






# Dokumentacija za Blog

Moguca je **registracija**, preko koje se kasnije proverava da li korisnik ima dozvoljen **pristup odredjenim Use-Caseovima**. Moguce je dohvatiti korisnika, kao i **modifikovati User Access**, tj privilegije koje taj User ima. Na nasem blogu moguce je objaviti post, komentarisati taj post, odgovoriti na neki od komentara, lajkovati post, lajkovati i komentar. Moguce je dohvaćanje i pretraga Postova, Komentara, Lajkova na komentarima, kao i **filtriranje**. Postoji i **paginacija**, kao i **Swagger** specifikacija. Korisceno je autorizovanje preko **JWT tokena**. Za end-pointe koji menjaju stanje sistema, obezbedjena je **validacija**. Koriscen je **DataBase First pristup**.

- ▼ ASP Blog
  - ▼  Users
    - POST Register user
    - GET Get users
    - GET Get user likes
    - PUT Modify user access
  - ▼  Posts
    - POST Create a post
    - GET Get post
  - ▼  Logs
    - GET Get Use Cases Log
    - GET Get Errors Log
    - GET Get Errors Log ById
  - ▼  Likes
    - POST Like a post
    - POST Like a post Copy
  - ▼  Comments
    - POST Comment a post
    - GET Get comment
    - GET Get comments
    - POST Create token

**EndPointi:**

## Create Token :

[POST]{path}/api/auth – Putanja za kreiranje tokena, u body-u je obavezno proslediti email I password, kroz JSON format.

Primer validnog body-a:

```
{  
  "email": "nemanja@gmail.com",  
  "password": "Asd123asd"  
}
```

## Users:

[POST]{path}/api/users/ - Endpoint za registraciju usera. Nije obavezan JWT token, obzirom da user jos uvek nije ni registrovan.

Primer validnog body-a:

```
{  
  "email": "nemanja@gmail.com",  
  "username": "Nemanja",  
  "firstName": "Nemanja",  
  "lastName": "Marjanovic",  
  "password": "Asd123asd"  
}
```

[GET]{path}/api/users?(filteri) – Endpoint za pretragu Usera. Moguce je proslediti filtere kroz QueryString.

## Posts:

## Logs:

**Likes:**

**Comments:**