

# Metodyka tworzenia portali biznesowych

## WYKŁAD 9

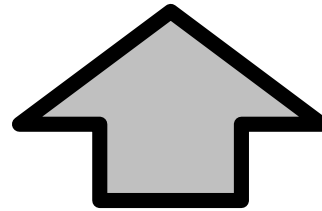
# URI

- URI (*Uniform Resource Identifier*) – ujednolicony identyfikator zasobów.
- URL jest szczególnym przypadkiem URI (URL reprezentuje adres zasobu).

# REST

- REST (*REpresentational State Transfer*) – transfer stanu reprezentacyjnego
  - podejście architektoniczne określające sposób przesyłania informacji w webowych aplikacjach rozproszonych zaproponowane w roku 2000 przez R.T. Fieldinga.
- REST nie jest protokołem, jest zbiorem zasad dostarczania usług sieciowych (typu RESTful).

# REST



# REST

- 6 zasad:

## ZASADA 1

### - ujednolicony interfejs

- żądania dotyczące zasobu pochodzące od różnych klientów muszą wyglądać tak samo
- ten sam zasób nie może mieć dwóch różnych identyfikatorów URI

# REST

- 6 zasad:

## ZASADA 2

- odseparowanie klienta od serwera
  - klient i serwer działają niezależnie
  - klient i serwer wchodzą w interakcję tylko za pomocą żądań i odpowiedzi

# REST

- 6 zasad:

## ZASADA 3

### - bezstanowość

- nie występują żadne sesje
- każde żądanie klienta powinno być niezależne od kontekstu
- każde żądanie klienta powinno zawierać komplet informacji potrzebnej serwerowi

# REST

- 6 zasad:

## ZASADA 4

### - cache-owalne zasoby

- odpowiedź serwera powinna zawierać informacje czy przesłane dane są cache-owane czy nie
- zasoby cache-owane powinny być dostarczone z numerem wersji



# REST

- 6 zasad:

## ZASADA 5

- system wielowarstwowy

- może występować kilka warstw serwerów pomiędzy serwerem wysyłającym odpowiedź i klientem

# REST

- 6 zasad:

## ZASADA 6 (opcjonalna)

- kod na żądanie

- odpowiedź serwera może zawierać kod, np. JavaScript, który może zostać wykonany przez klienta

# RESTful

- RESTful wykorzystuje przede wszystkim cztery metody protokołu HTTP:
  - POST
  - GET
  - PUT
  - DELETE

# RESTful

- Podstawowe usługi RESTful wykorzystujące protokół HTTP realizują funkcje odpowiadające tradycyjnym operacjom bazodanowym nazywanym CRUD:
  - Create – utworzenie – **POST**
  - Read – odczyt – **GET**
  - Update – aktualizacja – **PUT**
  - Delete – usunięcie - **DELETE**

# RESTful

- Operacje wykonywane są na zasobach w systemie.
- Zasoby identyfikowane są przez adres URI.
- Przesyłane są reprezentacje zasobów, np. w formatach:
  - JSON,
  - XML,
  - HTML.

# RESTful

- Kolekcje zawierają wiele zasobów tego samego typu.

# RESTful

- Adresy URL usług typu RESTful zawierają tylko informacje o zasobie lub kolekcji zasobów, np.:

`https://api.github.com/users/Tornajane`

## RESTful

- Do zmiany sposobu prezentacji kolekcji (np. poprzez filtrowanie, sortowanie) używa się parametrów zapytań w ścieżce wyszukiwania adresu URL, np.:

`http://api.joinind.in/v2.1/events`

`http://api.joinind.in/v2.1/events?  
filter=past`



## **RESTful**

- W odpowiedziach usług typu RESTfull mogą znajdować się zagnieżdżone informacje i odnośniki do innych zasobów lub kolekcji.
- Odnośniki do innych zasobów lub kolekcji nazywają się hipermediami.

# RESTful

- Aktualizowanie zasobów:
  - pobranie zasobu za pomocą żądania GET z danego URI,
  - modyfikacja zasobu,
  - przesłanie zasobu za pomocą żądania PUT do oryginalnego URI.

## Projektowanie usług

- Usługi sieciowe (Web services) odpowiedzialne są za komunikację online M2M (*machine-to-machine*) w aplikacjach rozproszonych przy wykorzystaniu sieci Internet.
- SOAP oraz REST umożliwiają tworzenie własnego API (*Application Programming Interface*).

# Projektowanie usług

## SOAP:

- ustandaryzowany protokół definiujący ścisłe reguły używania
- głównie technologie oparte o C++, Java, .NET
- **function-driven** – wykonywanie różnych operacji (funkcji, procedur, metod)
- format XML

## REST:

- zbiór zasad
- potrzebuje mniej zasobów
- jest bardziej elastyczny
- **data-driven** – operacje tworzenia, odczytu, aktualizowania i usuwania zasobów
- formaty XML, JSON, HTML i inne