RESTful API Implementation in Python

CS403/534 - Distributed Systems
Assignment #3 for Spring 2020

Ş. S. Mağara, E. Savaş
Computer Science & Engineering
Sabancı University
İstanbul

March 5, 2020

Abstract

You are required to develop a ticket automation system in Python using RESTful API. 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 a movie theater with a few auditorium and a server manages movie schedule and ticket operations. Using the API, admin of the system can add new movie sessions to the system and view the sales report, customers can check available movies or buy tickets using the API. You can test your API by the client side code provided to you.

2 API Description

Here, we give API endpoints and the associated operations.

2.1 /movies

2.1.1 `GET`

Get the list of movies with their respective information such as movie name, date, and screen number.

JSON(Request)	
JSON(Response)	[{`name`, `date`, `time`, `screen_no`}]
HTTP Status Code	200

2.1.2 POST

Create movie session by providing movie information

JSON(Request)	{`name`, `date`, `time`}
JSON(Response)	{`screen_no`}
HTTP Status Code	if `screen_no exist:` exists: 201 else: 404

2.2 /movies/<name>/<date>

2.2.1 GET

Get movie information with a given name and date

JSON(Request)	
JSON(Response)	[{`name`, `date`, `time`, `screen_no`}]
HTTP Status Code	if `date and name` exists: 200 else: 404

2.2.2 DELETE

Cancel a movie screening with a given date and name

JSON(Request)	
JSON(Response)	
HTTP Status Code	if `date and name` exists: 200 else: 404

2.3 /ticket

2.3.1 POST

Buy a ticket for the movie with the given name date and time.

JSON(Request)	{`name`, `date`, `time`, `screen_no`}
JSON(Response)	{`reservation_no`}
HTTP Status Code	if `session` exists and seats are available: 201, else if `session` exists but no seats are available: 409 else if `session` does not exist: 404

2.3.2 GET

View the ticket with a given `reservation_no`.

JSON(Request)	{`reservation_no`}
JSON(Response)	if `reservation_no` exists: {`name`, `date`, `time`, `screen_no`, `seat_no`}
HTTP Status Code	if `reservation_no` exists: 200, else if `reservation_no` does not exist: 404

View all tickets

JSON(Request)	
JSON(Response)	[{`name`, `date`, `time`, `screen_no`, `seat_no`, `reservation_no`}]
HTTP Status Code	200

2.3.3 PUT

Change the seat of the reservation with the given `reservation_no` and `seat_no`

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

2.3.4 DELETE

Cancel the ticket with a given `reservation_no`

JSON(Request)	{`reservation_no`}
HTTP Status Code	if `reservation_no` 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 movie screenings, 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+"/movies"))
content = response.json()
interesting_movies_date=interesting_movies['date']
```

```
interesting_movies_name=interesting_movies['name']
interesting_movies_time=interesting_movies['time']

response = requests.post((URL+"/movies"), json={'name':  interesting_movies_name,
'date':  interesting_movies_date, 'time':  interesting_movies_time})
if response.status_code == 200:
...content = response.json()
...reservation_num=content['reservation_no']

response = requests.put((URL+"/ticket"), json={'reservation_no':  reservation_num,
'seat_no':  10})
if response.status_code == 409:
...response = requests.put((URL+"/ticket"), json={'reservation_no':
reservation_num,'seat_no'=13})

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

5 Assumptions

You can make the following assumptions for sake of simplicity

- All movies are two hours.
- Schedule will be for only thirty days.
- Movies are scheduled for 7/24.

6 Notes

- The deadline is March 9, 2020 @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 Şeyma Selcan Magara (sselcan@sabanciuniv.edu), Tolun Tosun (toluntosun@sabanciuniv.edu) or Erkay Savaş (erkays@sabanciuniv.edu).