**Як запустити прект.**

1. **Розархівовуєш в папку.**
2. **відкриваєш директорію в ідеєшці і запускаєш консольку.**
3. **Прописуєш в консольці і запускаєш « npm i »**
4. **Після того, як закінчиться загрузка, прописуєш “gulp”**
5. **В браузері відкриється сторінка**
6. **Тестовий юзер для входу: логін: q пароль: q**
7. **Працює все крім додавання і редагування категорій в адмінці. Їм треба ід а їх я не прописував. Якщо питання, я на зв’язку.**

**Documentation for Alcohol Rest API**

**Template:**

**Entity name**

**1.url: http: //localhost: 8080/entity/save(example),**

**2.Type method: GET POST DELETE PUT,**

**3.Body to API(type JSON):**

**Example:**

**{**

name: Name,

},

**4.Body from API(type JSON):**

{

name: Name,

}

**User GET**

1.http: //localhost: 3000/user/GET,

2.GET

3.{

email: str

password: str

4.{

id: int,

name: str,

sname: str,

birthday: {

day: int,

month: int,

year: int

}

}

**User add**

1.http: //localhost: 3000/user/add,

2.POST

3.{

name: str

sname: str

phone: str

email: str,

birthday: {

day: int,

month: int,

year: int

}

password: str

4.{

id: 1,

name: str,

sname: str,

email: str,

birthday: {

day: int,

month: int,

year: int

}

}

User update

1.http: //localhost: 3000/user/update,

2.PUT

3.{

id: int,

name: str

sname: str

phone: str

email: str,

birthday: {

day: int,

month: int,

year: int

}

password: str

4.{

id: int,

name: str,

sname: str,

email: str,

birthday: {

day: int,

month: int,

year: int

}

}

**User delete**

1.http: //localhost: 3000/user/delete,

2.DELETE

3.{

id: int

4.void

**Category**

**Get category**

1.http: //localhost: 3000/categories/GET,

2.GET

3.void

4.[{

id: int,

name: str,

},{}

]

**Save category**

1.http: //localhost: 3000/categories/add,

2.POST

3.{

name: str

}

4.{

id: int,

name: str

}

**Update category**

1.http: //localhost: 8080/categories/update,

2.PUT

3.{

id: 1,

name: str

}

4.{

id: 1,

name: str

}

**Delete category**

1.http: //localhost: 8080/categories/delete,

2.DELETE

3.void

4.void

**Commodity**

**All commodities**

1.http: //localhost: 8080/commodities/GET,

2.GET

3.void

4.[

{

id: int

name: str,

category: {

id: int,

name: str

},

brand: str,

country: str,

strength: int

volume: int, (ml)

price: int,

price2: int,

image: ????,

description: str,

promo: true\false

},{}

]

**Save commodities**

1.http: //localhost: 8080/commodities/add,

2.POST

3.{

name: str

category: int,

brand: str,

country: str,

strength: int

volume: int,

price: int,

price2: int,

image: ????,

description: str,

promo: true\false

}

4. {

id: int

name: str

category: str,

brand: str,

country: str,

strength: int

volume: int,

price: int,

price2: int,

image: ????,

description: str,

promo: true\false

}

**Update commodities**

1.http: //localhost: 8080/commodities/update,

2.PUT

3.{

id: int

name: str

category: str

brand: str,

country: str,

strength: int

volume: int,

price: int,

price2: int,

image: ????,

description: str,

promo: true\false

}

4.{

{

id: int

name: str

category: str,

brand: str,

country: str,

strength: int

volume: int,

price: int,

price2: int,

image: ????,

description: str,

promo: true\false

}

**Delete commodities**

1.http: //localhost: 8080/commodities/delete,

2.DELETE

3.void

4.void

**Banner**

**Get banner**

1.http: //localhost: 3000/banner/GET,

2.GET

3.void

4.[{

id: int,

picture: str,

commodity\_id:int

},{}

]

**Save banner**

1.http: //localhost: 3000/banner/add,

2.POST

3.{

picture: base64,

commodity\_id: int

}

4.{

id: int,

picture: str,

commodity\_id: int

}

**Update banner**

1.http: //localhost: 8080/banner/update,

2.PUT

3.{

id: int,

picture: str,

commodity\_id: int

}

4.{

id: int,

picture: str,

commodity\_id: int

}

**Delete banner**

1.http: //localhost: 8080/banner/delete,

2.DELETE

3.void

4.void

**Order**

**Order**

1.http: //localhost: 8080/orders/add,

2.add

3.{

id: int

archive: false,

name: str,

sname: str

email: str,

phone: str,

goodsOrdered: [

{

id: int,

count: int

},{}],

address {

street: str,

house: str,

flat: str,

stair: str,

entrance: str,

}

comments: str

}

4. {

archive: true/false,

name: str,

sname: str

email: str,

phone: str,

goodsOrdered: [

{

id: int,

name: str,

count: int,

price: int

},{}

],

address: {

street: str,

house: str,

flat: str,

stair: str,

entrance: str,

}

comments: str

}