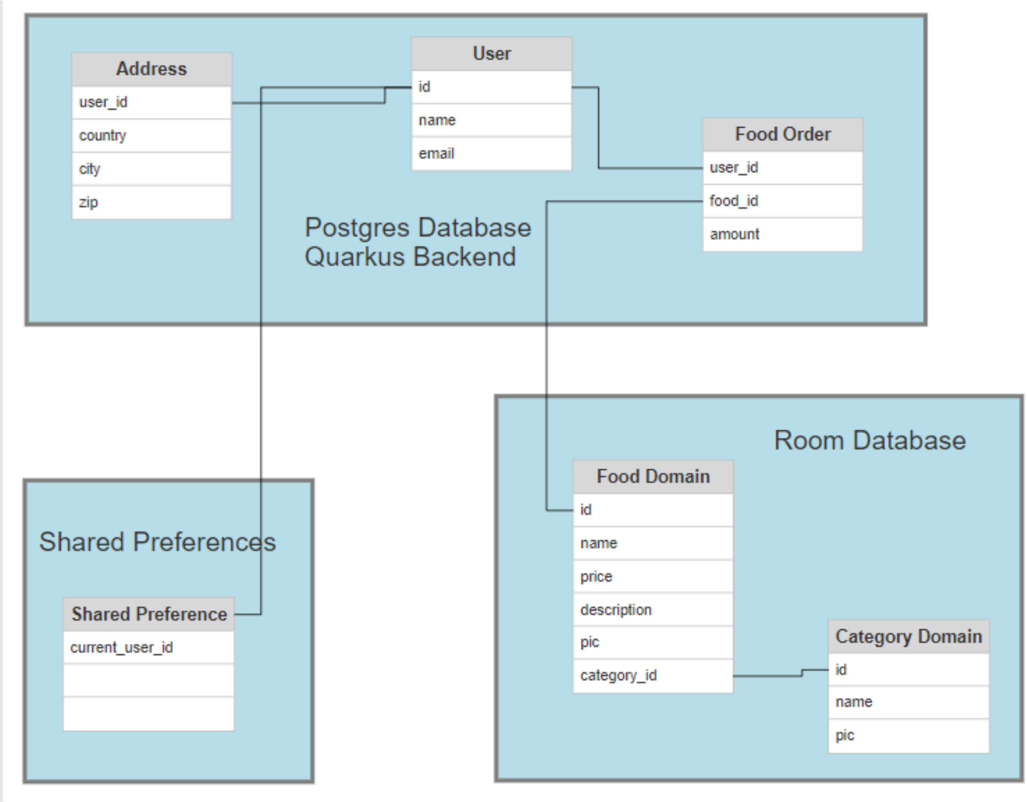
FoodDelivery

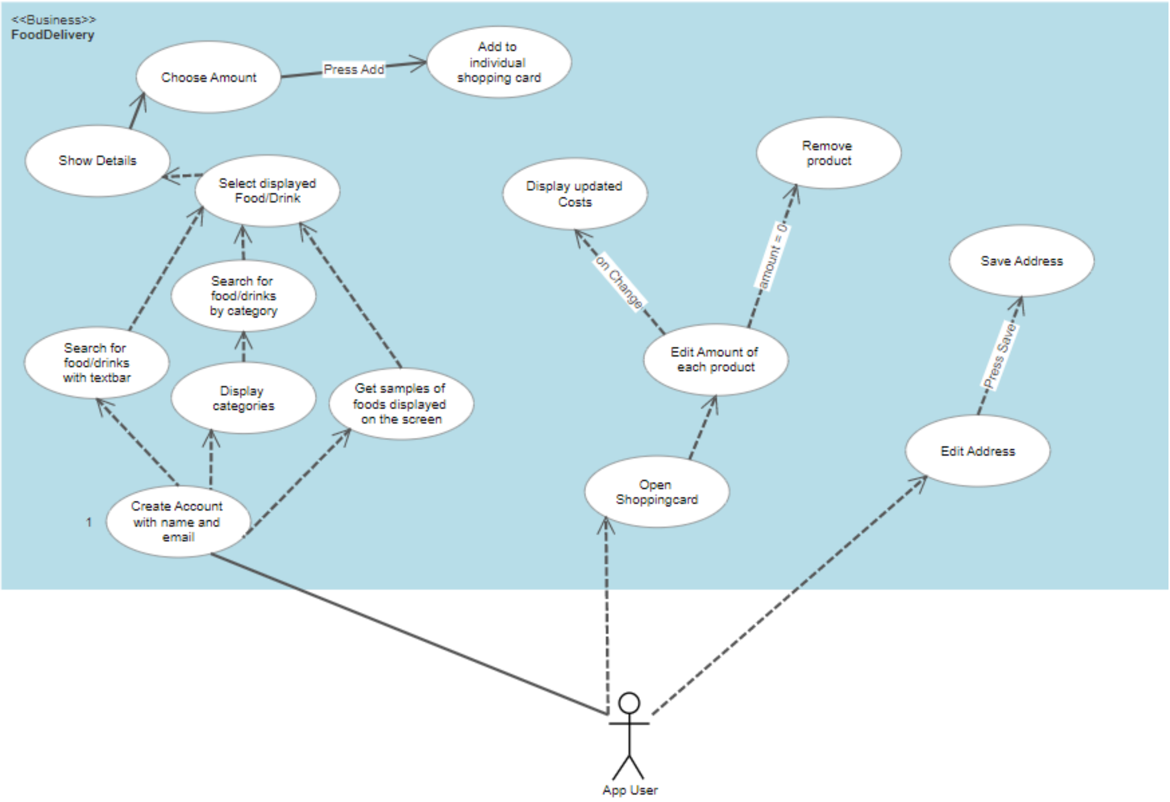
Inhalt

Project Structure

ERD



UseCase

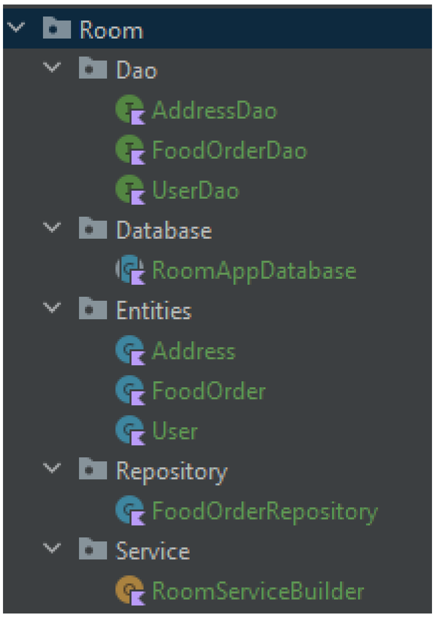

<<Business>>
FoodDelivery
Choose Amount
Press Add
Add to individual shopping card
Show Details
Select displayed Food/Drink
Search for food/drinks by category
Search for food/drinks with textbar
Display categories
Get samples of foods displayed on the screen
1
Create Account with name and email
Display updated Costs
on Change
Remove product
amount = 0
Edit Amount of each product
Open Shoppingcard
Save Address
Press Save
Edit Address
App User

