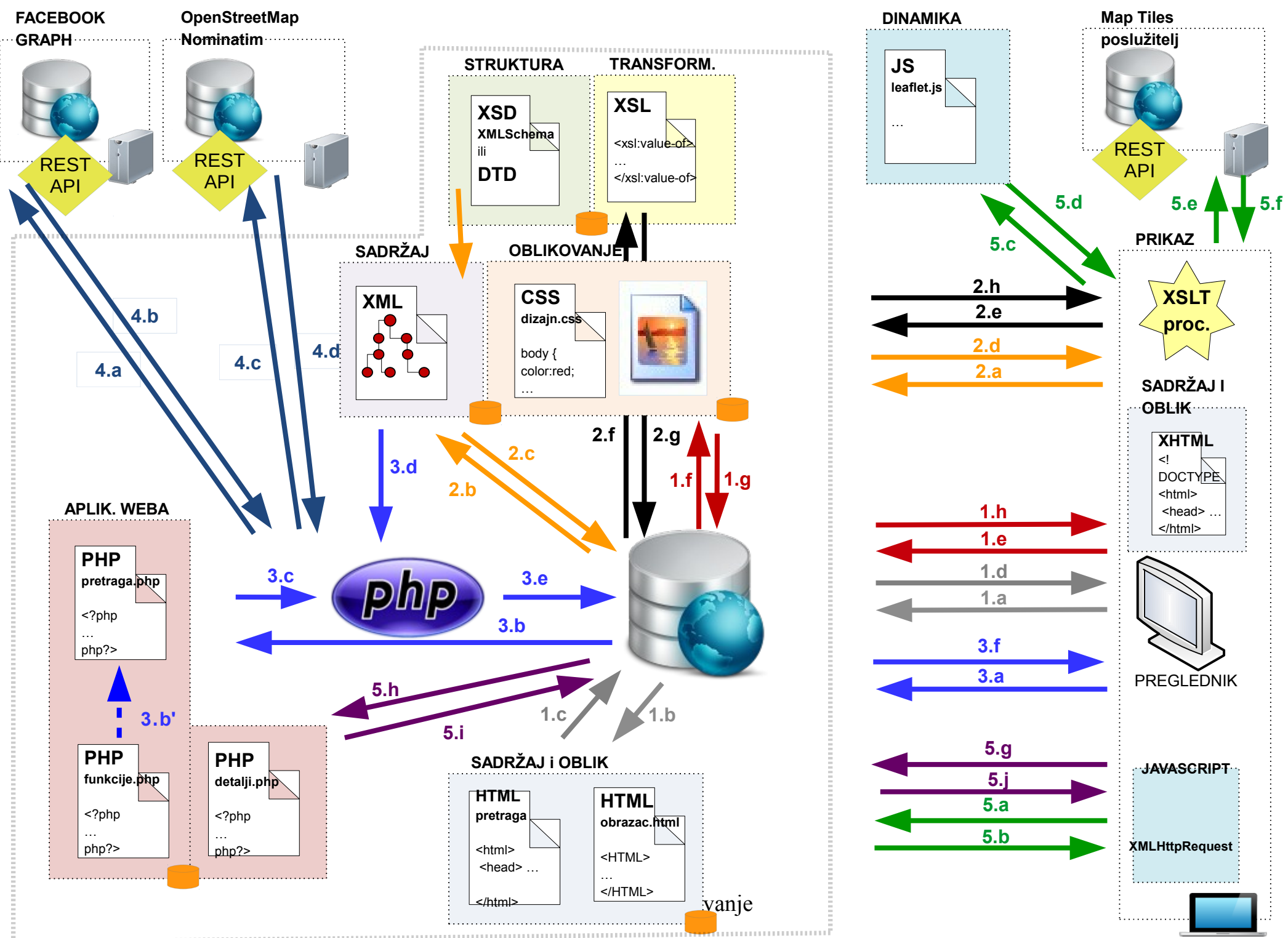


5. pripremno predavanje

- JavaScript
- XMLHttpRequest
- vanjski izvori podataka – pristup s klijenta



5. laboratorijska vježba

- **(pred)**Zadnja vježba 😊
- Završno "dotjerivanje" Web aplikacije
- Naglasak na:
 - 1)klijentskim tehnologijama (**JavaScript**)
 - 2)dohvaćanju informacija s poslužitelja
bez osvježavanja stranice (**XMLHttpRequest**)
 - 3)pristupu vanjskim izvorima podataka s klijentske strane
- Datoteke:
 - promjena HTML-a, CSS-a i postojeće PHP skripte
 - izrada manje PHP skripte: *detalji.php*
 - izrada JavaScript skripte: *detalji.js*

