

Grupa1 - Cupid

# LOVID-20 API izvještaj

iz predmeta Objektno orijentisana analiza i  
dizajn

Zejd Čelenka, Adna Halilović i Ajla  
Džaferbegović  
6/7/2020

## **OPIS API-ja**

API naše aplikacije omogućava rad sa bazom podataka koja čuva podatke o korisničkim recenzijama vezane za funkcionalnosti aplikacije. API je iskorišten u web aplikaciji tako što omogućava eksternim korisnicima uvid u korisničke tekstualne ocjene o aplikaciji i vraća ih u JSON formatu. Podaci su prikazani u random poretku, preko implementiranog algoritma u određenom kontroleru.

## **RUTE I KONKRETNA IMPLEMENTACIJA**

Rute koje omogućavaju dohvaćanje konkretnih podataka:

- `api/Recenzija` (dohvata sve recenzije iz baze)
- `api/Recenzija/id` (dohvata recenziju pomoću random algoritma gdje je varijabla `x = id`)

```
// GET: api/Recenzija
[HttpGet]
0 references
public async Task<ActionResult<IEnumerable<Recenzija>>> GetRecenzija()
{
    return await _context.Recenzija.ToListAsync();
}
```

```
// GET: api/Recenzija/5
[HttpGet("{id}")]
0 references
public async Task<ActionResult<IEnumerable<Recenzija>>> GetRecenzija(int id)
{
    var list = await _context.Recenzija.ToListAsync();
    RNGCryptoServiceProvider provider = new RNGCryptoServiceProvider();
    int n = list.Count;
    while (n > 1)
    {
        byte[] box = new byte[1];
        do provider.GetBytes(box);
        while (!(box[0] < n * (Byte.MaxValue / n)));
        int k = (box[0] % n);
        n--;
        Recenzija value = list[k];
        list[k] = list[n];
        list[n] = value;
    }

    return list.Take(id).ToList();
}
```

## **UPOTREBA API-JA U PROJEKTU**

Upotreba API-ja u aplikaciji je sljedeća; u RecenzijaController-u preko GET metode našeg API dobavljamo recenzije iz baze podataka, koje će se ta ista metoda vratiti u listu, a ta ista randomizirana lista, odnosno zasebni elementi liste, će biti prikazani na web-servisu preko Reviews pogleda korisnicima aplikacije.

## **VANJSKI SERVISI**

Nisu korišteni vanjski servisi za realizaciju ovog zadatka.