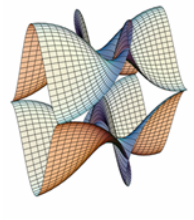


Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

RAČUNARSKI PRAKTIKUM II

Predavanje 12 - REST, SPA, vue.js

11. lipnja 2018.

Sastavio: Zvonimir Bujanović

REST

- Protokol HTTP je *stateless*.
  - Svaka komunikacija klijenta sa serverom je potpuno nezavisna sa svim prethodnim komunikacijama.
  - Sessioni u PHP su pomalo umjetan pokušaj da se zaobiđe ovo ograničenje.

- REST - Representational State Transfer
  - Arhitekturalni stil za izgradnju mrežnih aplikacija.
  - Stateless, klijent-server, *cacheable* komunikacijski protokol.
  - Koristi razne HTTP zahtjeve (GET, POST, PUT, DELETE) za slanje, čitanje i brisanje podataka.
    ⇝ implementira sve četiri CRUD (Create/Read/Update/Delete) operacije (termin iz baza podataka).

REST

Komponente i svojstva REST arhitekture:

- REST API se sastoji od mreže resursa. Svaki pojedini resurs identificira jedan logički (a ne fizički!) URL.

  Primjeri:

  - `http://www.parts-depot.com/parts/` – kolekcija resursa
  - `http://www.parts-depot.com/parts/00345` – jedan resurs

- Pristup resursu korištenjem HTTP metode:
  - GET - Vraća reprezentaciju resursa (ili kolekcije) u odgovarajućem formatu (npr. JSON) [read].
  - PUT - Zamijeni resurs (ili cijelu kolekciju) novim resursom poslanim klijentskim zahtjevom [update].
  - POST - Dodaj novi resurs u kolekciju (obično se ne koristi za cijele kolekcije) [create].
  - DELETE - Obriši resurs (ili cijelu kolekciju) [delete].

REST

- Ne postoji "stanje konekcije" niti nešto slično session-u.
  - Svaka interakcija klijenta sa serverom je posve neovisna o prethodnima.
  - Svaki novi zahtjev klijenta treba sadržavati sve informacije potrebne za njegovo izvršavanje i ne smije ovisiti o prethodnim interakcijama sa serverom.
  - Postoje rješenja kako riješiti autentifikaciju i autorizaciju klijenta (bez da se uvijek isponova šalju username i password).

- Za API koji implementira ove komponente i svojstva kažemo da je RESTful.

- Primjeri:
  Paypal, Twitter, Google Translate, Flickr, Dropbox, Bing Maps.

REST - Implementacija u PHP-u

```php
// Npr. http://bookstore.com/api/v1/books/123
$request = explode('/', $_SERVER['PATH_INFO']);
$method = strtolower($_SERVER['REQUEST_METHOD']);

switch($method) {
    case 'get':    // handle a GET request
        get_book($request); break;

    case 'post':   // handle a POST request
        post_book($request); break;

    case 'put':    // handle a PUT request
        put_book($request); break;

    case 'delete': // handle a DELETE request
        delete_book($request); break;

    default:       // unimplemented method
        http_response_code(405);
}
```

REST - Implementacija u PHP-u (GET)

```php
function get_book($request)
{
    // Iz baze dohvatimo knjigu iz request-a, npr.
    // $book[123] = [
    //     'id' => 123,
    //     'data' => [ 'title' => 'PHP Cookbook',
    //                 'author' => ... ],
    // ];

    $json = json_encode($book[123]);

    // Resurs postoji -> 200: OK
    http_response_code(200);

    // Resurs šaljemo kao odgovor:
    header('Content-Type: application/json');
    echo $json;
}
```

REST

- Logičke adrese resursa mogu se napraviti pomoću `.htaccess`:

