



Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

RAZVOJ WEB-APLIKACIJA

Predavanje 06 - Dodatak: Slanje mailova

31. listopada 2025.

Sastavio: Zvonimir Bujanović



Slati mail iz Pythona možemo pomoću ekstenzija poput **smtplib** i **email**.

- U Flasku je još jednostavnije koristiti ekstenziju **Flask-Mail**.
- Da bismo slali mailove, potrebno je imati pristup tzv. SMTP serveru.
 - Takav server (npr. **Postfix**) možete instalirati i na lokalno računalo.
 - U svrhu učenja može poslužiti i obični mail-account na **gmail-u**.
 - Za potrebe web-aplikacija, najbolje je ipak koristiti servis poput **Brevo** - besplatna registracija i korištenje za mali broj mailova.

U nastavku opisujemo kako koristiti Brevo SMTP:

- Otvorite novi Brevo account.
- U Brevo Settings->SMTP & API kliknite na "Generate a new SMTP key".
- Kopirajte dobiveni "SMTP key value" i "SMTP Login" s iste stranice.

app.config

```
1 MAIL_SERVER = 'smtp-relay.brevo.com';
2 MAIL_PORT = 587;
3 MAIL_USE_TLS = True;
4 MAIL_USE_SSL = False;
5
6 # "SMTP Login" iz Brevo (Settings->SMTP & API)
7 MAIL_USERNAME = 'neki-broj@smtp-brevo.com';
8
9 # "SMTP key value" iz Brevo (Settings->SMTP & API)
10 MAIL_PASSWORD = 'dugi-niz-znakova';
11
12 # Umjesto email@example.com: adresa iz Brevo Settings->Senders, domains, IPs.
13 MAIL_DEFAULT_SENDER = ('Ime pošiljatelja', 'email@example.com');
```

Korištenje flask-mail

```
1 from flask import Flask;
2 from flask_mail import Mail, Message;
3
4 # ----- Konfiguracija
5 app = Flask( __name__ );
6 app.config.from_pyfile( 'app.config' );
7
8 mail = Mail( app );
9
10 # ----- Rute
11 @app.route( '/send-mail' )
12 def send_mail():
13     msg = Message(
14         subject = 'Poruka poslana iz Flask aplikacije!',
15         recipients = ['primatelj@example.com'],
16         body = 'Ovo je probni email poslan iz Flaska!',
17     );
18
19     mail.send( msg );
20     return 'Mail je uspješno poslan. Provjeri inbox (i spam...)';
```

Registracija korisnika u web-aplikaciji

Kod registracije korisnika u web-aplikacijama običaj je unijeti e-mail.

Uvijek je potrebno napraviti provjeru pripada li zaista taj e-mail korisniku koji se registrira.

- 1 Kada korisnik stvori novi account, u tablicu `users` u bazi spremimo: `username`, `password_hash`, `email`, `regseq`, `has_registered`.
 - Za `regseq` generiramo random niz od npr. 64 znaka (za svakog korisnika drugi).
 - U `has_registered` upišemo 0.
- 2 Korisniku prilikom stvaranja accounta pošaljemo e-mail u kojem je link na rutu poput `/users/register?regseq=XYZ`, gdje umjesto `XYZ` stavimo gornji random niz spremljen u bazu. (Za development server ruti dodajte prefiks `http://127.0.0.1:5000/`)
- 3 Kada korisnik otvori mail i klikne na link, na toj ruti iz `request.args.get('regseq')` dohvatimo niz, te ga potražimo u bazi u tablici `users`. Korisniku kojem taj niz pripada stavimo `has_registered` na 1. Time je korisnik potvrdio svoju e-mail adresu.
- 4 Korisniku ne dozvoljavamo ulogiravanje sve dok u `has_registered` ne piše 1.