Frontend

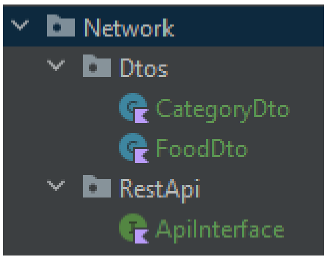
Dependencies

```
// retrofit dependencies
implementation 'com.squareup.retrofit2:retrofit:2.6.1'
implementation 'com.squareup.retrofit2:converter-gson:2.6.1'
implementation 'com.jakewharton.picasso:picasso2-okhttp3-downloader:1.1.0'

// Room
implementation("androidx.room:room-runtime:2.4.1")
annotationProcessor("androidx.room:room-compiler:2.4.1")

// To use Kotlin annotation processing tool (kapt)
kapt("androidx.room:room-compiler:2.4.1")
```

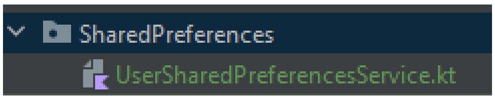
Architecture



```
Room
    Dao
        AddressDao
        FoodOrderDao
        UserDao
    Database
        RoomAppDatabase
    Entities
        Address
        FoodOrder
        User
    Repository
        FoodOrderRepository
    Service
        RoomServiceBuilder
```



```
Network
    Dtos
        CategoryDto
        FoodDto
    RestApi
        ApiInterface
```

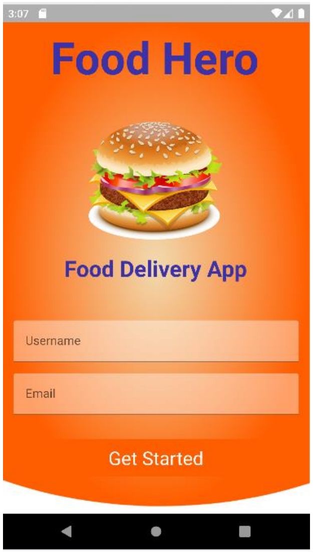


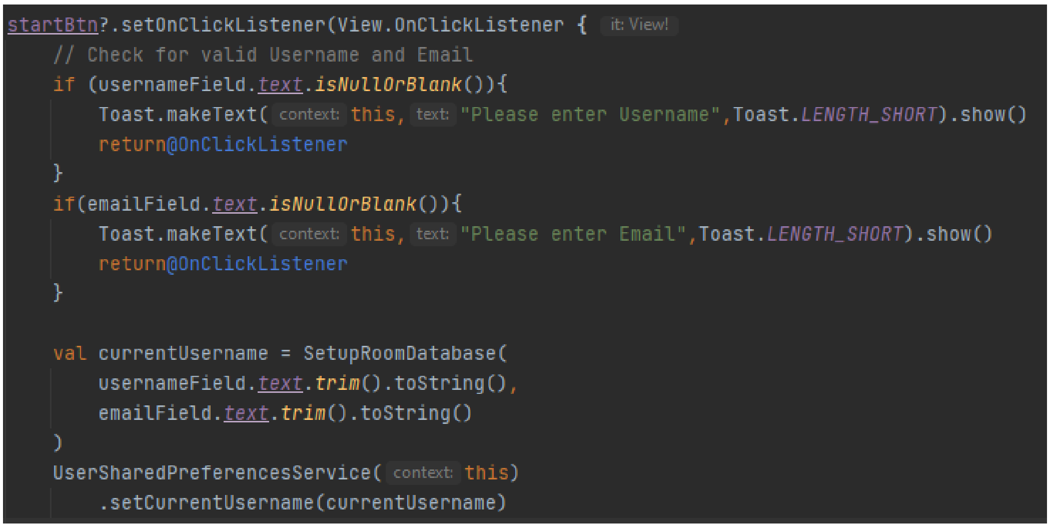
```
SharedPreferences
    UserSharedPreferencesService.kt
```

Marc Kruiß Donnerstag, 10. Februar 2022

FoodDelivery

Introscreen

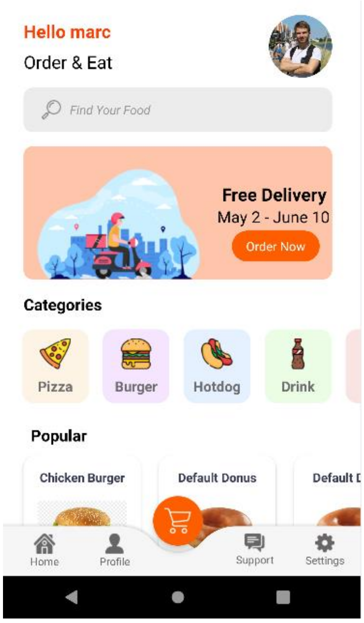




```
startBtn?.setOnClickListener(View.OnClickListener {  it: View!
    // Check for valid Username and Email
    if (usernameField.text.isNullOrBlank()){
        Toast.makeText( context: this,  text: "Please enter Username",Toast.LENGTH_SHORT).show()
        return@OnClickListener
    }
    if(emailField.text.isNullOrBlank()){
        Toast.makeText( context: this,  text: "Please enter Email",Toast.LENGTH_SHORT).show()
        return@OnClickListener
    }

    val currentUsername = SetupRoomDatabase(
        usernameField.text.trim().toString(),
        emailField.text.trim().toString()
    )
    UserSharedPreferencesService( context: this)
        .setCurrentUsername(currentUsername)
```

After entering the Username and Email, a new Room-User-Entry will be created. Save the input user in the shared preferences Key-Value Pair.
The Input Email/Name will be controlled before redirect to the Home-Fragment.

