```
            status_code=status.HTTP_404_NOT_FOUND,
            detail="Event not found"
        )
```

In the preceding code block, we instruct the route function to first check whether the current user is the creator, otherwise raise an exception. Let's take a look at an example where another user attempts to delete another user's event:

```
$ curl -X 'DELETE' \
  'http://0.0.0.0:8080/event/6265a83fc823a3c912830074' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXV
CJ9.eyJ1c2VyIjoiZmFzdGFwaUBwYWNrdC5jb20iLCJleHBpcmVzIjoxNjUwOD
MzOTc2LjI2NzgzMX0.MMRT6pwEDBVHTU5C1a6MV8j9wCfWhqbza9NBpZz08xE'
```

An event not found is returned as the response:

```
{
  "detail": "Event not found"
}
```

However, the ideal owner can delete an event:

```
$ curl -X 'DELETE' \
  'http://0.0.0.0:8080/event/6265a83fc823a3c912830074' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC
J9.eyJ1c2VyIjoicmVhZGVyQHBhY2t0LmNvbSIsImV4cGlyZXMiOjE2NTA4MzQz
OTUuMDkzMDI3fQ.IKYHWQ2YO3rQc-KR8kyfoy_54MsEVE75WbRqoVbdoW0'
```

Here's the response:

```
{
  "message": "Event deleted successfully."
}
```

We have successfully secured our application and its routes. Let's wrap up this chapter by configuring a Cross-Origin Resource Sharing (CORS) middleware in the next section.

Configuring CORS

Cross-Origin Resource Sharing (CORS) serves as a rule that prevents unregistered clients access to a resource.

When our web API is consumed by a frontend application, the browser will not allow cross-origin HTTP requests. This means that resources can only be accessed from the exact origin as the API or origins permitted by the API.

FastAPI provides a CORS **middleware**, `CORSMiddleware`, that allows us to register domains which can access our API. The middleware takes an array of origins which will be permitted to access the resources on the server.

> **What is a middleware?**
> A middleware is a function that acts as an intermediary between an operation. In web APIs, a middleware serves as an mediator in a request-response operation.

For example, to allow only Packt to access our API, we define the URLs in the origin array:

```
origins = [
    "http://packtpub.com",
    "https://packtpub.com"
]
```

To allow requests from any client, the origins array will contain only one value, an asterisk *(*)*. The asterisk is a wildcard that tells our API to allow requests from anywhere.

In `main.py`, lets configure our application to accept requests from everywhere:

```
from fastapi.middleware.cors import CORSMiddleware


# register origins

origins = ["*"]

app.add_middleware(
```

```
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)
```

In the code block above, we started by importing the `CORSMiddleware` class from FastAPI. We registered the origins array and finally registered the middleware into the application using the `add_middleware` method.

> **More information**
>
> The FastAPI documentation has more details on CORS - `https://fastapi.tiangolo.com/tutorial/cors/`

We have successfully configured our application to allow requests from any origin on the world wide web.

Summary

In this chapter, we learned how to secure a FastAPI application with OAuth and JWT. We also learned what dependency injection is, how it is used in FastAPI applications, and how to protect routes from unauthorized users. We also added a CORS middleware to permit access to our API from any client. We made use of the knowledge from previous chapters.

In the next chapter, you will be introduced to testing your FastAPI application. You will learn what testing an application is, why you should test applications, and how to test a FastAPI application.

Part 3: Testing And Deploying FastAPI Applications

On completing this part, you will be able to write and execute tests and deploy FastAPI applications using the knowledge obtained from the chapters included.

This part comprises the following chapters:

- *Chapter 8*, *Testing FastAPI Applications*
- *Chapter 9*, *Deploying FastAPI Applications*