

Rest API

Maciej Wiatr

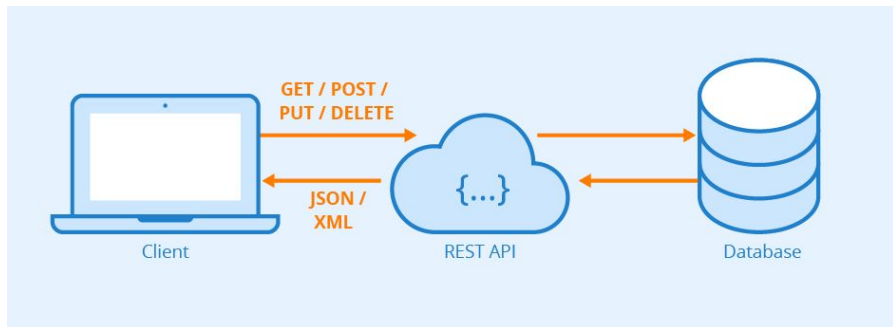
Czym jest Rest API

REST API jest to uniwersalny interfejs HTTP do komunikacji między oprogramowaniem klienta a serwerem za pośrednictwem sieci. REST (Representational State Transfer) oznacza architekturę w której zdefiniowano konkretne metody, zasoby oraz bezstanowość, API (application programming interface) jest to po prostu interfejs, a HTTP (Hypertext Transfer Protocol) jest to protokół przesyłania dokumentów w sieci WWW.

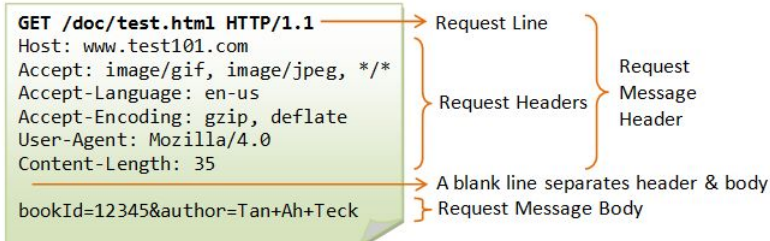


W jaki sposób przebiega komunikacja w Rest API

Komunikacja w Rest API opiera się na zapytaniach wysyłanych przez klienta na odpowiednie adresy, nazywane najczęściej endpointami, oraz odpowiedziach od strony serwera. Rest API korzysta z metody oraz ścieżki zapytania do określenia jaka akcja powinna być wykonana oraz z ciała zapytania/odpowiedzi żeby przekazywać dane.



HTTP Method	CRUD Action
GET	Retrieve a resource.
POST	Create a resource.
PUT	Update a resource.
DELETE	Delete a resource.



Przykłady

HTTP Method	Action	Examples
GET	Obtain information about a resource	<code>http://example.com/api/orders</code> (retrieve order list)
GET	Obtain information about a resource	<code>http://example.com/api/orders/123</code> (retrieve order #123)
POST	Create a new resource	<code>http://example.com/api/orders</code> (create a new order, from data provided with the request)
PUT	Update a resource	<code>http://example.com/api/orders/123</code> (update order #123, from data provided with the request)
DELETE	Delete a resource	<code>http://example.com/api/orders/123</code> (delete order #123)