Cilj vježbe



- 1) Dopuniti Web aplikaciju promjenjivim elementima na strani klijenta (*DHTML, JavaScript*)
 - promjena izgleda i prikazivanje/skrivanje elemenata
- 2) Omogućiti dohvat dodatnih podataka komunikacijom klijent-poslužitelj bez osvježavanja stranice (*XMLHttpRequest*)
- 3) Povezati podatke iz vanjskih izvora s dodatnim uslugama na klijentskoj strani - *mashup*
 - U Web aplikaciju uključiti vanjske, javno dostupne biblioteke određene namjene (*JavaScript*)

Proučiti:

- **Osnove osnova jezika JavaScript**
 - uključivanje JavaScripta u HTML
 - rad s varijablama
 - pozivi funkcija
- **Koncept DHTML-a**
 - model DOM
 - rukovanje događajima (*event handling*)
 - promjena stilova
- **Objekt XMLHttpRequest**
 - slanje zahtjeva s parametrima
 - obrada - prosljeđivanje odgovora
- **Rad s bibliotekom *leaflet.js* i pločicama karata**

Zadatak (1. dio) - DHTML

- Prelaskom strelice miša preko pojedinog retka **pobožati redak** drugom bojom
- promjena HTML/JavaScript kôda kojeg generira *pretraga.php*
- registracija *event handlera* za događaj *onmouseover*


```
<tr onmouseover="promijeniBoju(this)">
    <td>
        Osoba A
    </td>
</tr>
moj_red_tablice.style.backgroundColor =
    "#336699"
```

Zadatak (2. dio) - detaljne informacije

- Tablica rezultata pretraživanja ima novi stupac "Akcija"
 - s tekстом ("Više o..."), slikom ili gumbom u svakom retku
 - klikom na **pojedini aktivni element** (događaj *onclick*) priprema se **komunikacija** s poslužiteljem
- Objekt **XMLHttpRequest**
 - nema osvježavanja stranice
 - **novoj** skripti na poslužitelju (***detalji.php***) treba poslati **parametar (id)** pojedinog osnovnog elementa
 - skripta *detalji.php* **vrlo slična** skripti *pretraga.php*
 - HTTP zahtjev, metoda GET

Zadatak (2. dio) - obrada odgovora



- Skripta *detalji.php*
 - pretražuje XML **po id-u**
 - vraća dodatne informacije o elementu određenog id-a
 - **nema komunikacije sa Facebookom!**
- Izlaz skripte je "**mali**" **HTML kôd - segment**
 - spreman za "ubacivanje" u novi dio stranice
 - sadrži sve potrebne oznake, poput `<table>`, `<div>`, definicije klasa za CSS i slično
- Na strani klijenta, po primitku odgovora, JavaScript prenosi dobiveni kôd u okvir za dodatne informacije
 - Obavješćavanje korisnika o čekanju odgovora 

Zadatak (3. dio) - *mashup*

- Vanjske datoteke i izvori podataka
- Uvijek postoji kratki uvod u korištenje biblioteke
- Osnovna ideja:
 - preuzeti .js datoteke i uključiti ih u svoj kôd
 - **pri aktivaciji događaja** (*onclick*) iscrtati ili ažurirati kartu
 - ucrtati marker na kartu
- Tekst se obično predaje kao HTML
 - dozvoljena sva formatiranja, definiranje klasa za CSS...

