

Appointment API

Backend to a Modular Scheduling Site
Nicholas Morin

ABSTRACT

My senior project was to design and develop an API for a modular appointment system. I brainstormed case stories and then developed endpoint methods based on those stories. Each method was thoroughly documented and tested using a third party software called Postman.

Besides endpoints, I also implemented unit tests for each endpoint that would be run on each commit to GitHub via a GitHub actions script. In total there were 14 endpoints and 18 unit tests developed.

INTRODUCTION

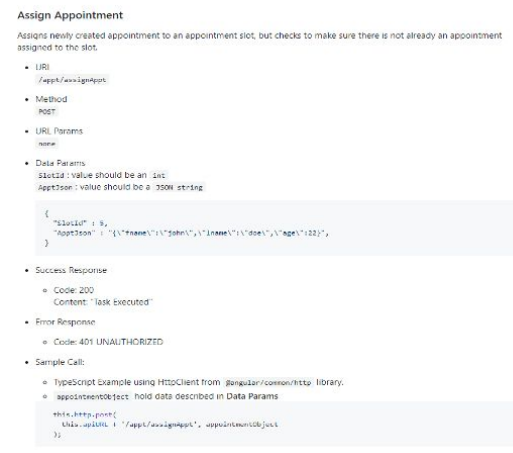
In the summer of 2019, I was hired as an intern for a small company called Community Software Group based out of Worcester, Massachusetts. This company writes software for social service agencies and government offices in the state of Massachusetts.

Prior to returning to school for my senior year I was asked to develop an API for a modular appointment system. This appointment system needed to cater towards all the agencies and be able to be dynamic enough for different configurations. This was a fairly new project so there was not a frontend or any preexisting code for me to begin with.

DATA AND ANALYSIS

The project was written in C# using .NET Core and the Entity Framework with Microsoft SQL Server being used as my database. There were two tables in the database: OfficeLocations and AppointmentSlots. OfficeLocations held information regarding agencies' offices and their locations in reference to the user's location. AppointmentSlots was the main table used for the API and each tuple acted as a slot where an appointment could be booked.

Each slot or tuple had 5 columns of data associated with it. There was a SlotId, Date and Time of the appointment, a JSON string containing appointment information and a timestamp that would be updated every time data was changed.



```
Assign Appointment
Assigns newly created appointment to an appointment slot, but checks to make sure there is not already an appointment assigned to the slot.

• URI
  /api/assignapp

• Method
  POST

• URL Params
  none

• Data Params
  slotId : value should be an int
  appointment : value should be a json string

  {
    "slotId": 9,
    "appointment": "{ \"name\": \"John\", \"time\": \"12:00\" }"
  }

• Success Response
  • Code: 200
  Content: "task executed"

• Error Response
  • Code: 401 UNAUTHORIZED

• Sample Call
  • TypeScript example using HttpClient from @angular/common/http library:
  • appointmentObject: hold data described in Data Params
  http.post('http://localhost:3000/api/assignapp', appointmentObject)
```

I used a third party testing software called Postman to test my endpoints once developed. This software is able to mimic a call to the endpoint by sending request data and returning a response. Once the endpoint was tested and I was satisfied with the outcome I would document the endpoint in a .md (markdown) file like the image above.

CONCLUSION

In conclusion, I was able to develop, test, and document all necessary endpoints. I was also able to write unit tests for each endpoint but due to lack of experience was not able to fully configure the unit test project to the API project. I gained significant experience in C#, .NET Core and the development process as a whole. Moving forward I hope to meet with another developer who can assist me in configuring the unit test project.

SUMMARY

At the end of this project, I developed fourteen endpoints that are tested and fully functional. I was also able to develop 18 unit tests with some endpoints having more than one test. Even though the unit tests are not fully configured, there is automated testing set up where upon pushing a commit to GitHub a script will execute and run the tests.

REFERENCES

- <https://www.communitysoftwaregroup.com/>