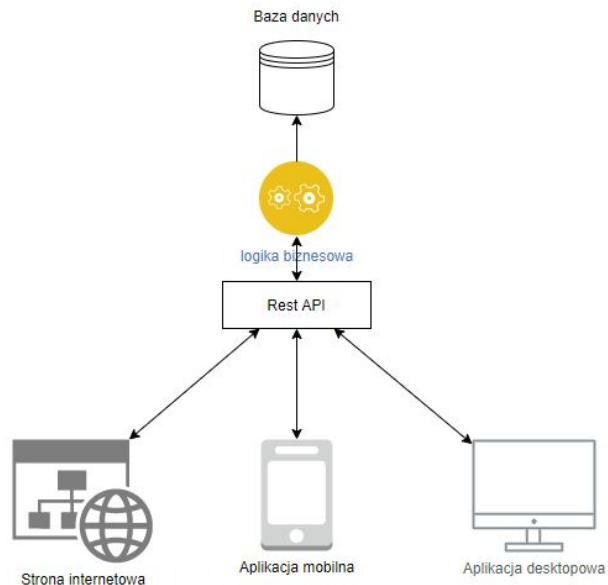


Dlaczego używamy Restowych
api zamiast standardowego
podejścia

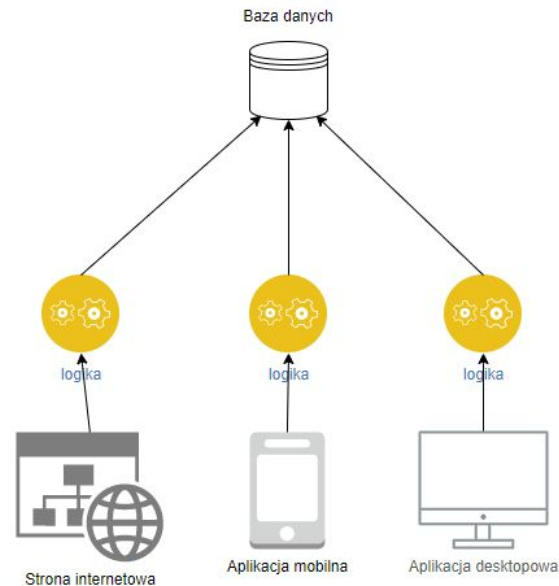
Uniwersalność

W podejściu bez rest api każda próba kontaktu z np. bazą danych musi być bezpośrednio zakodowana w aplikacji, w przypadku kiedy API istnieje mamy możliwość wykorzystania jednego interfejsu dla wszystkich naszych aplikacji, niezależnie od technologii w których zostały napisane oraz ich zastosowania.

Api

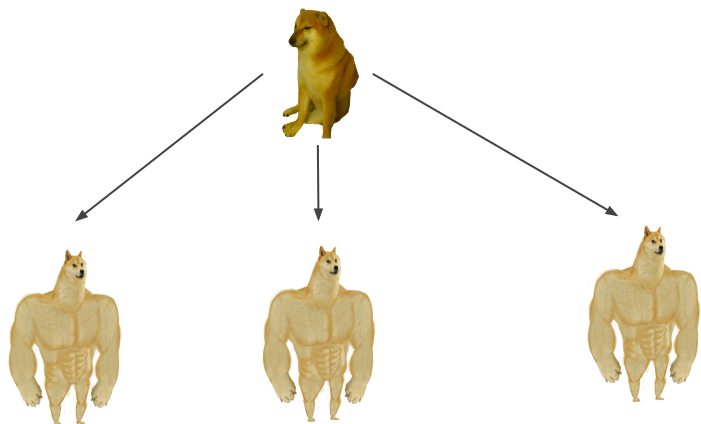


"Tradycyjne" Podejście



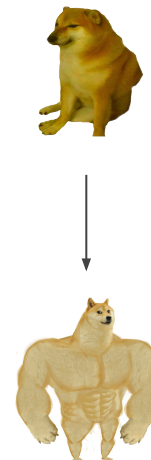
Skalowalność

Api:



Api w znacznie łatwiejszy sposób
jesteśmy w stanie skalować
horyzontalnie

Tradycyjne podejście:



Separacja warstwy widoku od logiki biznesowej

Dzięki stosowaniu API jesteśmy w stanie wyseparować logikę biznesową naszej aplikacji od warstwy widoku. Tradycyjne podejście gdzie oba te światy się mieszają można porównać do sytuacji kiedy podczas

wymiany zderzaka w samochodzie musiałbyś

wyjmować silnik.



Standaryzowana architektura

Żeby API mogło być określane restowym musi spełniać ściśle wyznaczone wytyczne.



Sample endpoints

`api_site.com/{apikey}/users`
// gets all users

`api_site.com/{apikey}/users/{userId}`
// gets a specific user

`api_site.com/{apikey}/rewards`
// gets all rewards

`api_site.com/{apikey}/rewards/{rewardId}`
// gets a specific reward

`api_site.com/{apikey}/users/{userId}/rewards`
// gets all rewards for a specific user

`api_site.com/{apikey}/users/{userId}/rewards/{rewardId}`
// gets a specific reward for a specific user

A "RESTful web service" has "endpoints" that return "resources."

In a well-designed API, you can predict the logic of the requests.



Kiedy nie warto użyć Rest API?

Nie użyłbym rest api gdyby

Moja aplikacja była prosta i nie miała
by perspektyw na gwałtowny rozwój.

Kiedy chciałbym skorzystać z
gotowego rozwiązania pokroju
wordpress





Demo