Zadatak (3. dio) - *mashup*

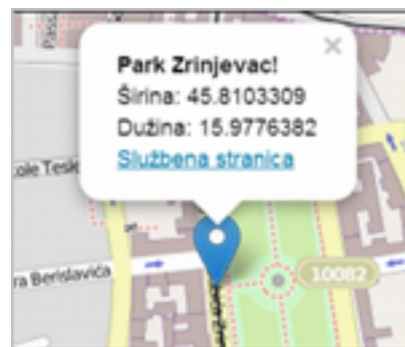
- Prikaz geografske lokacije na karti
- Vanjske biblioteke i izvori podataka
 - *leaflet.js* – biblioteka za prikaz pločica karte
 - različite skupine pločica (*tiles*)
- Dio s kartom **nije** vidljiv na početku
 - prikazuje se zajedno s pozivom za prikaz detalja
- Podaci
 - sakriveni u JavaScript kôdu, izgenerirani u *pretraga.php*
 - **nema ponovne komunikacije s poslužiteljem!**

Zadatak (3. dio) - *mashup*



- Prikazati

- naziv
- geografske koordinate
- link na službenu stranicu
- **adresu**



- (ne piše u skripti :-()

- Uključivanje biblioteke

```
<link rel="stylesheet"
```

```
    href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css" />
```

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js" />
```

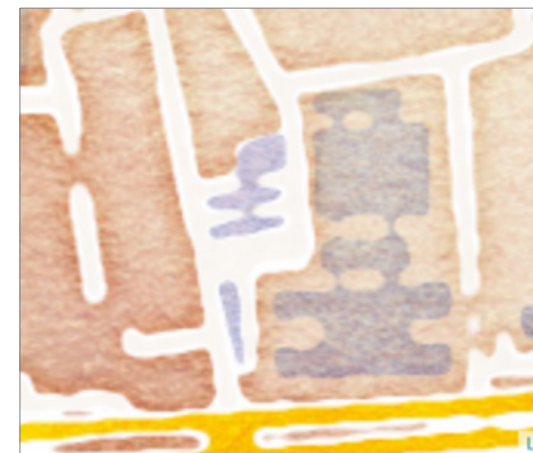
- Iscrtavanje mape

```
var map = L.map(<div_id>).setView([<širina>, <dužina>],  
    razina_povećanja);
```

Zadatak (3. dio) - *mashup*



- Izvori pločica za kartu
 - različiti uvjeti korištenja
 - isti način korištenja
- primjeri pločica:
 - <http://leaflet-extras.github.io/leaflet-providers/preview/>

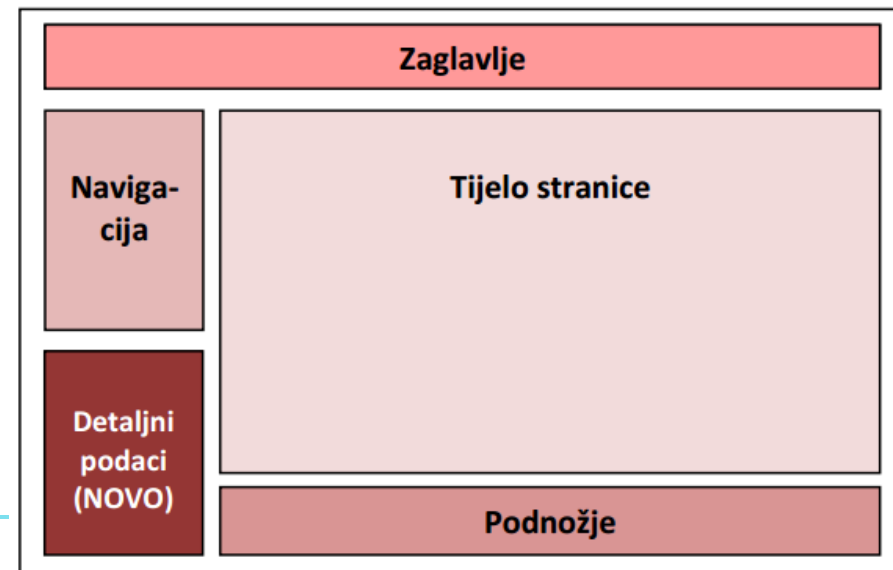