Homescreen



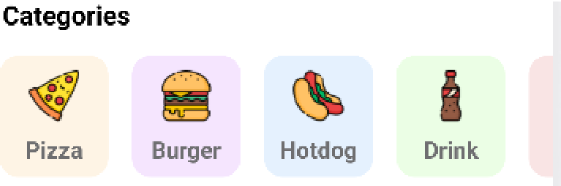


Figure 2 Categories from the backend

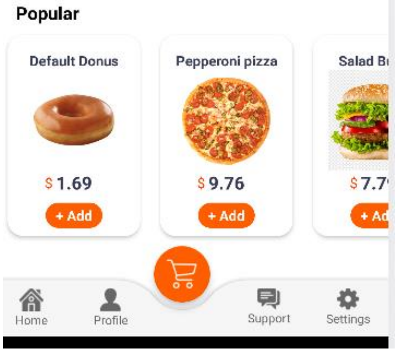


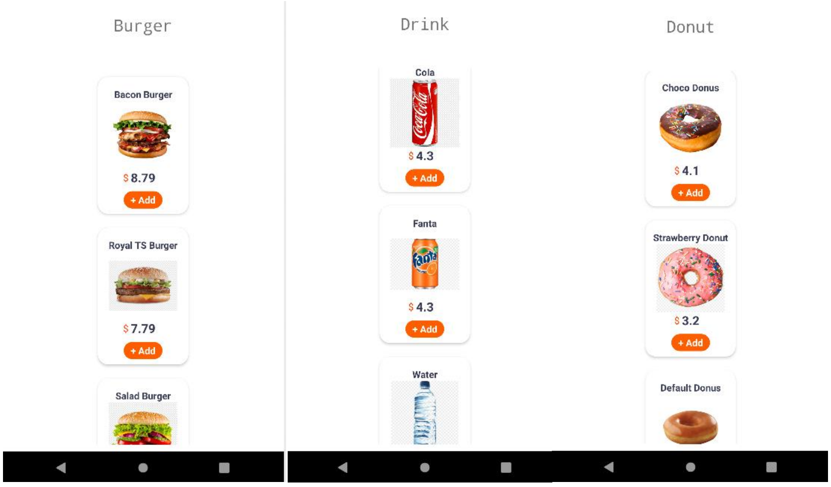
Figure 1 4 Random Sample data from backend

The app loads the Categories and 4 Sample Food-Data from the database and display these things on the home-fragment. The Categories and "Populars" are clickable to redirect to a new page.

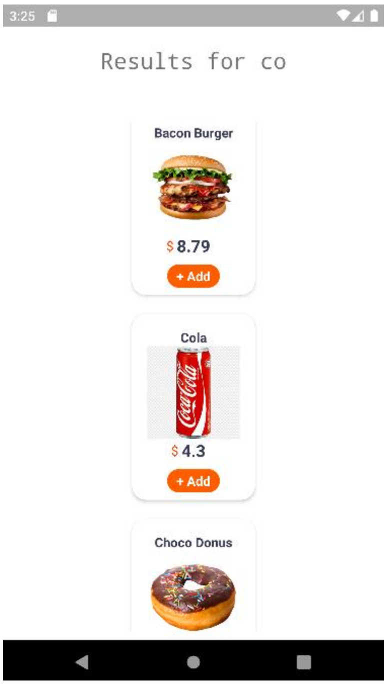
Also, the Searchbar can be used to search for a product

FoodDelivery

Filter by Categories

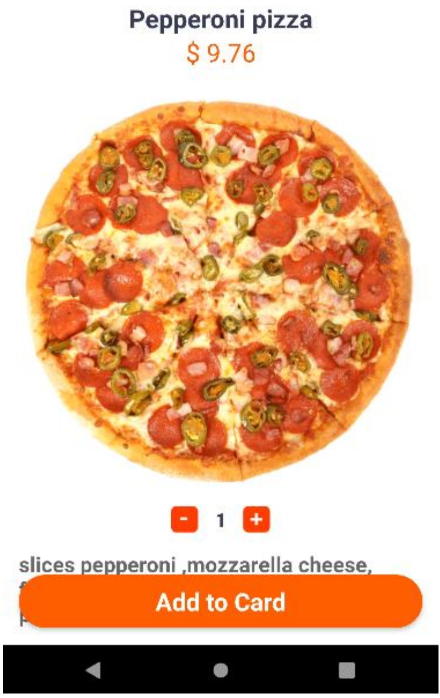


Filter by search (example "co")



