

Application : VIDLY (<https://floating-savannah-86891.herokuapp.com>)

- Rent movies and we use and API for access

## **Sprint 1**

Story1:

As a manager I want an API for user login

- Remark: Only an admin user can delete movies or genre movies
- From API we cannot create an admin user, this should be perform only by system admin manipulating DB directly

Story2:

As a manager I want an API for customers

### **1. /api/users**

```
{
  name: {
    type: String,
    required: true,
    minlength: 5,
    maxlength: 50
  },

  email: {
    type: String,
    required: true,
    minlength: 5,
    maxlength: 255,
    unique: true
  },

  password: {
    type: String,
    required: true,
    minlength: 5,
    maxlength: 1024
  },

  isAdmin: Boolean
}
```

## Operations:

**POST /api/users : create a user**

**GET /api/users/me : information about current user (use with /api/auth generated token)**

**!!!isAdmin is not allowed , this should only be added by the system admin!!!**

Ex: creating a user:

POST <https://floating-savannah-86891.herokuapp.com/api/users>

```
{
  "name": "testare",
  "email": "testare@yahoo.com",
  "password": "justtesting"
}
```

Receive a 200 OK and an ID

```
{
  "_id": "5e5c2b2233e7f200173d17d0",
  "name": "testare",
  "email": "testare@yahoo.com"
}
```

## 2. /api/auth: used for login

```
email: {
  type: String,
  required: true,
  minlength: 5,
  maxlength: 255,
  unique: true
}

password: {
  type: String,
  required: true,
  minlength: 5,
  maxlength: 1024
}
```

## Operations:

**POST /api/auth : for login, we get a token**

```
{  
  "email": "testare@yahoo.com",  
  "password": "justtesting"  
}
```

Token will be used in Header :

x-auth-token : <token>

*Remark: see <https://jwt.io/> (Json Web Token)*

## 3. /api/customers : used for clients

```
name: {  
  type: String,  
  required: true,  
  minlength: 5,  
  maxlength: 50  
},  
isGold: {  
  type: Boolean,  
  default: false  
},  
phone: {  
  type: String,  
  required: true,  
  minlength: 5,  
  maxlength: 50  
}  
}
```

## Operations:

### POST /api/customers

```
{
  "name" : "customer1",
  "isGold": true,
  "phone": "0123456789"
}
```

Will return an id: `"_id": "5e5c353e33e7f200173d17d1"`

PUT /api/customers/<id> : to modify a client

GET /api/customers : to return all clients

DELETE /api/customers/<id> : to delete a client

---

## Sprint 2 :

### Story3:

As a manager I want an API for genres

As a manager I want an API for movies

As a manager I want an API for rentals

## 4. /api/genres : used to add/see genres movies

```
name: {
  type: String,
  required: true,
  minlength: 5,
  maxlength: 50
}
```

## Operations:

### GET

/api/genres : to see all genres

/api/genres/<id> : to see a specific genre

## POST

/api/genres : to add a genre

!!!! you cannot add a genre with a authentication token , see /api/auth

## PUT : to modify a genre

/api/genres/<id>

## DELETE: to delete a genre

/api/genres/<id>

!!!! you can only delete a genre if you are an admin

## 5. /api/movies :used to add/see what movies can we rent

```
title: {
  type: String,
  required: true,
  trim: true,
  minlength: 5,
  maxlength: 255
},
genreId: {
  type: genreSchema, (id from the genres)
  required: true
},
numberInStock: {
  type: Number,
  required: true,
  min: 0,
  max: 255
},
dailyRentalRate: {
  type: Number,
  required: true,
  min: 0,
  max: 255
}
```

### Operations:

**GET /api/movies** : to return all movies

**GET /api/movies/<id>** : to return a specific movie

**POST /api/movies** : to add a movie

```
{
  "title": "titlu1",
  "genreId": "5e6b5e690424310017b74bbb",
  "numberInStock": 10,
  "dailyRentalRate": 3
}
```

**PUT /api/movies<id>**: to modify a movie

**DELETE /api/movies/<id>**: to delete a movie

## 6. /api/rentals

```
customerId :{
  type : customerId form customer
  required: true
},
movieId: {
  type : movieId form movie
  required: true
}

dateOut: {
  type: Date,
  required: true,
  default: Date.now
},
dateReturned: {
  type: Date
},
rentalFee: {
  type: Number,
  min: 0
}
```

### Operations:

**GET /api/rentals** : to return all rentals

**GET /api/rentals/<id>** : to return a specific rental

**POST /api/rentals : to add a rental**

```
{  
  "customerId": "5e654c52d0a8b200176d64aa",  
  "movieId": "5e6b67db0424310017b74bbc"  
}
```