

Arhitektura, protokoli i usluge weba (222464)

Upute za 1. laboratorijsku vježbu

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

10.11.2022

Zadatak

Osmisliti, oblikovati i programski ostvariti jednostavnu web uslugu koja izlaže programske funkcionalnosti koristeći načela REST arhitekturnog stila. Web uslugu možete osmisliti sami ili možete kopirati neku od popularnih web usluga ili aplikacija (npr. Twitter¹ ili Instagram²). Obvezne komponente su:

- REST sučelje
- dokumentacija za REST sučelje
- autentikacija korisnika
- automatizirano ispitivanje funkcionalnosti

Nije potrebno ostvariti bogato korisničko sučelje, spajanje na bazu podataka (iako je neki oblik perzistencije podataka koristan, npr. SQLite) niti "administracijsko" sučelje.

Izgrađena web usluga mora imati **barem DVIJE** grupe sredstava koji se preslikavaju na URL adrese. Npr. web usluga za prikazivanje fotografija ima URL adresu za pristup fotografijama (npr. /photos) i URL adresu za pristup profilu korisnika (npr. /users). Dodatno, potrebno je podržati **ug-niježđeno dohvaćanje sredstava na barem jednom paru sredstava**; npr. opisana usluga mora omogućiti dohvat svih fotografija određenog korisnika (npr. /users/1/photos).

ZA SVAKO izloženo sredstvo ostvariti SVE CRUD METODE KOJE ZA TO SREDSTVO IMAJU SMISLA, a ZA BAREM DIO sredstava ostvariti SVE CRUD metode. Nije prihvatljivo osmisliti web uslugu u kojoj su sva sredstva semantički toliko jednostavna da za njih nema smisla ostvariti više od jedne CRUD metode.

¹<http://twitter.com>

²<http://instagram.com>

Zaštititi pristup sredstvima koristeći neki od standardnih mehanizama autentikacije (BasicAuthentication, Sessions, ...).

Programsko sučelje mora biti dostupno na izabranom sufiksu URL adrese (npr. <http://localhost:8000/api>). Korijenska URL adresa (npr. <http://localhost:8000/>) mora sadržavati **dokumentaciju u obliku popisa svih URL adresa te svih pripadajućih CRUD metoda s opisom funkcionalnosti**. Izgled dokumentacije je proizvoljan, a dozvoljeno ga je generirati i automatski.

Za potrebe izgradnje web usluge dozvoljeno je koristiti bilo koji programski jezik i platformu. Neki od popularnijih radnih okvira za izgradnju web usluga/aplikacija su Django/Python, Ruby On Rails/Ruby, CodeIgniter/PHP, node.js/Javascript itd. **NIJE dozvoljeno** korištenje naprednih sustava za upravljanje sadržajem (CMS) - cilj je samostalno ostvarenje web usluge!

Ako niste sigurni u kojem programskom jeziku ili radnom okviru rješavati ovaj zadatak, preporuča se korištenje programskog jezika **Python** te **Django** radnog okvira (<https://www.python.org/>, <https://www.djangoproject.com/>). Dodatno, ako se odlučite za preporučenu tehnologiju, moguće je koristiti i **Django REST framework** (<http://www.django-rest-framework.org/>). Navedena kombinacija omogućuje automatiziranu izgradnju i ispitivanje većeg dijela ove laboratorijske vježbe te predstavlja jedan od industrijskih standarda za izgradnju REST web usluga.

Automatizirano ispitivanje funkcionalnosti

Za izgrađenu web uslugu potrebno je omogućiti automatizirano ispitivanje funkcionalnosti. Preporučeni pristup je korištenje alata za **automatizirano ispitivanje ugrađenog u radni okvir koji koristite** (proučite samostalno dokumentaciju). Ugrađeni alati za ispitivanje najčešće koriste neku inačicu xUnit ispitivanja gdje se ispravnost ispituje **assert*** funkcijama. Alternativno, **ali NE i preporučeno**, moguće je koristiti Selenium (<http://seleniumhq.org/>) ili curl (<http://curl.haxx.se/>).

Za SVAKU izloženu kombinaciju URL adrese i HTTP metode napisati barem jedan ispitni slučaj koji omogućava provjeru ispravnosti usluge nakon izmjene programskog koda i/ili nakon postavljanja usluge na novi poslužitelj. Oblik pojedinog ispitnog slučaja je proizvoljan. Preporučeno je ispitne slučajeve razvrstati u nekoliko zasebnih grupa koje demonstriraju (ispituju) funkcionalnost određenog segmenta web usluge, npr. stvaranje,

dohvaćanje, brisanje i pretraživanje objekata.

Ispitivanje mora biti automatizirano. Ako koristite neki od radnih okvira za testiranje, automatizirano ispitivanje je podržano i može se relativno jednostavno ostvariti. Curl naredbe možete objediniti u skriptu, npr.:

1. Ispitivanje slanja statusa na društvenu mrežu:

```
curl -u username:password -d status="test" http://test-web-usluga/status
```

2. Ispitivanje web obrasca:

```
curl http://test-web-usluga/obrazac -F "polje1=a" -F "polje2=b"  
curl http://test-web-usluga/trazi/&polje1=a
```

Dodatno, ako koristite curl onda je potrebno **“ručno” ispitati** svaki odgovor od poslužitelja. Npr. (bash):

```
response = $(curl http://test-web-usluga/test)  
if [ $response -eq ... ] then ...
```

Što je potrebno znati na predaji laboratorijske vježbe?

1. pokrenuti i demonstrirati rad izgrađene web usluge i po potrebi uvesti manje modifikacije u programski kod
2. pokrenuti automatizirane ispitne slučajeve i pokazati da je izgrađena usluga funkcionalna

Predaja vježbe

Termin predaje

Vježba se izrađuje samostalno, ili uz konzultacije s nastavnicima, kod kuće. Ostvarena vježba demonstrira se usmeno u terminu određenom prema službenom kalendaru nastave na FER-u.

Način predaje

Vježba se predaje nastavniku/asistentu u fakultetskoj učionici. Potrebno je donijeti vlastito računalo s unaprijed pripremljenom okolinom za demonstraciju vježbe.

Kako bi ispitivanje proteklo što brže i uz manje zastoja, savjetujemo pripremu programskog koda i okoline za demonstraciju **prije neposredne predaje**.