FoodDelivery

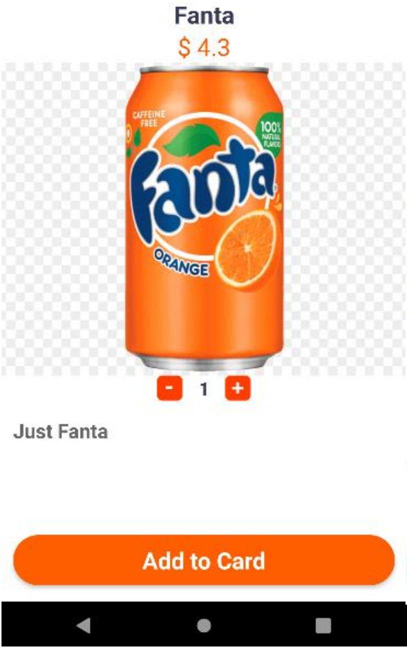
Product Detail



Pepperoni pizza

$ 9.76

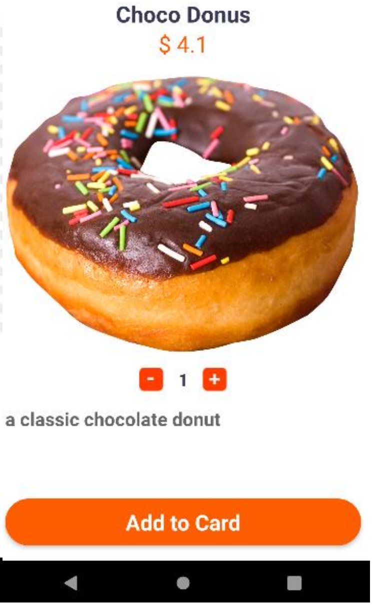
slices pepperoni ,mozzarella cheese,



Fanta

$ 4.3

Just Fanta

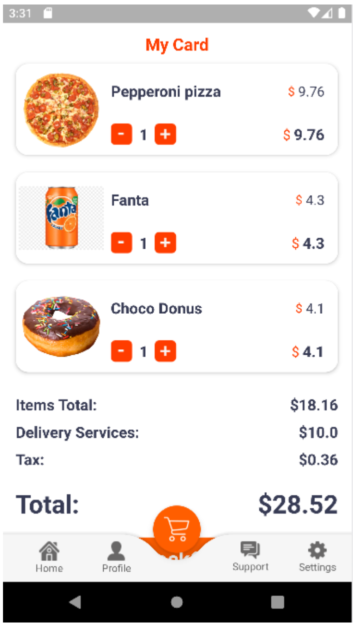


Choco Donus

$ 4.1

a classic chocolate donut

Shopping Cart



My Card

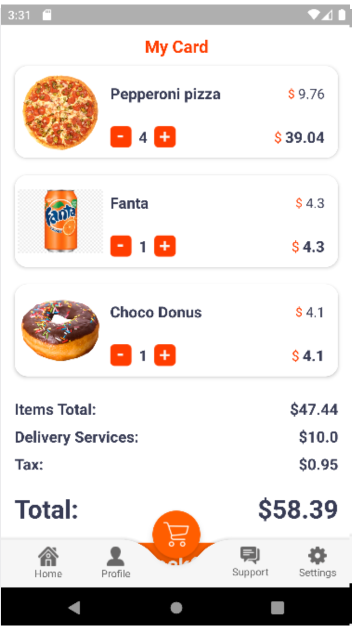
Pepperoni pizza	1	$ 9.76 $ 9.76
Fanta	1	$ 4.3 $ 4.3
Choco Donus	1	$ 4.1 $ 4.1

Items Total: $18.16

Delivery Services: $10.0

Tax: $0.36

Total: $28.52



My Card

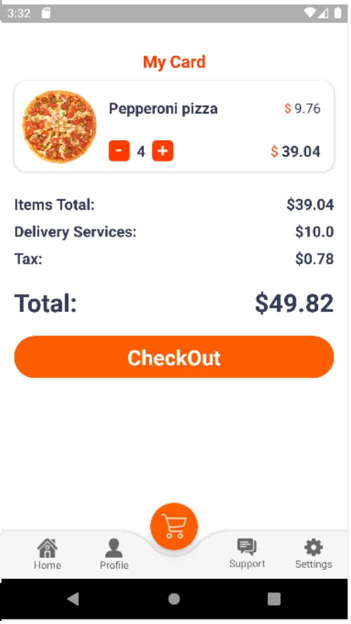
Pepperoni pizza	4	$ 9.76 $ 39.04
Fanta	1	$ 4.3 $ 4.3
Choco Donus	1	$ 4.1 $ 4.1

Items Total: $47.44

Delivery Services: $10.0

Tax: $0.95

Total: $58.39



My Card

Pepperoni pizza	4	$ 9.76 $ 39.04

Items Total: $39.04

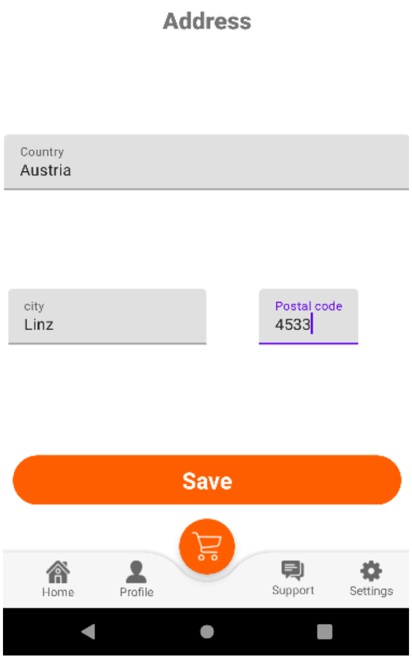
Delivery Services: $10.0

Tax: $0.78

Total: $49.82

These things will be saved, so when you re-log into the app, the shopping card will look the same as when you left it.

Address

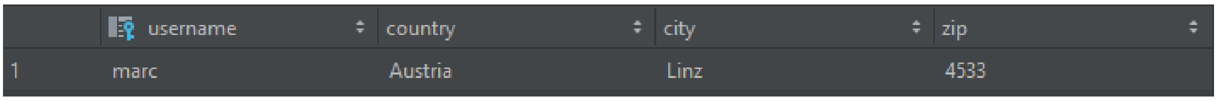


Address

Country
Austria

city
Linz

Postal code
4533



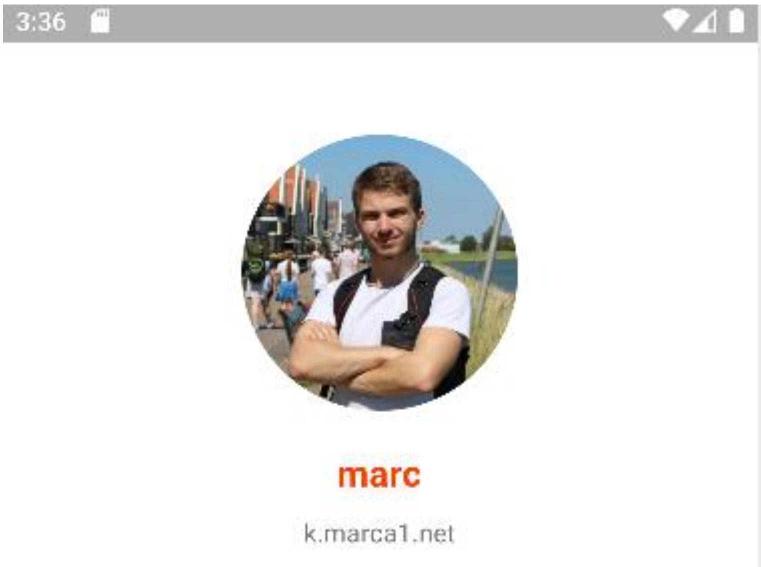
	username	country	city	zip
1	marc	Austria	Linz	4533

Figure 3 Room Database Address Entry

This Entry can also be updated when changed and saved.