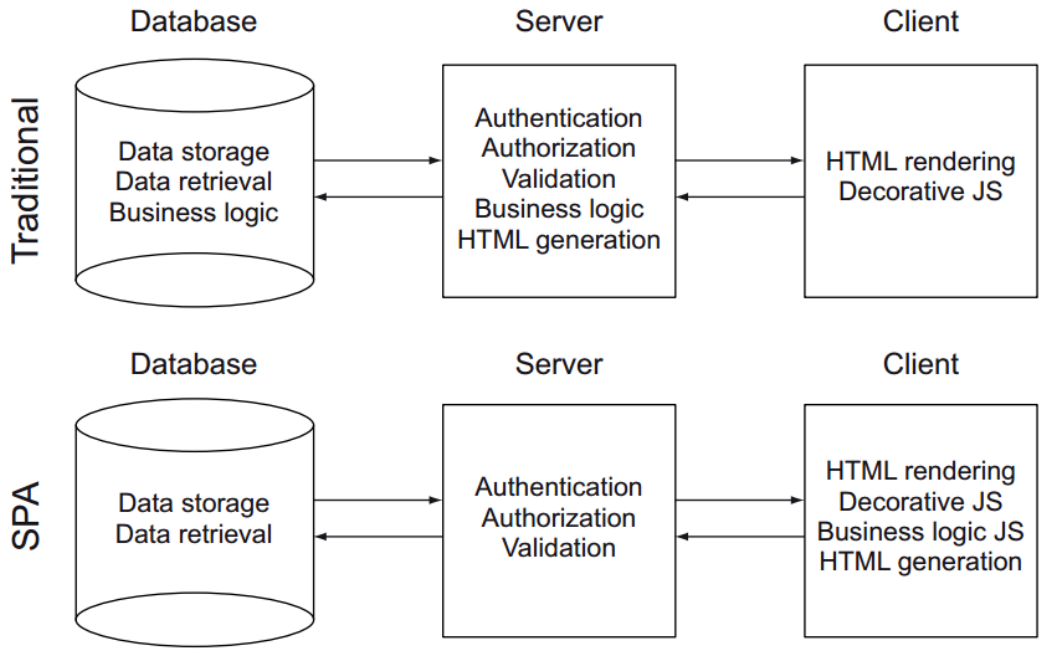
```
RewriteEngine on
RewriteBase /api/v1/
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?PATH=$1 [L,QSA]
```

  Sada se `http://bookstore.com/api/v1/books/123` mapira u `http://bookstore.com/api/v1/index.php?PATH=books/123`.

- U REST-u postoje standardizirani HTTP response kodovi:
  - 200 - OK
  - 400 - Bad Request (kriva sintaksa ili druga generička greška)
  - 401 - Unauthorized (treba dati autentifikaciju za pristup)
  - 403 - Forbidden (zabranjen pristup bez obzira na autentifikaciju)
  - 404 - Not Found (resurs ne postoji)
  - 405 - Method Not Allowed (metoda nije dozvoljena za resurs)
  - 410 - Gone (resurs više ne postoji)
  - 429 - Too Many Requests (prijeđen dozvoljeni broj upita)

Single page web-applications (SPA)