```
var OpenStreetMap_Mapnik =  
L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',  
{ attribution: '&copy;  
<a href="http://openstreetmap.org">OpenStreetMap</a>  
contributors, <a href="http://creativecommons.org/licenses/by-  
sa/2.0/"> CC-BY-SA</a>'}  
);
```

Promjene u postojećim datotekama



- *pretraga.php*
 - smanjiti tablicu (treba imati 3 - 5 stupaca)
 - dodati stupac Akcija
 - dodati JavaScript kôd za
 - identificiranje podatka (**id**) za svaki redak
 - reakcije na događaje (*može i u .js datoteci*)
 - priprema podataka za ispis na mapu za svaki redak
 - **ispod navigacije dodati okvir za dodatne podatke** (`<div>`)
 - **ispod tablice dodati skriveni okvir za kartu**
- CSS stilovi
 - prilagoditi izgled i strukturu



Brzi uvod u:

- JavaScript
- XMLHttpRequest

Uključivanje JavaScripta

- U **<head>** ili **<body>** dijelu
 - razlike u trenutku učitavanja i "garanciji" učitavanja prije nekog događaja
- *Inline*
- **Vanjska JavaScript datoteka**

```
<head>
  <script type="text/javascript" src="mojJS.js">
  </script>
</head>
```

Sintaksa JavaScripta

- Slična C-u/Javi/PHP-u...

- if, for, pridruživanja...
- **new** (instanciranje objekata)

- Spajanje stringova (+)

`datum = godina + '-' + mjesec + '-' + dan;`

- Varijable

- globalne i lokalne
- tip nije određen pri deklaraciji, ovisan o vrijednosti varijable, promjenjiv

`var id = 5;`

- Funkcije

```
function ispisi (mojTekst) {
    alert(mojTekst);
    return true;
}
```


Događaji (*events*)

- Događaji se odnose na elemente HTML-a (odlomke teksta, linkove, gumbe, obrasce...)
- Događaje okidaju:
 - miš (**onclick**, **onmouseover**, **ondblclick**...)
 - tipkovnica (**onkeypress**, **onkeyup**...)
 - HTML objekti i obrasci (**onload**, **onresize**, **onsubmit**...)
- Povezivanje događaja na nekom elementu s naredbama/funkcijama koje treba izvršiti

Registracija događaja

- Povezivanje događaja na nekom elementu s naredbama/funkcijama koje treba izvesti
- Inline (unutar HTML-a)
 - ne razdvaja JavaScript od HTML-a (logika i prezentacija)


```
<a href="link.html" onclick="promijeni()">
```
 - još uvijek vrlo raširen
- Uobičajeni (unutar JavaScript kôda)
 - razdvaja logiku od prezentacije – **preporučeno**

```
element.onclick = promijeni;
```
- W3C DOM Level 2
 - dodatne, napredne mogućnosti


```
element.addEventListener('click', promijeni, false)
```

DOM: važna svojstva i metode

- Pristupanje elementima (W3C)
 - jednom elementu s definiranim atributom *id*
`document.getElementById()`
 - svim elementima s definiranim atributom *name*
`document.getElementsByName()`
 - svim elementima iste oznake
`document.getElementsByTagName()`

```
nav = document.getElementById("navigacija");
```

- Dobro je provjeriti je li element pronađen
 - ne želimo raditi s *null* vrijednostima i dobivati JS pogreške ☹️

DHTML: važna svojstva i metode



- Dinamička promjena svojstva elemenata

- promjena sadržaja elementa

```
document.getElementById('naslov').innerHTML = 'Novi  
naslov';
```

- promjena izgleda

```
document.getElementById('naslov').style.color = 'red';
```

- promjena vidljivosti

```
document.getElementById('naslov').style.visibility =  
'hidden';
```

```
document.getElementById('naslov').style.display =  
'none';
```

- promjena naziva gumba

```
document.getElementById('gumb').value = 'Novi';
```

DOM: primjer dodavanja elementa



- Dodavanje teksta na stranicu
 - Izravni upis – zastarjeli, neispravan način
`document.write("<p>Novi tekst</p>")`
 - Dodavanje novog elementa u DOM

