

TESODEV CASE API DOCUMENTATION

API Evaluation and Notification

DOĞUKAN CANERLER

Abstract

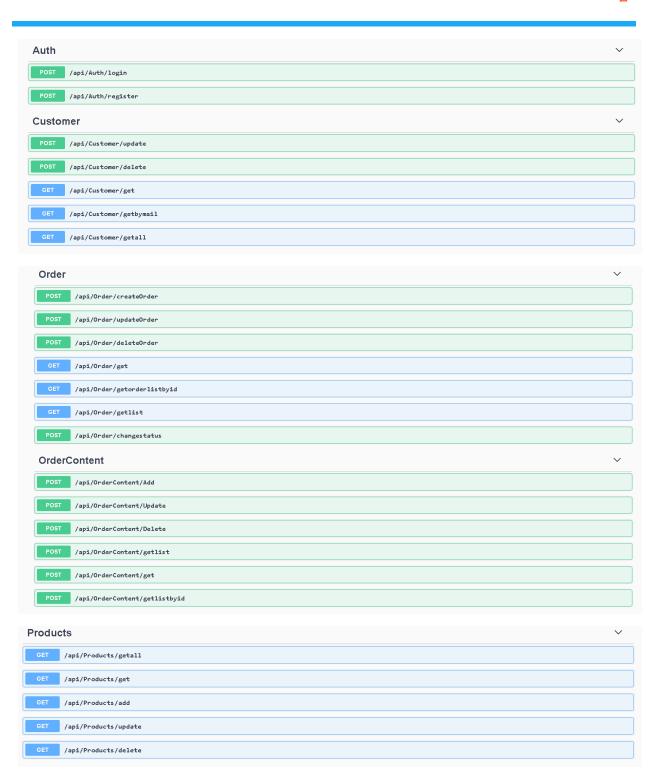
It is a RestFul WebAPI that I have prepared according to the document instructions I have received from you. In this API, .Net 5.0, C#, MSQQL, EntityFramework, Autofac, Jwt, as well as principles and architectural patterns such as Solid Principles and Layered Architecture has been made. Please evaluate me by considering that I am a Junior developer, my desire to improve myself, my effort and work. I think it is a very simple and useful WebAPI as in the directive. I hope you will be satisfied. I hope I will get a good result and I will have the opportunity to develop good projects with you.

Api's Mechanics

The images below are the Http requests that can be made for the API and the request schemes accordingly. However, there is one thing we should not forget that all requests in the API can be made by authorized and authenticated users in accordance with the JWT architecture. Therefore, you need to run the database script on GitHub as it is. If the API does not see the necessary authorizations on the user's token on the system, this process results in a 403 Http status code. You should not draw attention to the "nullable" part written in the Schemas section, if the API performs the necessary data validations. In order to be able to operate on the API, you need to log in and get a token and send it with a Bearer - token http request. Otherwise, you cannot do anything on the API.

API REQUESTS

Address	
POST /api/Address/update	
POST /api/Address/delete	
POST /api/Address/add	
POST /api/Address/GetAll	
POST /api/Address/GetAllByCustomer	
POST /api/Address/bycustomeraddress	



API SCHEMAS

Schemas

```
Address v {
                       integer($int32)
   customerId
                       string
                       nullable: true
   adressLine
                       string
                       nullable: true
   city
                       string
                       nullable: true
   country
                       string
                       nullable: true
   cityCode
                       integer($int32)
}
```

```
Customer ∨ {
   id
                       string
                       nullable: true
                       string
   na∎e
                       nullable: true
   e∎ail
                       string
                       nullable: true
   createdAt
                      string($date-time)
   updatedAt
                      string($date-time)
   passwordSalt
                      string($byte)
                       nullable: true
   passwordHash
                       string($byte)
                       nullable: true
}
```

```
Order 🗸 {
   id
                        string
                        nullable: true
   customerId
                       string
                       nullable: true
   address Id
                       string
                       nullable: true
   createdAt
                      string($date-time)
   updatedAt ....
                      string($date-time)
   status
                       string
                       nullable: true
}
```

```
OrderContent v {
   id
                      string
                      nullable: true
   orderId
                      string
                      nullable: true
   product Id
                     string
                     nullable: true
   quantity
                     number($double)
   createdAt
                     string($date-time)
   updatedAt
                     string($date-time)
}
```