

# BCI 在线处理后台 ReadMe

这是一个基于 Django 的服务器工程, 名为 mysite。本文档将指导你了解项目结构及其功能。

## 项目结构

**mysite:**

- ◆ 这是项目的主文件夹, 包含所有全局路由和配置文件。

## 子模块

**bciDB:**

- ◆ 负责管理服务器中的数据库。
- ◆ 通过编辑 models.py 中的自定义类, 可以修改数据库的数据结构。

**chat:**

- ◆ 一个用于维持 Websocket 长连接的应用。
- ◆ 主要功能处理逻辑位于 consumers.py, 可以修改此文件以调整服务器对在线数据的接收、保存和处理。

**data\_analysis:**

- ◆ 用于离线数据分析的应用。
- ◆ 包含一个独立的文件上传页面, 可在 views.py 中修改对离线数据的处理方式。

**dataset:**

- ◆ 用于数据生成和缓存, 以及在线接收数据流时的保存路径。

**static:**

- ◆ 存放 JavaScript 静态文件的目录。

temp:

- ◆ 用于存放缓存文件的目录。

## 安装和运行

### Docker 部署

1. 构建 Docker 镜像:
  - 在项目根目录下创建一个名为 Dockerfile 的文件，并填写以下内容：

```
FROM python:3.8
ENV PYTHONUNBUFFERED 1
RUN mkdir /code
WORKDIR /code
COPY requirements.txt /code/
RUN pip install -r requirements.txt
COPY . /code/
```
  - 这将设置基础镜像，安装必要的 Python 依赖，并复制当前目录下的项目文件到镜像中的 /code 目录。
2. 构建并运行容器:
  - 使用以下命令构建 Docker 镜像：

```
docker build -t mysite .
```
  - 运行 Docker 容器：

```
docker run -d -p 8000:8000 mysite
```