Profile

In the profile section you can also have a look at your credentials read from the locale room database



marc

k.marca1.net

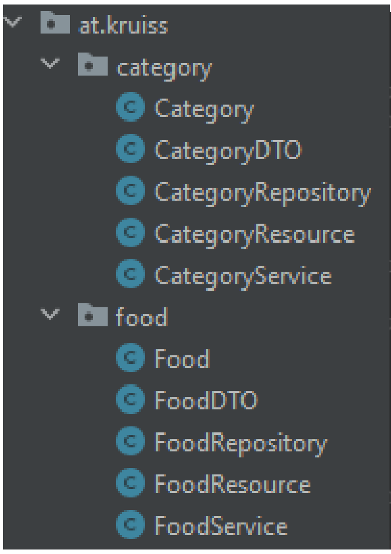
FoodDelivery

Backend

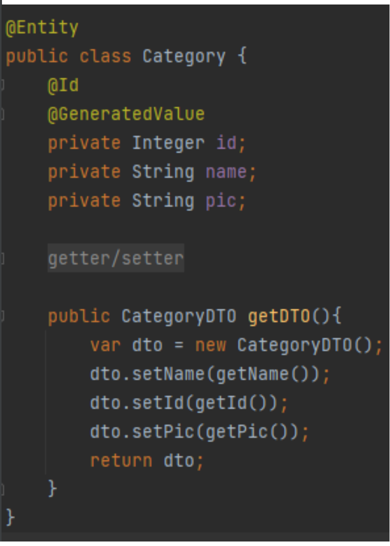
Datasource config

```
# datasource configuration
quarkus.datasource.db-kind = postgresql
quarkus.datasource.username = admin
quarkus.datasource.password = postgres
quarkus.datasource.jdbc.url =
jdbc:postgresql://localhost:5432/postgres?currentSchema=public

# drop and create the database at startup (use update to only update the
schema)
quarkus.hibernate-orm.database.generation=drop-and-create

quarkus.http.port=8080
quarkus.http.cors=true
```

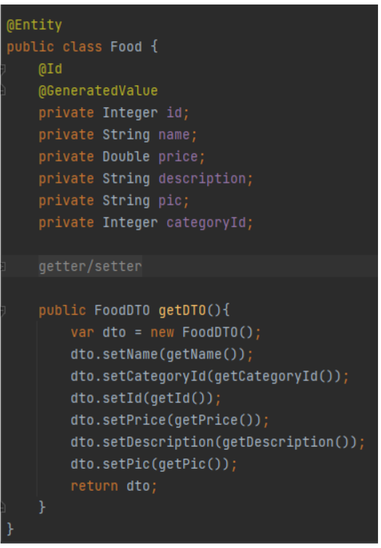
Folder structure



```
at.kruiss
  category
    Category
    CategoryDTO
    CategoryRepository
    CategoryResource
    CategoryService
  food
    Food
    FoodDTO
    FoodRepository
    FoodResource
    FoodService
```



```
@Entity
public class Category {
    @Id
    @GeneratedValue
    private Integer id;
    private String name;
    private String pic;

    getter/setter

    public CategoryDTO getDTO(){
        var dto = new CategoryDTO();
        dto.setName(getName());
        dto.setId(getId());
        dto.setPic(getPic());
        return dto;
    }
}
```

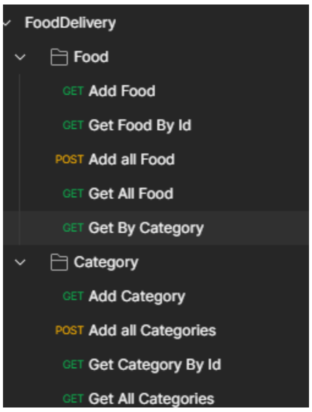


```
@Entity
public class Food {
    @Id
    @GeneratedValue
    private Integer id;
    private String name;
    private Double price;
    private String description;
    private String pic;
    private Integer categoryId;

    getter/setter

    public FoodDTO getDTO(){
        var dto = new FoodDTO();
        dto.setName(getName());
        dto.setCategoryId(getCategoryId());
        dto.setId(getId());
        dto.setPrice(getPrice());
        dto.setDescription(getDescription());
        dto.setPic(getPic());
        return dto;
    }
}
```

Docker setup

docker run --name dbiPostgres -e POSTGRES_PASSWORD=postgres -e POSTGRES_USER=admin -d -p 5432:5432 -v dbiPostgres:/var/lib/postgres postgres:alpine

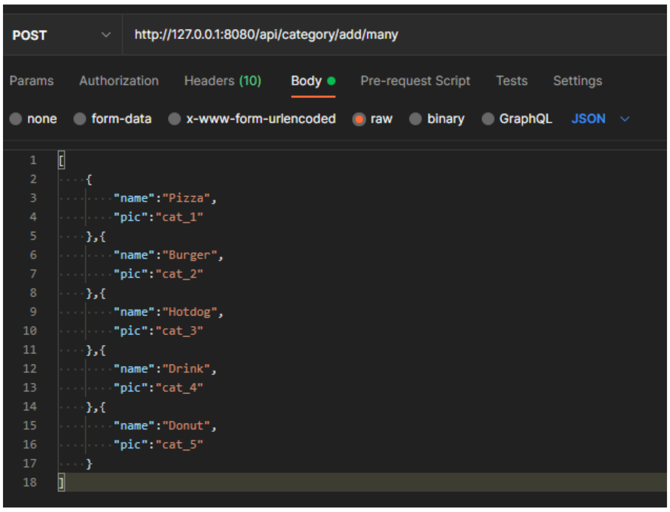
Postman data



FoodDelivery
- Food
  - GET Add Food
  - GET Get Food By Id
  - POST Add all Food
  - GET Get All Food
  - GET Get By Category
- Category
  - GET Add Category
  - POST Add all Categories
  - GET Get Category By Id
  - GET Get All Categories

Service testing with Postman

Add All Categories

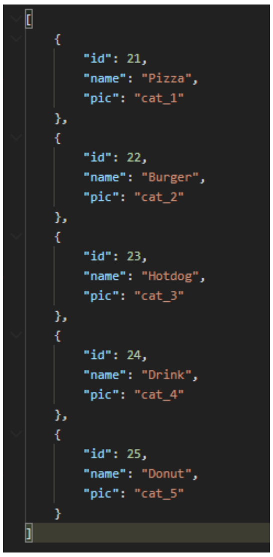


POST http://127.0.0.1:8080/api/category/add/many

● raw JSON

```json
[
    {
        "name":"Pizza",
        "pic":"cat_1"
    },{
        "name":"Burger",
        "pic":"cat_2"
    },{
        "name":"Hotdog",
        "pic":"cat_3"
    },{
        "name":"Drink",
        "pic":"cat_4"
    },{
        "name":"Donut",
        "pic":"cat_5"
    }
]
```

