

Unierzitet u Sarajevu  
Elektrotehnički fakultet  
Računarstvo i informatika

# **Pregled funkcionalnosti modula On-Line Kompajler**

Predmet: Praktikum - Napredne web tehnologije

Predmetni nastavnik:

Doc. dr Vensada Okanović

Studenti:

Ivona Ivković

Lejla Nukić

Amar Panjeta

Džana Feratović

Sarajevo, mart 2017.

# 1. Opis modula

Modul On-Line Kompajler je jednostavni modul koji omogućava kompajliranje i izvršavanje ispravnih C++ kodova. Korisnici šalju kodove putem odgovarajućeg interfejsa nakon čega modul odgovara sa porukom o uspješnosti i rezultatima izvršavanja. Ovaj modul se koristi od strane modula Zadaci.

## 2. Osnovne funkcionalnosti

Osnovne funkcionalnosti modula su:

- Kompajliranje i pokretanje C++ kodova

### 2.1 Kompajliranje i pokretanje C++ kodova

Ovo je osnovna i jedina funkcionalnost koju ovaj modul realizira. Putem odgovarajućeg zahtjeva se šalju podaci o kodu, testovima i korisniku koji šalje kod. Potom se kod kompajlira i pokreće, uz pokretanje testova ukoliko su isti poslani. Nakon izvršavanja se dobija povratna informacija o tome da li je kod ispravan i o rezultatima njegovog izvršenja.

## 3. Modeli podataka

S obzirom da ovaj modul nema ulogu čuvanja podataka i ne vrši nikakvu interakciju sa bazom za njega nije moguće napraviti ER dijagram. Ovdje će biti prikazani modeli podataka koji se šalju prilikom poziva servisa i koji se dobijaju kao odgovor od strane servisa, jer je njihov format bitan za ispravno korištenje istog.

### 3.1 Klasa CodeClass

Klasa CodeClass služi za definisanje podataka koji se šalju servisu. Izgled klase je prikazan ispod:

```
public class CodeClass {  
    public String code;  
    public String[] tests;  
    public String username;  
}
```

*Izgled klase CodeClass*

JSON objekat koji se generiše za ovu klasu ima sljedeći oblik:

```
{  
    "code":"some code here",  
    "tests":["test1","result1","test2","result2",...],  
    "username":"username"  
}
```

## 3.2 ResultClass objekat

Klasa CodeClass služi za definisanje podataka koji se dobijaju kao odgovor od servisa. Izgled klase je prikazan ispod:

```
public class ResultClass {  
    public String status;  
    public String result;  
  
    public ResultClass(String status,String result){  
        this.status=status;  
        this.result=result;  
    }  
}
```

*Izgled ResultClass klase*

JSON objekat koji se generiše za ovu klasu ima sljedeći oblik:

```
{  
    "status":"working| not working",  
    "result":"code output"  
}
```

## 4. Servis i njegova upotreba

Ovaj servis se koristi tako što se vrši POST request sa podacima o kodu koji je potrebno izvršiti. Podaci se šalju u formatu opisanom u prethodnom poglavlju. Prikaz korištenja servisa je dat na slici ispod.

```
amar@amar-HP-ProBook-450-G0: ~  
amar@amar-HP-ProBook-450-G0:~$ curl -H "Content-Type: application/json" -X POST  
-d @code.json localhost:8080/compiler/run  
{"status":"working","result":"666\n"}amar@amar-HP-ProBook-450-G0:~$
```

*Poziv servisa za izvršavanje koda*

Podaci koji su poslani servisom se nalaze u fajlu code.json . Sadržaj fajla je prikazan na sljedećoj slici.

```
code.json x  
{  
    "code": " #include <iostream> \n using namespace std;int main()  
{cout<<666<<endl;return 0;}",  
    "tests": [],  
    "username": "amar"  
}
```

*Sadržaj fajla code.json*