```
var element = document.createElement("p");  
var tekst = document.createTextNode("Novi  
tekst");  
element.appendChild(tekst);  
document.getElementsByTagName("body").item(0).  
appendChild(element);
```
- DOM način je kompliciraniji, no omogućuje pozicioniranje elemenata i kasniju uporabu elementa

XMLHttpRequest (AJAX)

1. Instanciranje objekta
2. Povezivanje s URL-om
 - odabir metode HTTP zahtjeva
3. Slanje zahtjeva
4. Čekanje odgovora poslužitelja (asinkrono!)
 - "život" stranice za vrijeme čekanja normalno nastavljen
5. Ako je odgovor OK, poziv funkcije za parsiranje odgovora
6. (najčešće) Prikaz rezultata unutar Web stranice

Demistificirani XMLHttpRequest



```
if (window.XMLHttpRequest) {  
    var xmlHttp = new XMLHttpRequest();  
}  
else {  
    var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");  
}  
xmlHttp.open("GET", "http://www.primjer.hr");  
xmlHttp.onreadystatechange = promijeniStranicu;  
xmlHttp.send(null);
```

provjeri tip
Web preglednika,
instanciraj objekt

poveži objekt s
URL-om i odaberi
metodu zahtjeva

imenuj funkciju koja
će se izvršiti po
promjeni stanja
zahtjeva

pošalji zahtjev
(u metodi POST,
ovdje se dodaju
parametri)

Demistificirani XMLHttpRequest



