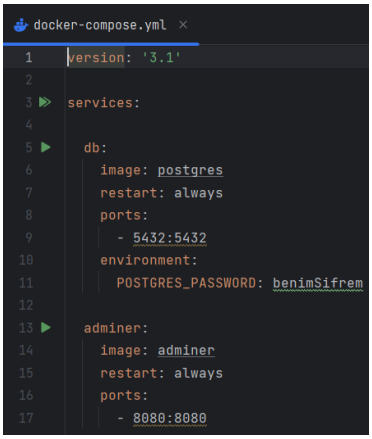
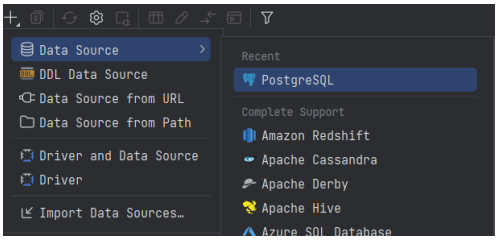
Proje Doküman

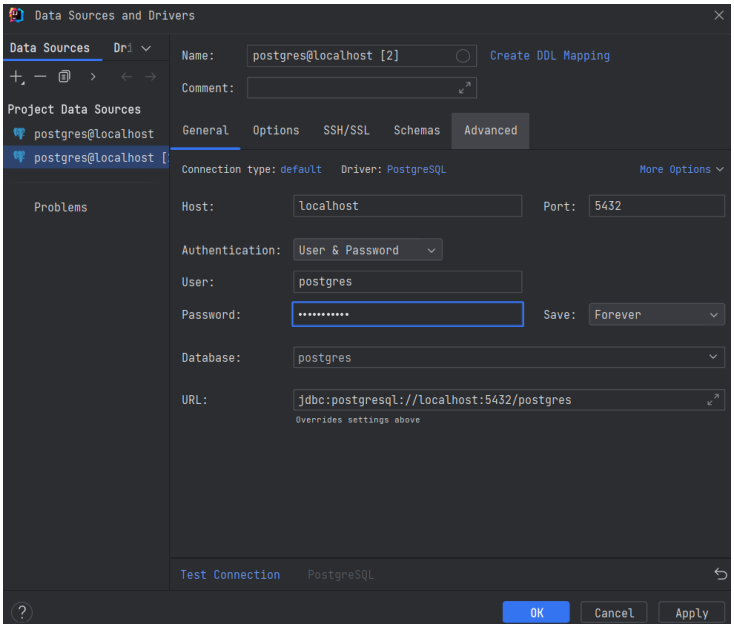
Projeyi ayağa kaldırmak



docker-compose.yml dosyasını services kısmından çalıştırın.



Data Source'den PostgreSQL kısmına tıklayın.



User ve Password girerek test edelim, apply ve OK diyelim.

Yapılanları localhost:8080'den de yapabilirsiniz.(adminer)

Projeyi ayağa kaldırılır ve tablolar görünür duruma gelir.

Proje dizayn:

- Controller
  - UserController
  - AddressController
- Entity
  - User
  - Address
- Enums
  - AddressType
- Repository
  - UserRepository
  - AddressRepository
- Service
  - UserService
  - AddressService
  - Impl
    - UserServiceImpl
    - AddressServiceImpl

Aşağıda yer alan API'ler:

User API

1. User Listele
2. İd'ye göre User Listele
3. User Sil
4. User Ekle
5. User Güncelle

Address API

1. Address Listele
2. İd'ye göre Address Listele
3. Address Sil
4. Address Ekle
5. Address Güncelle

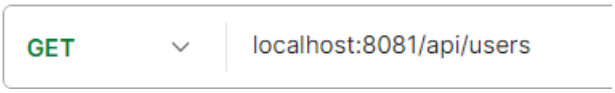
- **UserController API'si(/api/users)**

User'ları Çekme

GET

gettAllUsers → /api/users

Örnek Request:



Örnek Response:

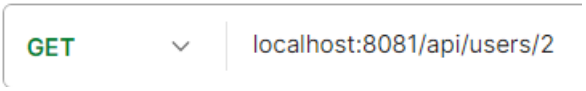

```
[
    {
        "id": 4,
        "name": "Anil",
        "surname": "Acar"
    },
    {
        "id": 2,
        "name": "Kaya",
        "surname": "Acar"
    }
```


İd'ye göre User Çekme

GET

getUserById →/api/users/{userId}

Örnek Request:



