

Image Server with REST API

Middleware Technologies for Distributed Systems

July 28th, 2020
Politecnico di Milano

Introduction

Image Server with REST API - Upload, retrieve and delete images as logged user

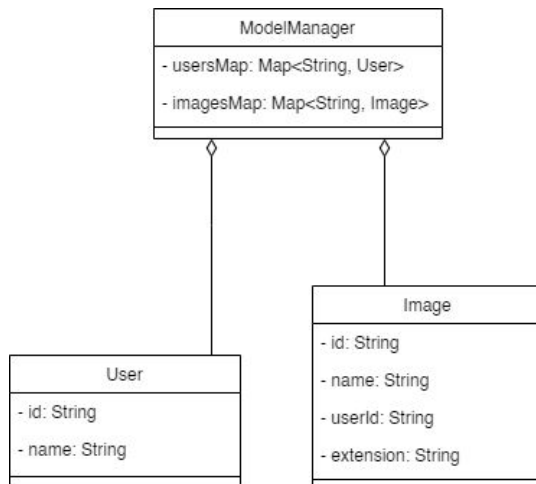
- Server implemented with Apache Spark in Java
- Running on AWS EC2 machine
- Serving Level 3 APIs with HATEOAS constraint
- Serving client static files to easily interact with APIs



Server structure

ModelManager: Single instance, reference point for data.

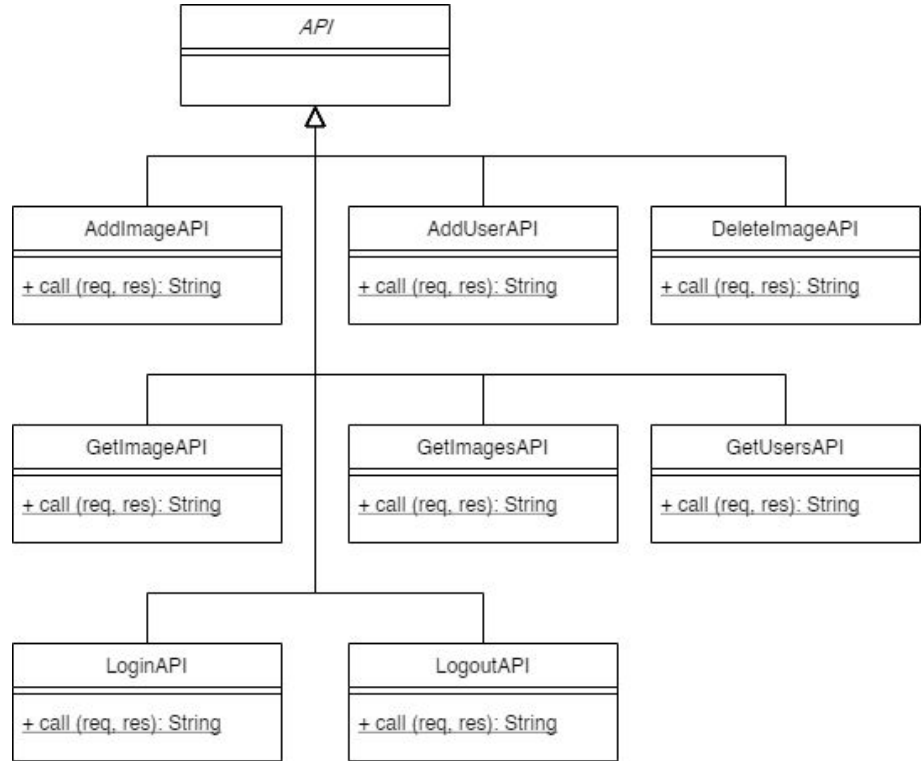
The only class to change if a database is needed



Server structure

API class: abstract with static methods

Each API has its own class with *call* method to compute the single operation needed



APIs

- GET /api/users
- POST /api/users
- GET /api/images
- POST /api/images
- GET /api/images/:id
- DELETE /api/images/:id
- GET /api/images/:id/preview
- GET /api/users/:userId



Authentication

- POST /login
- POST /logout

Client sends user identifier and password in the request body;
Server sets a Session Id cookie used to recognize the user in the next API calls.



Check authorization

When a new request is received, the server:

- Checks the session id from the cookies of the request
- Retrieve the user from the session list
- Checks if the user has the permission to access the requested resource
- Send response with the resource or an authorization error



Example - POST /api/users (Request)

Request body:

```
{  
  "id": "earthman",  
  "name": "Arthur Dent",  
  "password": "DontPanic"  
}
```



Example - POST /api/users (Response)

Response Status:

- 201: Successful
- 400: Bad Request - Invalid parameters
- 409: Conflict - Username already exist



Example - POST /api/users (Response)


Response Body:

```
{  
  "_links": {  
    "self": "/api/users",  
    "login": "/login",  
    "users": "/api/users"  
  }  
}
```




Example - GET /api/images (Response)

```
{
  "resource": [
    { "id": "8e895d2c-a4c8-48b3-b62b-0f70022dd72d",
      "name": "Sad Marvin",
      "url": "/api/images/8e895d2c-a4c8-48b3-b62b-0f70022dd72d",
      "previewUrl": "/api/images/8e895d2c-a4c8-48b3-b62b-0f70022dd72d/preview"
    }
  ],
  "_links": {...}
}
```



Example - GET /api/images (Response)

```
{  
  "resource": [...],  
  "_links": {  
    "logout": "/logout",  
    "image": "/api/images/{id}",  
    "image_previews": "/api/images/{id}/preview",  
    "self": "/api/images"  
  }  
}
```



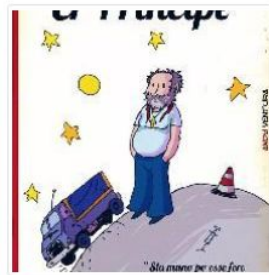
Client

Served statically with Spark
Allows to easily interact with
APIs

Welcome! Logout

My Gallery

+ Add image



Show

Delete



Show

Delete



Show

Delete



Show

Delete



Show

Delete