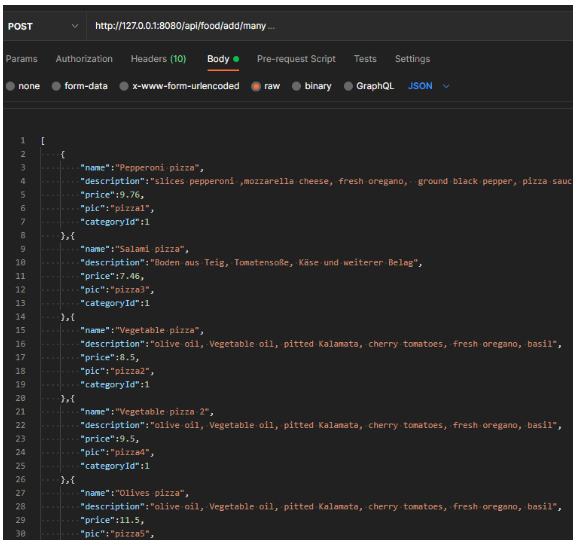
Figure 5 Add all categories



```json
[
    {
        "id": 21,
        "name": "Pizza",
        "pic": "cat_1"
    },
    {
        "id": 22,
        "name": "Burger",
        "pic": "cat_2"
    },
    {
        "id": 23,
        "name": "Hotdog",
        "pic": "cat_3"
    },
    {
        "id": 24,
        "name": "Drink",
        "pic": "cat_4"
    },
    {
        "id": 25,
        "name": "Donut",
        "pic": "cat_5"
    }
]
```

Figure 4 Response

Add All Products

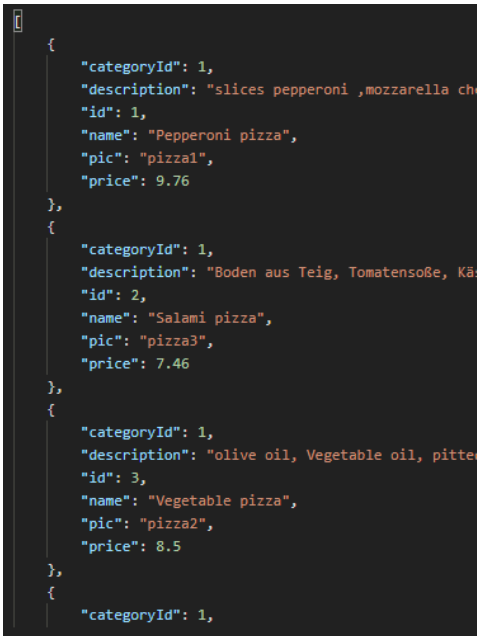


POST http://127.0.0.1:8080/api/food/add/many ...

● raw JSON

```json
[
    {
        "name":"Pepperoni pizza",
        "description":"slices pepperoni ,mozzarella cheese, fresh oregano,  ground black pepper, pizza sauc
        "price":9.76,
        "pic":"pizza1",
        "categoryId":1
    },{
        "name":"Salami pizza",
        "description":"Boden aus Teig, Tomatensoße, Käse und weiterer Belag",
        "price":7.46,
        "pic":"pizza3",
        "categoryId":1
    },{
        "name":"Vegetable pizza",
        "description":"olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",
        "price":8.5,
        "pic":"pizza2",
        "categoryId":1
    },{
        "name":"Vegetable pizza 2",
        "description":"olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",
        "price":9.5,
        "pic":"pizza4",
        "categoryId":1
    },{
        "name":"Olives pizza",
        "description":"olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",
        "price":11.5,
        "pic":"pizza5",
```

