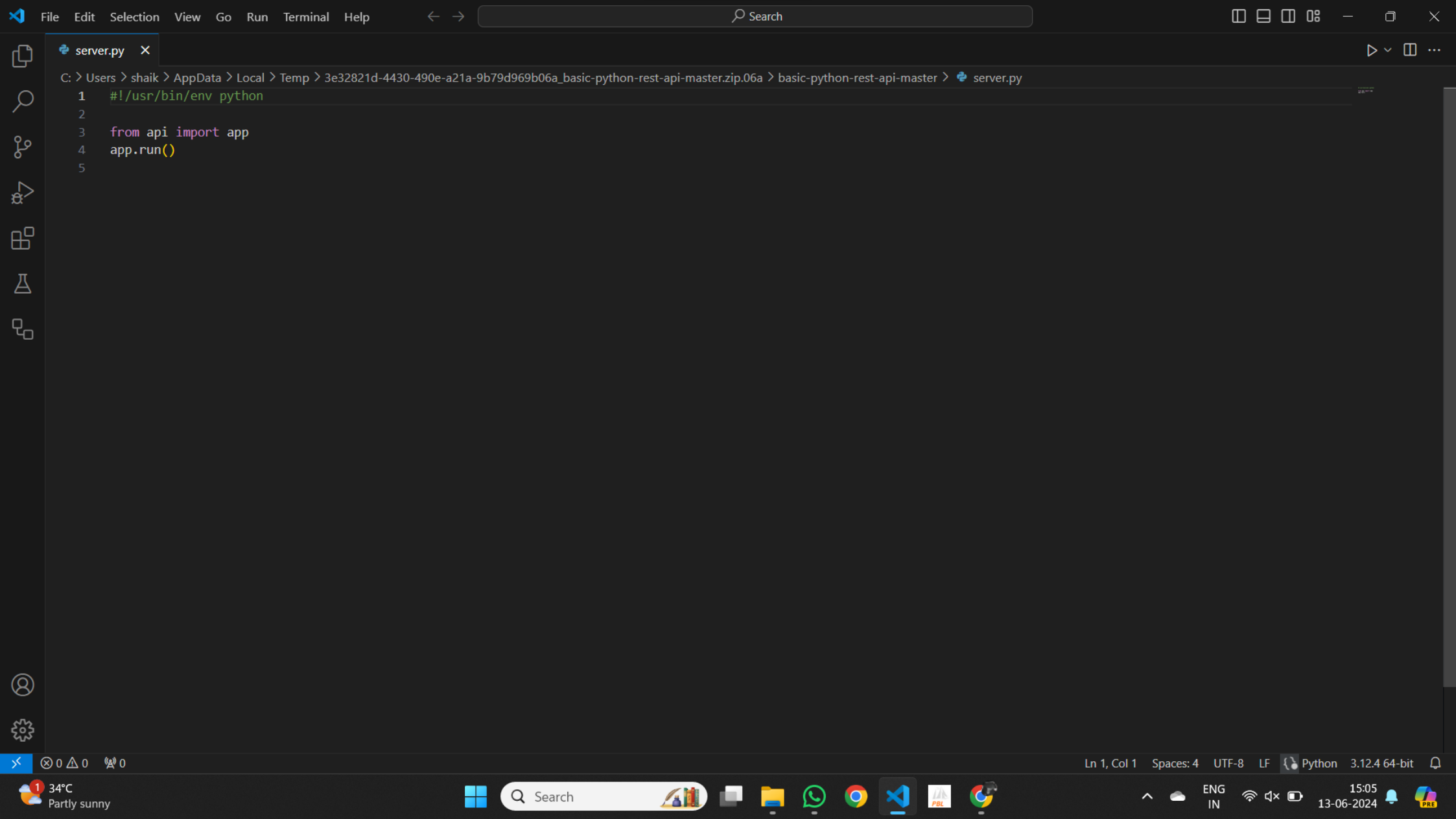


```

1 package com.sopromadze.blogapi;
2
3 import com.sopromadze.blogapi.security.JwtAuthenticationFilter;
4 import org.modelmapper.ModelMapper;
5 import org.springframework.boot.SpringApplication;
6 import org.springframework.boot.autoconfigure.SpringBootApplication;
7 import org.springframework.boot.autoconfigure.domain.EntityScan;
8 import org.springframework.context.annotation.Bean;
9 import org.springframework.data.convert.Jsr310Converters;
10
11 import javax.annotation.PostConstruct;
12 import java.util.TimeZone;
13
14 @SpringBootApplication
15 @EntityScan(basePackageClasses = { BlogApiApplication.class, Jsr310Converters.class })
16
17 public class BlogApiApplication {
18
19     public static void main(String[] args) {
20         SpringApplication.run(BlogApiApplication.class, args);
21     }
22
23     @PostConstruct
24     void init() {
25         TimeZone.setDefault(TimeZone.getTimeZone("UTC"));
26     }
27
28     @Bean
29     public JwtAuthenticationFilter jwtAuthenticationFilter() {
30         return new JwtAuthenticationFilter();
31     }
32
33     @Bean
34     public ModelMapper modelMapper() {
35         return new ModelMapper();
36     }
37