```
function promijeniStranicu() {  
  if (xmlHttp.readyState == 4) {  
    if (xmlHttp.status == 200) {  
      var odgovor = xmlHttp.responseText;  
      alert ("Dobio sam " + odgovor);  
    }  
    else {  
      alert ("Nešto loše se dogodilo");  
    }  
  }  
}
```

XML zbog DOM
parsiranja treba primiti
s .responseXML

funkcija koja
se izvršava pri
promjeni
stanja zahtjeva

provjeri je li
dovršeno
primanje
odgovora

provjeri je li
odgovor OK

Slanje parametara, primanje podataka

- Za metodu **GET**, parametri se šalju **unutar URL-a**
 - npr. `http://www.fer.hr/detalji.php?id=523&show=simple`
- Za metodu **POST**, parametri se pišu jednako kao i za GET, no šalju se prilikom poziva **funkcije *send***
 - npr. `req.send("id=523&show=simple");`
- Primanje podataka
 - `req.responseText` -> obično ga se "preseli" u neki `<div>`
 - `req.responseXML` -> obično ga se transformira DOM-om

Upucavanje nevolja (*troubleshooting*)

- Čudne sintaksne pogreške vezane za *stringove*
 - pomiješali ste jednostruke (') i dvostruke navodnike (")
 - pazite na navodnike unutar navodnika
 - `<tr onmouseover="alert('Čiča "Gliša"');">`
 - escape znakovi (\", \')
 - stavljanje teksta u varijablu (jedna razina navodnika manje)
 - promjena HTML navodnika u jednostruke
- Prelazak u "novi redak" ne radi. Čak ni \n ne radi.
 - HTML kôd -> `
` ili `</div>`
- XMLHttpRequest mi ne vraća ništa ili nešto krivo
 - probajte u preglednik upisati izravni URL
(.....detalji.php?id=24), radi lakšeg *debugiranja*

Pomoćni alati

- Vrlo, **vrlo** korisni
- JavaScript *debuggeri*
- Programi za praćenje prometa protokola HTTP
- Alati unutar preglednika
 - Developer Tools
 - DOM Inspector
 - Error Console
 - Network
 - ...

6. pripremno predavanje



Organizacija

- Neobavezna vježba!
 - ali ulazi u 50% za prag...
- Potrebno je "prijaviti" željeni API
 - link na stranici predmeta
- Odgovaranje vježbe zajedno s 5. vježbom
 - također zajednička predaja kôda na *Moodle*

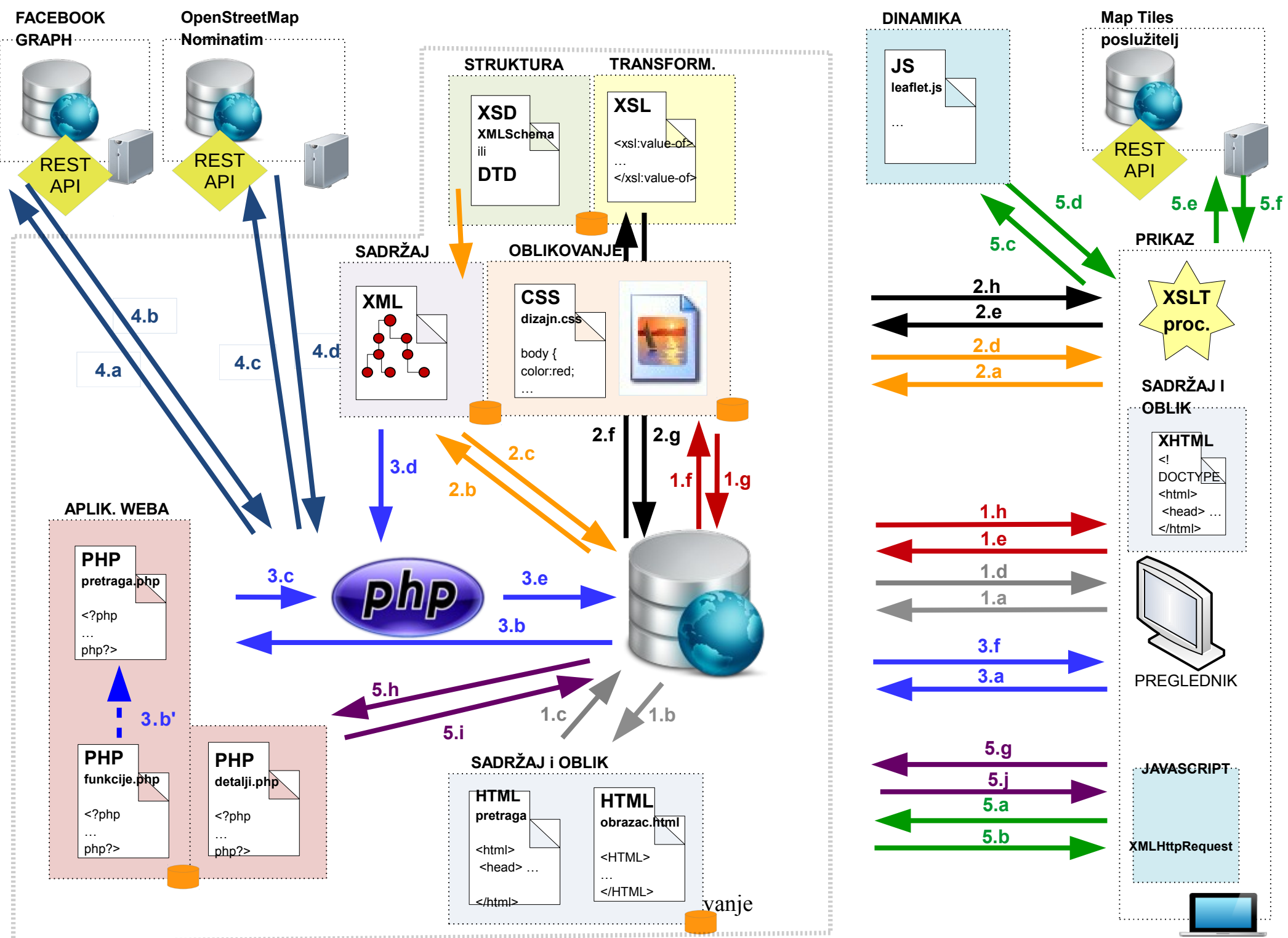
Odabir API-ja

- Web API, ne nužno REST
- **Serverska ili klijentska** strana – svejedno 😊
- Trebaju povećati korisnost aplikacije Weba!
- Moguća inspiracija:
 - <http://www.programmableweb.com/apis/directory>

Detalji



- Obratite pažnju na:
 - uvjete korištenja
 - korisnost
 - uklapanje u dizajn
- Ocjenjivanje
 - složenost i uloženi trud?
 - originalnost?
 - kreativnost?
 - "opći dojam"?
 - do 4 boda
 - bodovanje je **relativno**
 - može doći do promjene nakon pregleda svih radova





Pitanja?

- Probleme i pitanja možete i naknadno postaviti u forumu na stranicama predmeta