

Upute za 1. laboratorijsku vježbu

Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu
Računarstvo zasnovano na uslugama

November 3, 2017

Zadatak

Vaš zadatak je osmisлити, oblikovati i ostvariti jednostavnu web uslugu koja izlaže sučelje koristeći REST načela. Web uslugu možete sami osmisлити ili možete kopirati neku od popularnih web usluga ili aplikacija koje često koristite (npr. Twitter¹ ili Instagram²). Obavezne komponente su:

- REST sučelje
- dokumentacija za REST sučelje
- autentifikacija korisnika
- automatsko ispitivanje

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

Izgrađena usluga mora imati **barem DVIJE** grupe sredstava koji se preslikavaju na URL-ove. Npr. web usluga za prikazivanje fotografija mora imati (barem) jedan URL za fotografije (npr. /photos/) i URL za korisnike (npr. /users/). Dodatno, podržati **ugniježđeno 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 te zaštititi pristup metodama koristeći neki od standardnih mehanizama autentifikacije (BasicAuthentication, Sessions, ...).**

Programsko sučelje može biti dostupno kao poddirektorij (npr. `http://localhost:8000/api`). Korijski direktorij mora sadržavati **dokumentaciju**

¹`http://twitter.com`

²`http://instagram.com`

u obliku popisa svih URL-ova te svih pripadajućih CRUD metoda s opisom funkcionalnosti! Izgled dokumentacije je proizvoljan te može biti automatski izgrađen.

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.

Automatsko ispitivanje funkcionalnosti

Izgrađena web usluga mora biti funkcionalna! Preporučeni pristup je korištenje alata za **automatsko ispitivanje ugrađenog u radni okvir koji koristite** (proučite samostalno dokumentaciju). Ugrađeni alati za ispitivanje najčešće koriste neku inačicu xUnit ispitivanja gdje ispravnost ispitujete s **assert*** funkcijama. Alternativno, **ali NE i preporučeno**, moguće je koristiti Selenium (<http://seleniumhq.org/>) ili curl (<http://curl.haxx.se/>).

Za SVAKI izloženi URL napisati barem jedan ispitni slučaj koji omogućava provjeru ispravnosti usluge nakon mijenjanja programskog koda i/ili nakon postavljanja usluge na novi poslužitelj. Oblik pojedinog ispitnog slučaja jest proizvoljan. Dobro bi bilo napraviti nekoliko zasebnih grupa ispitnih slučajeva koje demonstriraju (ispituju) funkcionalnost nekog segmenta web usluge, npr. stvaranje, dohvaćanje, brisanje i pretraživanje objekata.

Napomena: **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 labosima?

1. izgraditi, pokrenuti i opisati kako radi web usluga i, ako bude potrebe, uvesti manje modifikacije u programski kod
2. pokrenuti automatske ispitne slučajeve i pokazati da je izgrađena usluga funkcionalna

Predaja vježbe

Ostvarenu vježbu demonstrirate usmeno na zadanim terminima prema kalendaru. Budući da vježbu predajete na **vlastitim prijenosnim računalima**, savjetujemo pripremu programskog koda i ostvarenja prije predaje same vježbe (npr. nakon što kod kuće pripremite vježbu za demonstraciju, postavite računalo u *stand-by* ili *hibernate*). Studenti koji nisu u mogućnosti donijeti vlastito prijenosno računalo mogu pokušati posuditi računalo. Ako ni to nije moguće, molim vas javite se asistentima na službeni email.