

REST API Implementation in Python

CS403/534 - Distributed Systems
Assignment #1 for Spring 2019

E. Savaş
Computer Science & Engineering
Sabancı University
İstanbul

March 6, 2019

Abstract

You are required to develop a ticket automation system in Python.
The further details are given in the subsequent sections of this document.

1 Description

In this assignment, you are asked to develop a Python program for server side that enables the client programs to invoke remote API endpoints. Suppose there is an airline company that launches flights and a server manages ticket operations. Using the API, admin of the system can add new flights to the system and view the sales report, customers can check available flights or buy tickets (and do check-in) using the API. You should test your API by implementing the client side. However, only the REST API will be graded.

2 API Description

Here, we give API endpoints and the associated operations.

2.1 /flights

2.1.1 GET

Get the list of flights with flight information such as destination, origin, and date.

JSON(Request)	
JSON(Response)	[{dest,from,date,flight_id}]
HTTP Status Code	200

2.1.2 PUT

Create a flight by providing flight information

JSON(Request)	{dest, from, date}
JSON(Response)	{flight_id}
HTTP Status Code	201

2.2 /flights/flight_id

2.2.1 GET

Get flight information with a given flight ID

JSON(Request)	
JSON(Response)	{dest, from, date, flight_id}
HTTP Status Code	if flight_id exists: 200 else: 404

2.2.2 DELETE

Cancel a flight with a given flight ID

JSON(Request)	
JSON(Response)	
HTTP Status Code	if flight_id exists: 200 else: 404

2.3 /ticket

2.3.1 PUT

Buy ticket for the flight with the given flight ID.

JSON(Request)	{flight_id}
JSON(Response)	if flight_id exists and seats are available: {PNR}
HTTP Status Code	if flight_id exists and seats are available: 200, else if flight_id exists but no seats are available: 409 else if flight_id does not exist: 404

2.3.2 GET

View the ticket with a given PNR. Note that if the passenger has not checked in, the seat number will be 0.

JSON(Request)	{PNR}
JSON(Response)	if PNR exists: {dest, from, date, flight_id, seat_number}
HTTP Status Code	if PNR exists: 200, else if PNR does not exist: 404

View all tickets

JSON(Request)	
JSON(Response)	[{ dest , from , date , flight_id }]
HTTP Status Code	200

2.3.3 POST

Check in for a flight with the given PNR

JSON(Request)	{ PNR , seat_number }
JSON(Response)	
HTTP Status Code	if PNR exists and seat_number is available: 200, else if PNR exists but seat_number is not available: 409 else if PNR does not exist: 404

2.3.4 DELETE

Cancel the ticket with a given PNR

JSON(Request)	{ PNR }
HTTP Status Code	if PNR exists: 200, else: 404

3 Bonus Part (20 Pts)

Bonus part is to add authentication to the system so that the following requirements are met:

- Only ADMIN can add new flights, view all the tickets
- For any non-authorized behavior, return HTTP Status Code 401

To check whether an API user is ADMIN, add **username** and **password** keys to all JSON attachments of HTTP requests specified in Section 2. Note that for security reason, you need to send the hash of the password, e.g., `SHA256(password)`.

For instance, the second method, which returns all purchased tickets in Section 2.3.2, should contain a JSON attachment in the request such as `{username,SHA256(password)}`.

4 Client Examples

```
URL="localhost:5000"
```

```
response = requests.get((URL+"/flight"))
content = response.json()
interesting_flight=content[4]
interesting_flight_dest=interesting_flight['dest']
```

```

interesting_flight_ID=interesting_flight['flight_id']

response = requests.put((URL+"/ticket"), json={'flight_id': interesting_flight_ID})
if response.status_code == 200:
    ...content = response.json()
    ...PNR=content['PNR']

response = requests.post((URL+"/ticket"), json={'PNR': PNR,'seat_number'=10})
if response.status_code == 409:
    ...response = requests.post((URL+"/ticket"), json={'PNR': PNR,'seat_number'=13})

response = requests.delete((URL+"/ticket"), json={'PNR': PNR})

```

5 Notes

- The deadline is March 13, 2019 @11:55pm.
- You can work in groups of two.
- Submit your assignments through SUCourse and name ALL files using the format “CS403_Assign1_SUusername1_SUusername2.zip” or “CS534_Assign1_SUusername1_SUusername2.zip” etc.
- One submission per group is sufficient.
- For assistance write to Tolun Tosun (toluntosun@sabanciuniv.edu), Atıl Utku Ay (atilutkuay@sabanciuniv.edu) or Erkan Savaş (erkays@sabanciuniv.edu).
- **Office hours:** Monday 18:40-20:30 @FENS 2014 (Tolun Tosun); Friday 8:40-10:30 @FENS 2014 (Atıl Utku Ay); Wednesday 14:40-16:30 @FENS 1098 (Erkan Savaş)