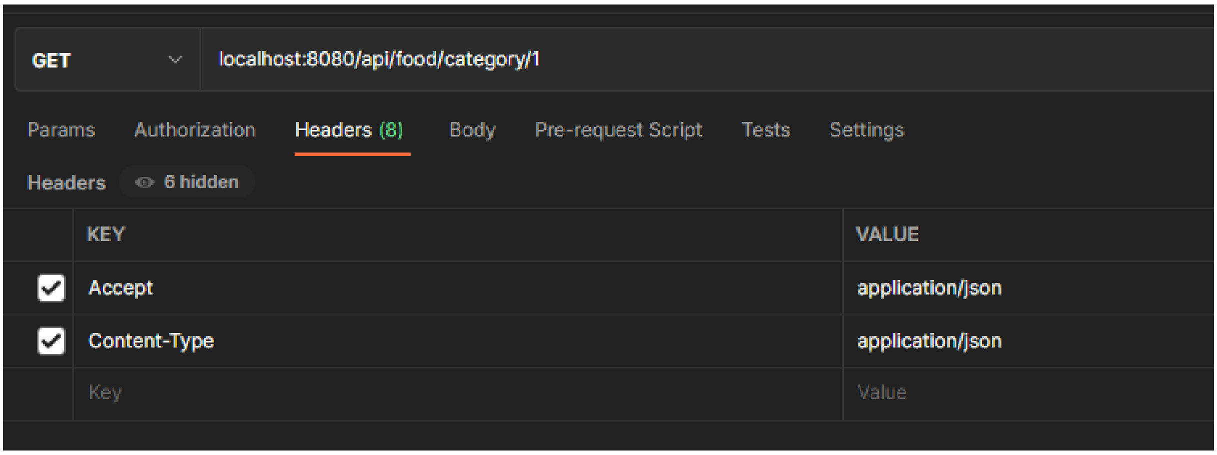
Figure 7 Add all products



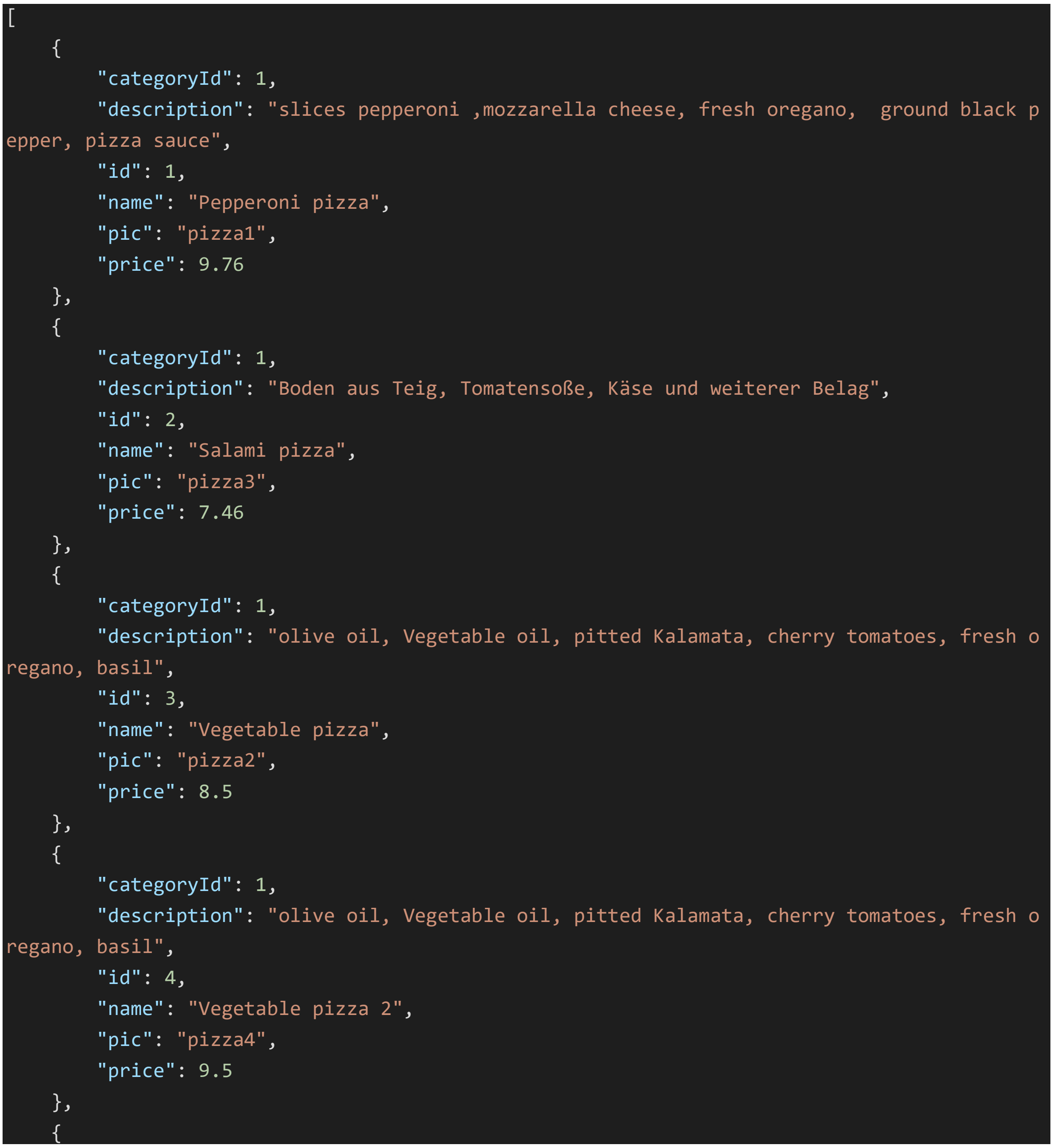
```json
[
    {
        "categoryId": 1,
        "description": "slices pepperoni ,mozzarella ch
        "id": 1,
        "name": "Pepperoni pizza",
        "pic": "pizza1",
        "price": 9.76
    },
    {
        "categoryId": 1,
        "description": "Boden aus Teig, Tomatensoße, Kä
        "id": 2,
        "name": "Salami pizza",
        "pic": "pizza3",
        "price": 7.46
    },
    {
        "categoryId": 1,
        "description": "olive oil, Vegetable oil, pitte
        "id": 3,
        "name": "Vegetable pizza",
        "pic": "pizza2",
        "price": 8.5
    },
    {
        "categoryId": 1,
```

Figure 6 Respoonse

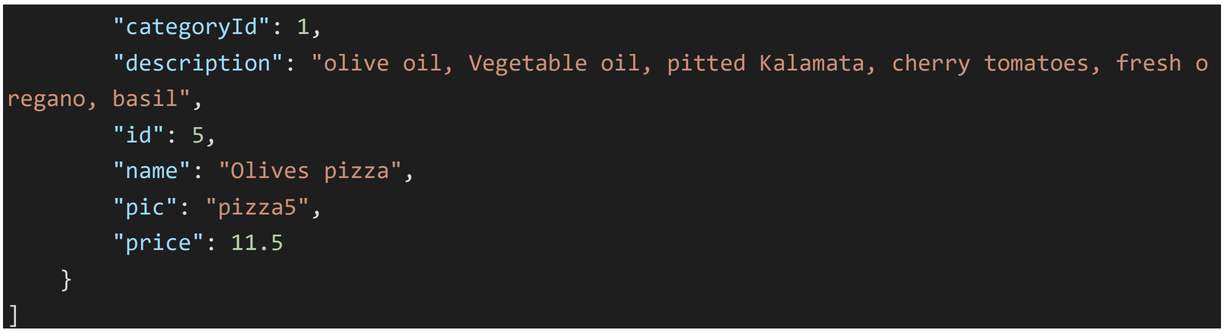
Get Products by Category 1 (= Pizza-Category)



	KEY	VALUE
☑	Accept	application/json
☑	Content-Type	application/json
	Key	Value

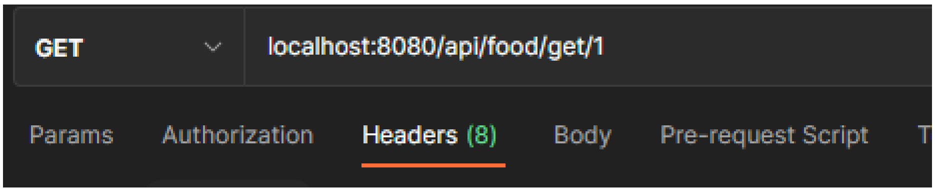


```
[
    {
        "categoryId": 1,
        "description": "slices pepperoni ,mozzarella cheese, fresh oregano,  ground black p
epper, pizza sauce",
        "id": 1,
        "name": "Pepperoni pizza",
        "pic": "pizza1",
        "price": 9.76
    },
    {
        "categoryId": 1,
        "description": "Boden aus Teig, Tomatensoße, Käse und weiterer Belag",
        "id": 2,
        "name": "Salami pizza",
        "pic": "pizza3",
        "price": 7.46
    },
    {
        "categoryId": 1,
        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh o
regano, basil",
        "id": 3,
        "name": "Vegetable pizza",
        "pic": "pizza2",
        "price": 8.5
    },
    {
        "categoryId": 1,
        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh o
regano, basil",
        "id": 4,
        "name": "Vegetable pizza 2",
        "pic": "pizza4",
        "price": 9.5
    },
    {
```

