在项目中后期，我们将使用fastapi+uniapp以前后端分离的方式进行web开发。前后分离主要分为两个方向，网页和后台服务。在网页方向，我们将按照以设计网页，获得页面数据、发送http请求、获得返回数据和页面渲染的顺序进行搭建。而在后台服务方向，我们将根据需求设计数据库，设计接口，实现接口，返回需求所需要的数据。

Web搭建初期我们仅实现最基本的功能，即对图像自动标注。最基础的目标是获取用户上传的图片并返回其图片的标签。在此基础上，我们会将上述基础功能再封装到常用的场景中，比如论坛中发送图片时自动生成标签，或者购物平台上商家上传商品信息时，也能自动生成。下面将就fastapi的特性和技术针对最基础功能进行介绍。

FastAPI 是一个用于构建 API 的现代、快速的 web 框架，使用 Python 3.6+ 并基于标准的 Python 类型提示。

* 快速：可与 NodeJS 和 Go 比肩的极高性能。[最快的 Python web 框架之一](https://fastapi.tiangolo.com/zh/" \l "_11)。
* 高效编码：提高功能开发速度约 200％ 至 300％。
* 更少 bug：减少约 40％ 的人为导致错误。
* 智能：极佳的编辑器支持。处处皆可自动补全，减少调试时间。
* 简单：设计的易于使用和学习，阅读文档的时间更短。
* 简短：使代码重复最小化。通过不同的参数声明实现丰富功能。bug 更少。
* 健壮：生产可用级别的代码。还有自动生成的交互式文档。
* 标准化：基于API 的相关开放标准：[OpenAPI](https://github.com/OAI/OpenAPI-Specification" \t "https://fastapi.tiangolo.com/zh/_blank)和 [JSON Schema](https://json-schema.org/" \t "https://fastapi.tiangolo.com/zh/_blank)。

HTTP请求是前后端分离开发模式中，前后端重要的交互方式。在该功能中，主要采用POST请求，大致格式如下。

from fastapi import FastAPI, Form

app = FastAPI()

@app.post("/login/")

async def login(username: str = Form(), password: str = Form()):

return {"username": username}

且由于我们需要采用表单收集用户上传的图片和标签，所以需同时使用 File 和 Form 定义文件和表单字段。需要注意的是，可在一个路径操作中声明多个 File 与 Form 参数，但不能同时声明要接收 JSON 的 Body 字段。因为此时请求体的编码为 multipart/form-data，不是 application/json。这不是 FastAPI 的问题，而是 HTTP 协议的规定。

对用户上传图片进行识别后，我们将图片和标注一起发送到页面进行显示。对于图片的显示问题，我们将使用七牛云平台。

1. fastapi 结合七牛云，通过直接调七牛云的API
2. 七牛云会给一个30天的域名这样就可以返回一个公网的图片地址
3. 地址存数据库
4. 从数据库中取出地址，送到前端

由于用户也上传了该图片的标注，我们将在后台对标注进行对比，如果标注内容不同

我们将保存图片和标注到训练集中，当训练集到达一定数量再重新训练模型。

最后，我们将基于宝塔面板部署到服务器中。

一、安装Docker管理器

在宝塔面板的软件商店里面安装Docker管理器，以及一个nginx即可，不需要完整的lnmp或者lamp环境。

1. docker安装

通过docker管理器获取官方fastapi镜像， tiangolo/uvicorn-gunicorn-fastapi:latest，官方镜像配置号了uvicorn和gunicorn，完全满足生产环境得需求。

获取镜像后，我们可以在镜像管理中看到下载好得镜像：

三：基于官方镜像生成包含自己工程得镜像

官方镜像只是demo，我们需要将自己得fastapi工程打包成一个镜像进行部署。

1. 上传工程到某一文件夹下，比如/root/chatserver/app:
2. 在app文件夹同一层级下，创建Dockerfile文件
3. 在app文件夹同级目录下，运行命令 docker build -t test .

四：创建容器

点击容器列表，点击创建容器，镜像选择刚刚生成得test镜像，绑定ip为0.0.0.0，端口映射是将容器内项目得端口80，映射到主机得端口8080，注意设置好以后一定要点击加号才会生效。

点击提交后，状态为绿色三角，即成功运行。通过ip打开，可以看到已经正常运行了。