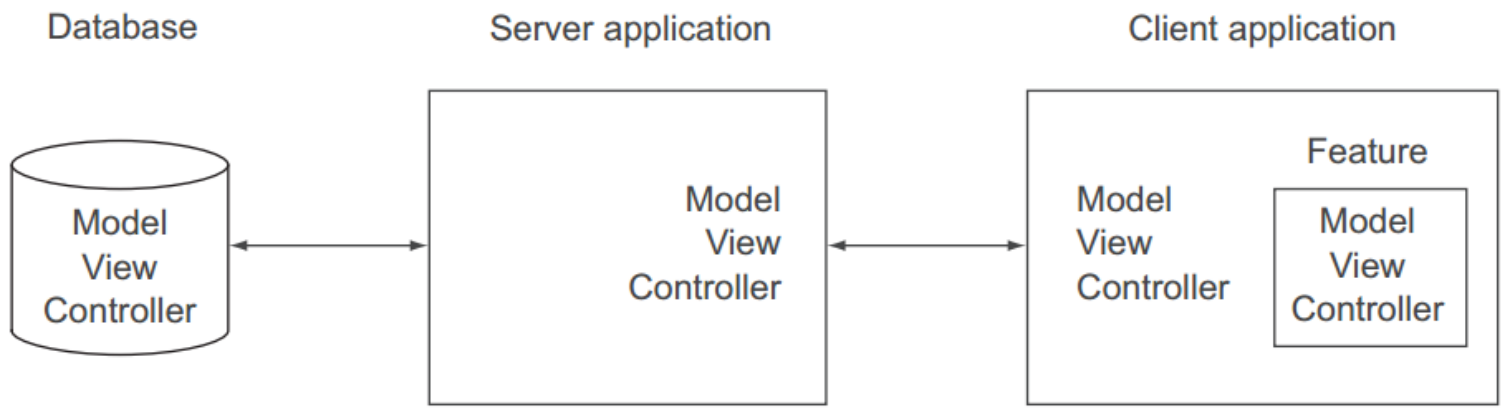
- U posljednje vrijeme je velik trend tranzicije web-aplikacija u model "Single page application".
  - Sav HTML, CSS i JavaScript se učita u prvom pristupu serveru.
  - Ovisno o korisnikovim akcijama, dodatni resursi se dinamički učitavaju i dodaju na stranicu, bez ponovnog učitavanja cijele stranice (npr. Ajax upitom na REST API).
  - Cilj je pružiti dojam tradicionalnih desktop aplikacija.
- U SPA, sva logika web-aplikacije se pomiče sa servera na klijenta [slika iz knjige: Minkowski, Powell: "SPA"]



Single page web-applications (SPA)

- Povećanje kompleksnosti klijentskog programa zahtijeva bolju organizaciju JavaScript koda.
  - Razdvojiti programsku logiku od dinamičkog generiranja HTML-a (prezentacije).
  - Razdvojiti reakciju na input korisnika od programske logike i HTML-a.
  - Déjà vu?

- MVC i prijatelji (MVVM) na klijentskoj strani.
  - Angular, Ember.js, Meteor.js, Backbone.js.
  - React.js, vue.js (fokus na View).



- vue.js
  - "Bibioteka za izgradnju interaktivnih web-sučelja".
  - Omogućuje jednostavno i reaktivno povezivanje podataka i komponenti sučelja.
  - Fokusira se na View sloj, no u njemu se mogu implementirati čitave SPA u skladu sa obrascem MVVM.
- Napravit ćemo samo dva vrlo jednostavna primjera da pokažemo osnovne koncepte.

- Resursi za učenje:
  - Izvrsni video-tutorial.
  - 5 praktičnih primjera.
  - Još dva primjera: prvi i drugi.

vue.js - Primjeri

Primjer 1:

- Cilj:
  - Unosom ispravnog imena u input prikazuje se gumb, u protivnom se prikazuje informacija o grešci.
- Ilustrira:
  - data-binding (reaktivne komponente);
  - event-handling u vue.js.

Primjer 2:

- Cilj:
  - TODO lista, zadaci se mogu dodavati i označavati kao obavljeni.
- Ilustrira:
  - iteriranje po listama podataka;
  - custom komponente.

Što nakon RP2?

- HTML, CSS
  - Bootstrap.
  - jQuery CSS efekti.
  - LESS, Sass.
- PHP - MVC
  - Laravel, Symfony, CakePHP.
- Drugi MVC okviri na serverskoj strani
  - Django, Ruby on Rails, Java Play.
  - Yesod (Haskell), Phoenix (Elixir).
- Full-stack JavaScript
  - Node.js.
  - MEAN (Mongo.db + Express.js + Angular.js + Node.js).