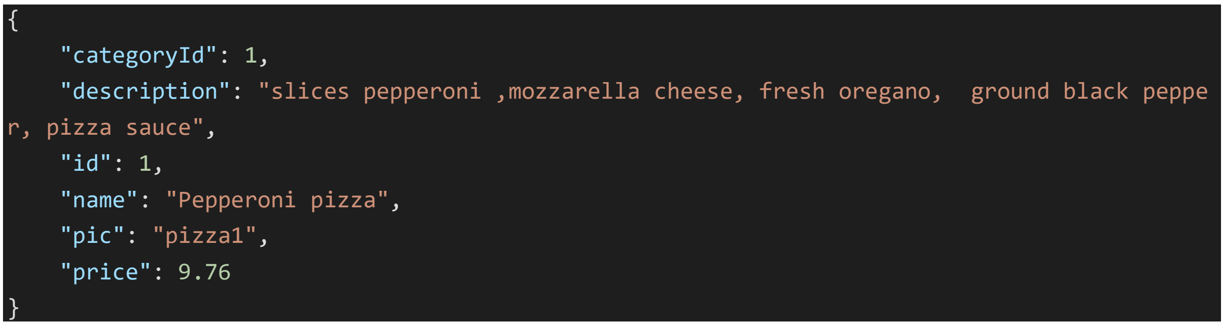


```
        "categoryId": 1,
        "description": "olive oil, Vegetable oil, pitted Kalamata, cherry tomatoes, fresh oregano, basil",
        "id": 5,
        "name": "Olives pizza",
        "pic": "pizza5",
        "price": 11.5
    }
]
```

Get Food By Id

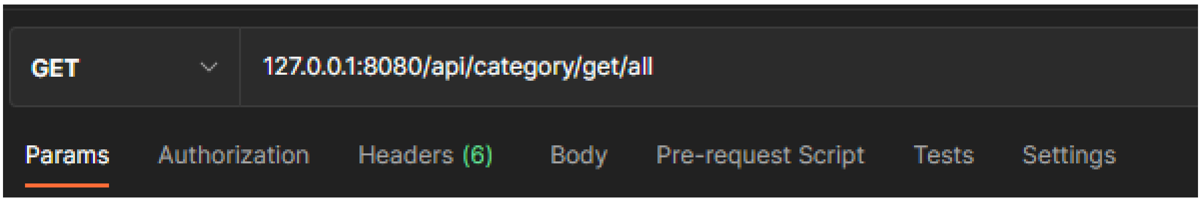


```
GET  localhost:8080/api/food/get/1
```

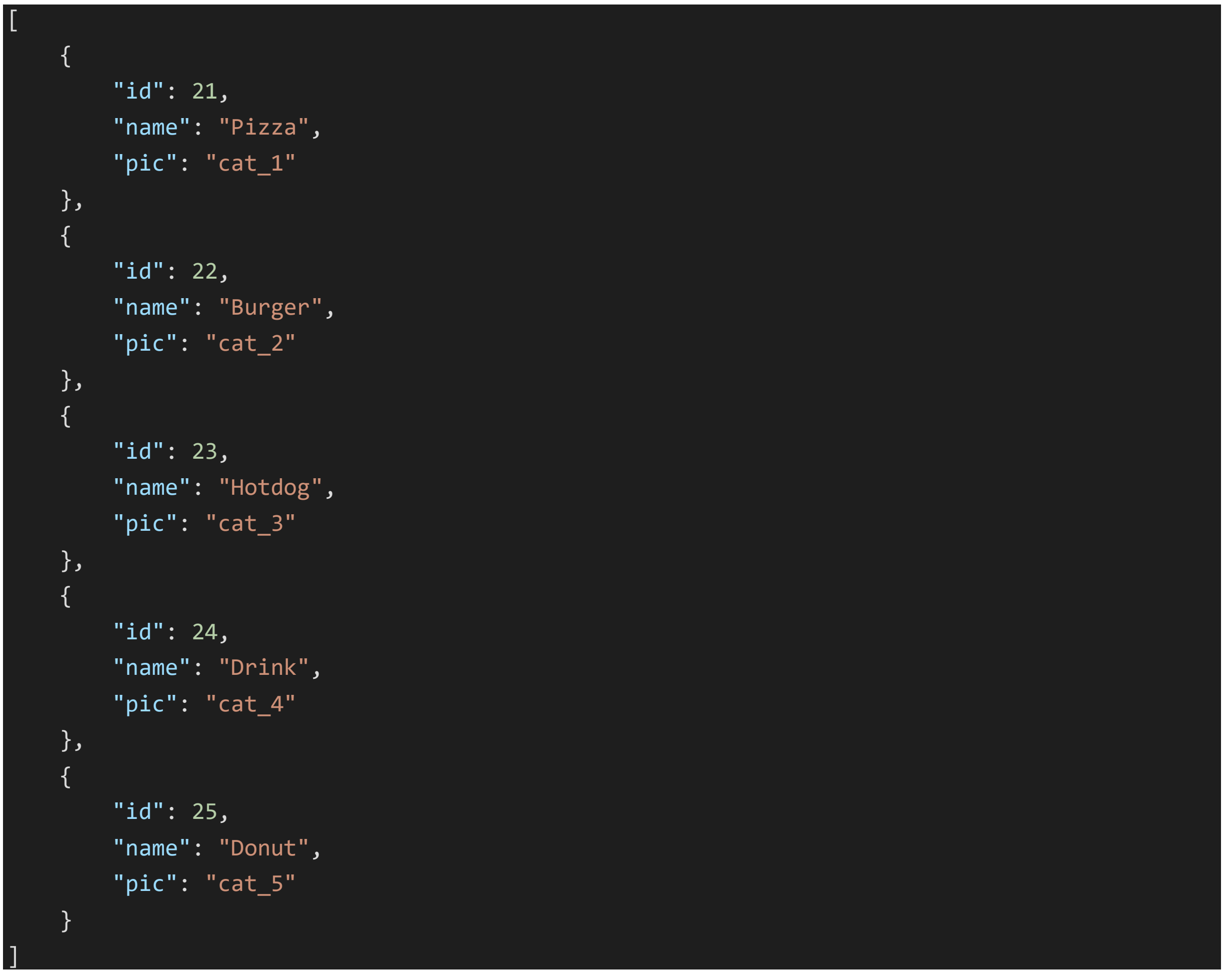


```
{
    "categoryId": 1,
    "description": "slices pepperoni ,mozzarella cheese, fresh oregano,  ground black pepper, pizza sauce",
    "id": 1,
    "name": "Pepperoni pizza",
    "pic": "pizza1",
    "price": 9.76
}
```

Get All Categories



GET 127.0.0.1:8080/api/category/get/all



```
[
    {
        "id": 21,
        "name": "Pizza",
        "pic": "cat_1"
    },
    {
        "id": 22,
        "name": "Burger",
        "pic": "cat_2"
    },
    {
        "id": 23,
        "name": "Hotdog",
        "pic": "cat_3"
    },
    {
        "id": 24,
        "name": "Drink",
        "pic": "cat_4"
    },
    {
        "id": 25,
        "name": "Donut",
        "pic": "cat_5"
    }
]
```