```



```
C: > Users > shaik > AppData > Local > Temp > bbac3e5f-e28e-4a78-996c-8b0b6cb01d05_basic-python-rest-api-master.zip.d05 > basic-python-rest-api-master > api > _init_.py
1  import logging
2  from logging.handlers import RotatingFileHandler
3  from flask import Flask
4
5  app = Flask(__name__)
6  app.config.from_object('api.config')
7  #app.config.from_envvar('API_CONFIG')
8
9  # ref: https://gist.github.com/ibeex/3257877
10 formatter = logging.Formatter(
11     "[%asctime)s] %(pathname)s:%(lineno)d} %(levelname)s - %(message)s"
12 )
13 handler = RotatingFileHandler(app.config['LOG_FILENAME'],
14                               maxBytes=10000000,
15                               backupCount=5)
16 handler.setLevel(logging.DEBUG)
17 handler.setFormatter(formatter)
18 app.logger.addHandler(handler)
19
20 # Output the access logs to the same file
21 log = logging.getLogger('werkzeug')
22 log.setLevel(logging.DEBUG)
23 log.addHandler(handler)
24
25 @app.errorhandler(404)
26 def page_not_found(e):
27     return "404 not found", 404
28
29 # Import the routes from all controllers
30 from api.examples import get_example
```

config.py

```
C: > Users > shaik > AppData > Local > Temp > c296fb5b-08d8-4b44-9a37-4b1524f7f140_basic-python-rest-api-master.zip.140 > basic-python-rest-api-master > api > config.py
1  """Default configuration settings"""
2
3  DEBUG = True
4  TESTING = False
5  LOGGER_NAME = 'api-server'
6  LOG_FILENAME = 'api-server.log'
7
```

README AGPL-3.0 license Security

Build Status

# Spring Boot, MySQL, Spring Security, JWT, JPA, Rest API

Build Restful CRUD API for a blog using Spring Boot, Mysql, JPA and Hibernate.

## Steps to Setup

### 1. Clone the application

```
git clone https://github.com/coma123/Spring-Boot-Blog-REST-API.git
```

### 2. Create Mysql database

```
create database blogapi
```

- run `src/main/resources/blogapi.sql`

### 3. Change mysql username and password as per your installation

- open `src/main/resources/application.properties`
- change `spring.datasource.username` and `spring.datasource.password` as per your mysql installation

### 4. Run the app using maven

```
mvn spring-boot:run
```

## Suggested workflows

Based on your tech stack



Docker image

Configure

Build a Docker image to deploy, run, or push to a registry.



Publish Docker Container

Configure

Build, test and push Docker image to GitHub Packages.

[More workflows](#)

[Dismiss suggestions](#)

README AGPL-3.0 license Security

### Sign Up -> /api/auth/signup

```
{
  "firstName": "Leanne",
  "lastName": "Graham",
  "username": "leanne",
  "password": "password",
  "email": "leanne.graham@gmail.com"
}
```

### Log In -> /api/auth/signin

```
{
  "usernameOrEmail": "leanne",
  "password": "password"
}
```

### Create User -> /api/users

```
{
  "firstName": "Ervin",
  "lastName": "Howell",
  "username": "ervin",
  "password": "password",
  "email": "ervin.howell@gmail.com",
  "address": {
    "street": "Victor Plains",
    "suite": "Suite 879",
    "city": "Wisokyburgh",
    "zipcode": "90566-7771",
  }
}
```