Django 项目 API 文档

概述

该API允许用户上传图像和通过名称查询已上传的图像。项目在本地服务器上运行，使用Django默认端口（8000）。

基础 URL

```
http://127.0.0.1:8000
```

接口

1. 上传图像

URL	方法	内容类型	参数	响应 (成功)	响应 (错误)	响应 (方法错误)
`/uploadimg/`	POST	`multipart/form-data`	`image` (file): 要上传的图像文件	{"status": "success", "data": {"_id": {"$oid": "664eddb10fc04e70bbc642ec"}, "image_name": "732.jpg", "depthimg_name": "", "uploaded_at": {"$date": 1717185679939}, "processed_image_name": "", "palette_dict": {}, "segments": [], "state": 0, "mongodb_id": "664eddb10fc04e70bbc642ec"}, "message": "Image uploaded successfully."}	`	

2. 通过名称查询图像

URL	方法	参数	响应 (成功)	响应 (警告)
`/findimgbyname/`	GET	`image_name` (string): 要查询的图像名称	{"status": "success", "data": {"_id": {"$oid": "664eddb10fc04e70bbc642ec"}, "image_name": "732.jpg", "depthimg_name": "", "uploaded_at": {"$date": 1716729528648}, "processed_image_name": "1794599054550896640.jpg", "palette_dict": {"0": [127, 127, 127], "1": [0, 0, 200], "2": [0, 200, 0], "3": [0, 238, 238], "4": [238, 0, 238]}, "segments": [{"label": 4, "label_mask_csv": "1794599054550896640_4.csv", "label_attribute": {"pixel_size": [44899, 47, 27], "attribute": [0.10961669921875, 0.00011474609375, 6.591796875e-05], "position": [[0, 501], [579, 449], [634, 463]], "pixel_min_len": 8, "pixel_max_len": 1779, "pixel_aver_len": 598.6666666666666, "pixel_median_len": 9, "pixel_tal_len": 1796, "pixel_aver_width": 20.592335027869744, "pixel_min_width": 0, "pixel_max_width": 46, "pixel_median_width": 22, "distance_img": "1794599054550896640_4_distance.jpg", "skeleton_img": "1794599054550896640_4_skeleton.jpg", "width_points_array": []}}], "state": 1}, "message": "find OK"}	

使用 Java Spring Boot 调用示例

依赖

确保在 `pom.xml` 文件中包含以下依赖：

```xml
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-json</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-http</artifactId>
</dependency>
<dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
</dependency>
<dependency>
```

```
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-rest</artifactId>
</dependency>
```

上传图像示例

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.ContentType;
import org.apache.http.entity.mime.MultipartEntityBuilder;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
import org.apache.http.util.EntityUtils;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.multipart.MultipartFile;

import java.io.IOException;

@RestController
public class ImageUploadController {

    @PostMapping("/uploadImage")
    public String uploadImage(@RequestParam("image") MultipartFile image) throws
IOException {
        String serverUrl = "http://127.0.0.1:8000/uploadimg/";

        try (CloseableHttpClient client = HttpClients.createDefault()) {
            HttpPost post = new HttpPost(serverUrl);
            MultipartEntityBuilder builder = MultipartEntityBuilder.create();
            builder.addBinaryBody("image", image.getInputStream(),
ContentType.MULTIPART_FORM_DATA, image.getOriginalFilename());
            HttpEntity entity = builder.build();
            post.setEntity(entity);
            HttpResponse response = client.execute(post);
            HttpEntity responseEntity = response.getEntity();
            return EntityUtils.toString(responseEntity);
        }
    }
}
```

通过名称查询图像示例

```
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.client.RestTemplate;

@RestController
public class ImageQueryController {
    @GetMapping("/findImageByName")
    public String findImageByName(@RequestParam("image_name") String imageName) {
        String serverUrl = "http://127.0.0.1:8000/findimgbyname/?image_name=" + imageName;
        RestTemplate restTemplate = new RestTemplate();
        ResponseEntity<String> response = restTemplate.getForEntity(serverUrl, String.class);
        return response.getBody();
    }
}
```

该API文档概述了在Django项目中上传和查询图像的接口，并提供了在Java Spring Boot应用中使用这些接口的示例代码。