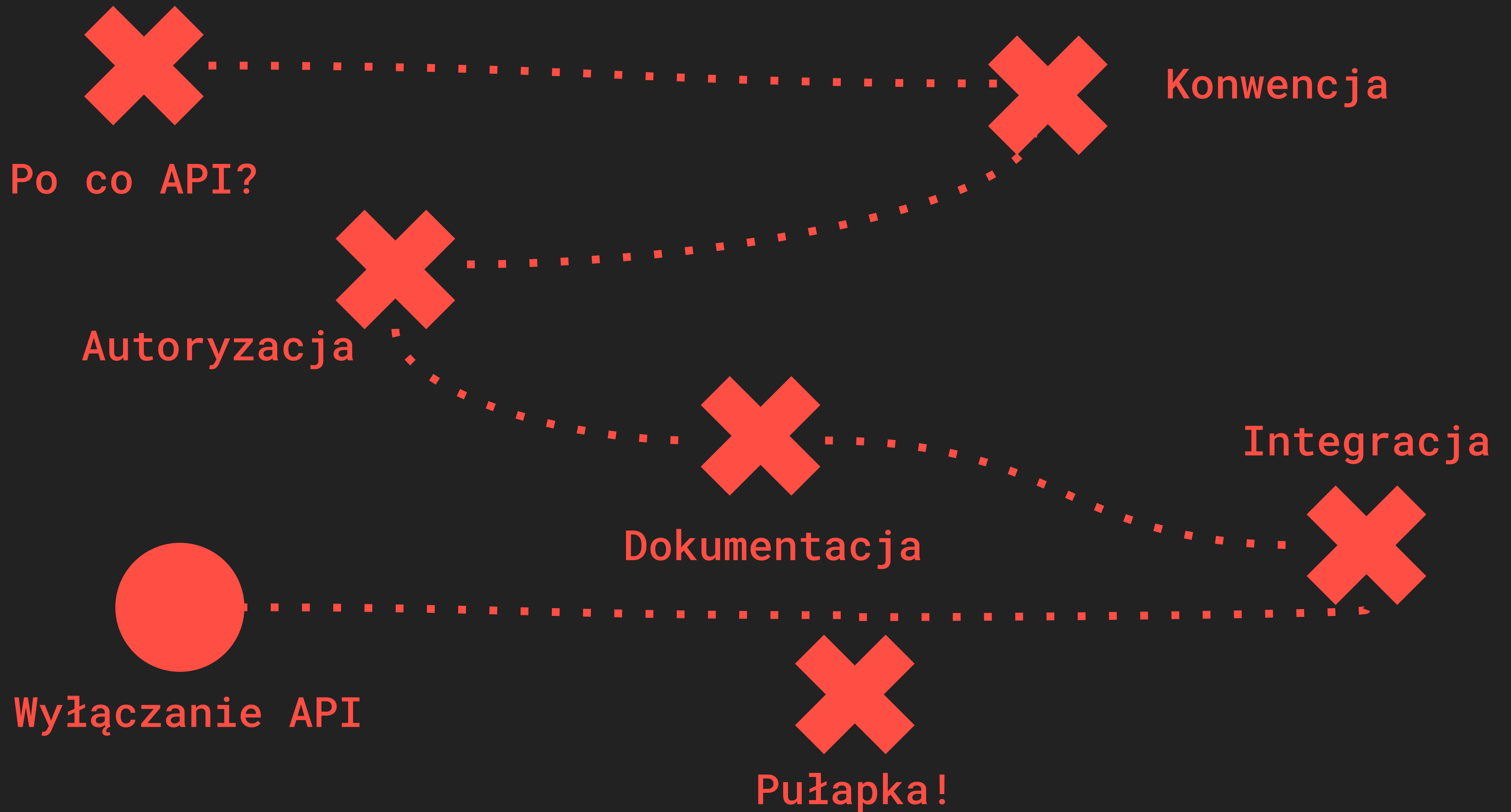
[analizuj skutki zmian]



[aktualizuj dokumentację
przy zmianach]



[dobierz sposób
wersjonowania do zmian]



wyłączenie API

dłaczego?

- mamy nowe
- nikt nie używa
- nie opłaca się rozwijać



twórca

klient



wyłączanie API

dłaczego?

- mamy nowe
- nikt nie używa
- nie opłaca się rozwijać



twórca

o nie!

- być skutecznie poinformowany o tym, że nastąpi wyłączenie
- w dokumentacji
- bezpośredni kontakt (np. mail)
- mieć namiar i czas na migrację na np. nową wersję
- być może trochę więcej czasu :)

klient





[komunikuj wyłączenie
API]



[daj czas na
'przesiadkę']



[monitoruj użycie
'starej' wersji]



[to potrwa dłużej niż
myślisz]





[API pozwala na
integrację z innymi
aplikacjami]



[w API
stawiaj na REST]



[autoryzacja w API
działa inaczej niż na
www]



[klient poznaje API
dzięki dokumentacji]



[dystribucja aplikacji
mobilnych trwa długo]



[dobierz sposób
wersjonowania do zmian]

Adrian
Ruby Developer

Od lat rozwija API inFakt



DZIĘKUJEMY!

Darek
android / iOS / microservices

Od lat jest klientem API inFakt

