API - Application Programming Interface
REST Representational State Transfer

7 Zasad przy budowie REST API

1. Stosuj rzeczowniki w adresach

Adresy w API REST-owym powinny być RZECZOWNIKAMI – wskazywać na konkretne zasoby, do których chcesz się dostać. Na przykład:

```
/books
/books/once-upon-a-time-in-a-hollywod
/books/831231201
/cars/audi/a4/b6/2009/diesel/2.0
/invoices/google-analytics/2019/03/12/151
```

W REST API nie stosuje się czasowników i wszystkie tego rodzaju adresy są błędne, jak na przykład:

```
/getBooks
/books/once-upon-a-time-in-a-hollywod/update/status
/invoices/google-analytics/2019/03/12/151/markAsPaid
```

Dzięki stosowaniu hierarchicznej struktury obiektów i prostej – RZECZOWNIKOWEJ – adresacji zasobów API budowane za pomocą REST-a jest dużo bardziej intuicyjne, skalowalne i prostsze w obsłudze przez wszystkich zainteresowanych.

2. Używaj metod zgodnie z przeznaczeniem

REST opiera się na wzorcach wypracowanych w protokole HTTP. Dlatego korzysta z metod znanych z HTTP.

Przykłady jak to powinno wyglądać w rzeczywistości?

- GET /books – pobiera listę książek,
- POST /invoices/google-analytics – dodaje fakturę,
- PUT /books/once-upon-a-time-in-a-hollywod – nadpisuje konkretną książkę,
- PATCH /books/831231201 – aktualizuje konkretną książkę,
- DELETE /invoices/google-analytics/2019/03/12/151 – usuwa fakturę z danej ścieżki,
- HEAD /cars/audi/a4/b6/2009/diesel/2.0 – zwraca informację czy dany zasób istnieje
- OPTIONS /books – zwraca listę metod, które są dostępne na danym zasobie.

3. Korzystaj ze statusów HTTP

Protokół HTTP, na którym opiera się REST, to nie tylko adresy i metody – ale także kody statusów. Tych jest ponad 50. Ich największą zaletą jest to, że są ustandaryzowane i wszędzie znaczą to samo.

4. Zwracaj dane w kopercie

Serwisy REST-owe nie mają narzuconej i wymaganej struktury danych. Z jednej strony jest to zaleta – bo daje dużą elastyczność, a z drugiej problem, bo nie możemy łatwo zdefiniować kontraktu, według którego powinny być przesyłane dane.

5. Zwracaj wartościowe odpowiedzi błędów

Gdy coś w komunikacji z twoim API poszło nie tak staraj się zwracać szczegółowe informacje.

Wiadomość Oops, something went wrong może nie być najlepszym pomysłem Stosuj kopertę z punktu 4 i umieszczaj w niej takie informacje jak:

- code – wewnętrzny kod błędu Twojej aplikacji, który pozwoli Ci zidentyfikować problem,
- msg – wiadomość tekstowa dotycząca błędu,
- details – opcjonalnie informacje ze szczegółami danego problemu.

Jak może wyglądać taka wiadomość? Na przykład:

```
{
  "error": {
    "code": "SK30",
    "msg": "Insufficient account balance",
    "details": "Unable to transfer 100 PLN. Not enough funds on the account"
  }
}
```

6. Stosuj paginację…

Masz endpoint do zwracania listy obiektów, na przykład książek w sklepie internetowym. Zwracasz więc wszystkie obiekty. A co jeśli obiektów nie jest 20, a 200? Jeszcze powinno być ok. A 2 tysiące? Albo… 2 miliony?
Za każdym razem stosuj paginację i zwracaj tylko podzbiór wszystkich obiektów. Podziękują Ci za to klienci, frontendowcy oraz szef… który będzie płacił mniejsze rachunki za serwery backendowe.

7. Wystaw Open API

Jeśli budując API na bieżąco współpracujesz z jego klientami warto wystawić Open API.