

10/29

/api/upload_img(待定名称):

用于向后端上传图片

传入为一序列化的JSON对象，结构为：

```
{
  "data": str, # 使用base64编码的图片原始文件,必要参数

  # 可选参数，用于裁剪图片，参数记录裁剪矩形区域左上点和右下点的坐标，坐标以像素为单位，图片左上角坐标为0, 0，水平向右为x轴正，垂直向下为y轴正
  "crop_position": ((int, int), (int, int)),
  # 可选参数，用于在图片过大时缩放图片，在后端会缩放图片最长轴使得其小于等于该参数。该参数若不指定，则后端会配置一个默认的值。该参数若大于后端设置的最大图片尺寸，则会被视为无效参数。
  "max_img_size": int,
  # 可选参数，用于强行配置图片缩放，若该参数存在，则后端会将图片强行缩放到该参数指定的分辨率，若该参数大于后端设置的最大图片尺寸，则会被视为无效参数。
  "max_img_size": (int, int)
}
```

返回为JSON对象，包含错误码和错误信息

```
{
  "errno": 0/1/2/3, #整型，表示错误类型，0表示无异常，具体分配(TODO)
  "msg": "xxx", #字符串，表示错误信息
  "uuid": str #字符串，仅在上传成功时候存在，表示图片的唯一uuid，使用该uuid可以获取到该图片
}
```

10/29

/api/load_img(待定名称):

(可以根据前端需求扩展其他获取图片的形式)

用于从后端下载图片

传入为一序列化的JSON对象，结构为：

```
{
  "uuid": str # 字符串，表示期望获取的图片的uuid
}
```

返回值为JSON对象，包含错误码和错误信息以及返回图片

```
{  
  "errno": 0/1/2/3,    # 整型，表示错误类型，0表示无异常，具体分配(TODO)  
  "msg": "xxx",        # 字符串，表示错误信息  
  "data": str          # 使用base64编码的图片文件,图片编码可能使用webp，待定  
}
```