

Upute za 1. laboratorijsku vježbu

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

21. listopada 2015.

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 Flickr²). Bitno je istaknuti da naglasak ove laboratorijske vježbe nije na količini mogućnosti izgrađene web usluge! Jedine obavezne komponente su (1) REST sučelje, (2) autentifikacija korisnika, (3) automatsko ispitivanje i (4) generiranje dnevnika. Nije potrebno ostvariti bogato korisničko sučelje, 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, usluga mora omogućiti dohvat svih fotografija određenog korisnika (npr. `/users/1/photos`). Za svako izloženo sredstvo ostvariti **sve** CRUD metode! Programsko sučelje mora biti dostupno kao poddirektorij (npr. `http://localhost/api`). Navedeni korijenski direktorij mora sadržavati **indeks** s funkcionalnim popisom svih URL-ova te svih pripadajućih CRUD metoda! Indeks mora omogućiti lagano upravljanje izgrađenom web uslugom (pregled/izmjena), a izgled je proizvoljan.

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!

¹<http://twitter.com>

²<http://flickr.com>

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 npr. mijenjanja programskog koda i 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, automatiziranje 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 ...
```

Dnevnik korištenja

Dnevnik korištenja (engl. *log*) jednostavna je tekstualna datoteka u koju vaša usluga bilježi pristup svakom URL-u. Web-usluga bilježi u tekstualni dnevnik zapise oblika: `url browser`, gdje je `url` dio URL-a koji opisuje

sredstvo, a **browser** identifikator web-preglednika koji je pristupio sredstvu. Primjer dnevnika prikazuje sljedeći isječak tekstualne datoteke:

```
/photo/23  firefox
/photo/23  chrome
/photo/24  chrome
/user/1     ie9
/user/2     ie9
/photo/67   chrome
```

Za sada je potrebno samo generirati dnevnik. Izgrađeni dnevnik će se koristiti u 2. laboratorijskoj vježbi.

Što je potrebno znati na labosima?

1. izgraditi, pokrenuti i opisati kako radi izgrađena web usluga i, ako bude potrebe, uvesti manje modifikacije u programski kod
2. pokrenuti automatske ispitne slučajeve i pokazati da je izgrađena usluga funkcionalna
3. po završetku automatskog ispitivanja prikazati generirane dnevnik korištenja

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.

Prije samog usmenog ispitivanja, laboratorijske vježbe na predmetu RZNU potrebno je postaviti na sustav SPRUT koji omogućuje automatsku evaluaciju programskih rješenja. Sustav za autentikaciju koristi FERWeb, tj. korisničko ime i lozinka jednake su onima na FERWebu. Sustav koristi samopotpisani SSL certifikat tako da možete očekivati upozorenje preglednika da stranica nije sigurna, ali na to se u ovom slučaju nemojte obazirati. Ako se ne možete prijaviti na sustav koristeći svoje FERWeb podatke, što prije se javite na e-mail listu predmeta. Rokovi za postavljanje vježbi na SPRUT te bodovanje po terminima bit će objavljeni na stranici predmeta. **Samo oni studenti koji postavе vježbu na SPRUT u zadnom roku će biti usmeno ispitani!**

Trenutno, sustav SPRUT služi samo za **arhiviranje** vaših rješenja (bez automatske evaluacije) te provjeru **plagijata**. Potrebno je sve datoteke pohraniti u **jednu ZIP arhivu** i ta arhiva se predaje na sustav. U ZIP arhivi **ne** smije biti nikakvih direktorija, tj. datoteke se trebaju nalaziti **izravno u korijenu arhive**. U arhivu se stavljaju **isključivo** datoteke s kodom rješenja, bez datoteka s metapodacima Eclipse/VS projekata i slično.

Vježbu postavljate na SPRUT klikom na link **Predaja rjesenja** za odgovarajuću vježbu. Nakon toga odaberete pripremljenu ZIP arhivu te odaberete **bilo koji programski jezik**. Odabrani programski jezik ne mora biti nužno jezik u kojem ste ostvarili vježbu (u ovom trenutku nije bitan odabir jezika jer sustav ne evaluira automatski vaše vježbe). Uspješno postavljanje rezul-

tira porukom **Rjesenje uspjesno predano** te porukom o integracijskim testovima koju možete ignorirati. Opisani slijed aktivnosti je dovoljan za postavljanje vježbe na SPRUT.

Napomena: Nemojte varati, tj. prikazivati tuđe rješenje ili dio tuđeg rješenja kao svoje. Nastavnici i asistenti će se maksimalno potruditi i automatiziranim i ručnim metodama otkriti i kazniti sve oblike varanja kako bi svi imali jednake uvjete za obavljanje ovih laboratorijskih vježbi. Dozvoljeno je s kolegama raspravljati o idejama i algoritmima koje ćete implementirati, ali bilo kakav oblik dijeljenja koda ili "pseudokoda" je zabranjen. Nemojte ignorirati ovu napomenu!