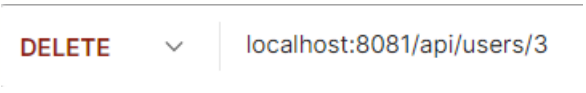
Örnek Response:


```
{
    "id": 2,
    "name": "Kaya",
    "surname": "Acar"
}
```


User Sil

deleteUser → /api/users/{userId}

Örnek Request:



User Ekle

POST

createUser → /api/users

Request Body:

```
{
   "name": "String",
   "surname": "String",
   "address": {
      "id": 2
   }

}
```

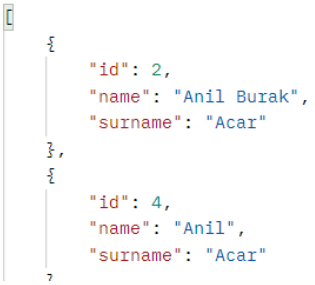
Response Body:

```
[
 {
   "id": "1",
   "name": "String",
   "surname": "String",
  },
{
   "id": "2",
   "surname": "String",
   "address": {
      "id": 3
    }
  }

]
```

Örnek Request:

GET localhost:8081/api/users

Örnek Response:



User Güncelle

PUT

updateUser → /api/users/{userId}

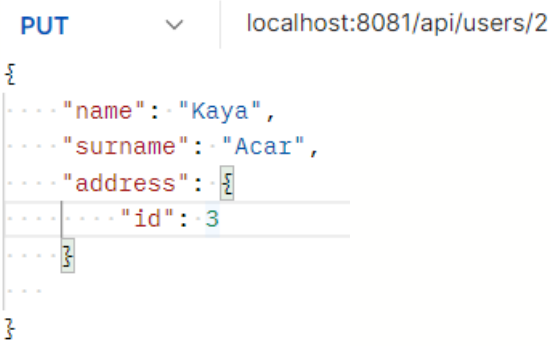
Request Body:

```
{
  "name": "Anil Burak",
  "surname": "Acar",
  "address": {
    "id": 3
  }

}
```

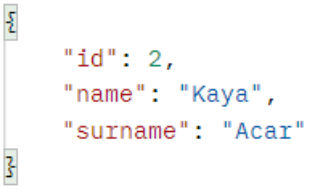
Response Body:

```
{
  "id": "1",
  "name": "String",
  "surname": "String",
 },
```

Örnek Request:



Örnek Response:



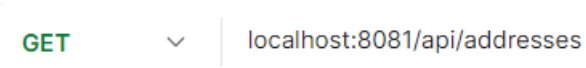
- **AddressController API'si(/api/addresses)**

Address'leri Çekme

GET

gettAllAddresses → /api/addresses

Örnek Request:



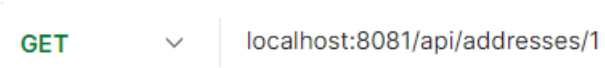
Örnek Response:

İd'ye göre Address Çekme

GET

getAddressById →/api/ addresses /{addressId}

Örnek Request:

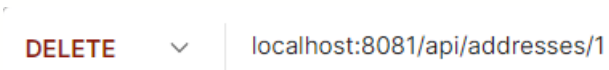


Örnek Response:

Address Sil

deleteAddress → /api/ addresses /{ addressId }

Örnek Request:



Address Ekle

POST

createAddress → /api/addresses

Request Body:

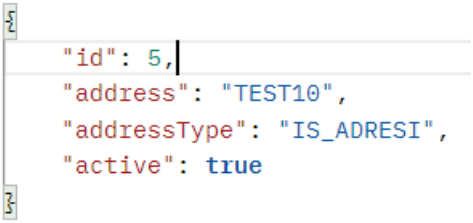
```
{
  "address": "String",
  "addressType": "1",
  "active":true

}
```

Response Body:

```
{
  "id": 5,
  "address": "String",
  "addressType": "enum",
  "active": true
}
```

Örnek Request:

Örnek Response:



Address Güncelle

PUT

updateAddress → /api/addresses/{addressId}

Request Body:

```
{
   "address": "String",
   "addressType": "1",
   "active":true

}
```

Response Body:

```
{
   "id": 5,
   "address": "String",
   "addressType": "enum",
   "active": true
}
```

Örnek Request:

PUT localhost:8081/api/addresses/5

```
{
    "address": "TEST85",
    "addressType": "1",
    "active":true

}
```

Örnek Response:

```
{
    "id": 5,
    "address": "TEST85",
    "addressType": "IS_ADRESI",
    "active": true
}
```