

# 后端 API 服务

医生人格特质标注系统的后端API服务，基于Go语言和Gin框架开发。

## 技术栈

- **语言:** Go 1.16+
- **框架:** Gin Web Framework
- **数据库:** PostgreSQL
- **ORM:** 原生SQL查询
- **HTTP路由:** Gin Router
- **CORS:** gin-contrib/cors

## 项目结构

```
backend/
├── cmd/                # 命令行工具
│   └── import/         # 数据导入工具
├── controllers/        # API控制器
│   └── physician.go    # 医生相关API
├── db/                 # 数据库相关
│   ├── database.go     # 数据库连接
│   ├── init.sql        # 数据库初始化脚本
│   └── *.sql           # 其他SQL脚本
├── models/             # 数据模型
│   └── models.go       # 数据结构定义
├── routes/             # 路由配置
│   └── routes.go       # API路由设置
├── main.go             # 应用入口点
└── go.mod              # Go模块依赖
```

## 环境变量配置

在运行应用前，需要设置以下环境变量：

```
DB_HOST=localhost      # 数据库主机
DB_PORT=5432           # 数据库端口
DB_USER=your_username  # 数据库用户名
DB_PASSWORD=your_password # 数据库密码
DB_NAME=physicians     # 数据库名称
DB_SSLMODE=disable     # SSL模式
```

## 快速开始

## 1. 安装依赖

```
cd backend
go mod download
```

## 2. 数据库设置

确保PostgreSQL已安装并运行，然后创建数据库：

```
CREATE DATABASE physicians;
```

## 3. 运行应用

```
# 使用环境变量启动
DB_HOST=localhost DB_PORT=5432 DB_USER=your_user DB_PASSWORD=your_pass
DB_NAME=physicians DB_SSLMODE=disable go run main.go

# 或者设置环境变量后运行
export DB_HOST=localhost
export DB_PORT=5432
export DB_USER=your_user
export DB_PASSWORD=your_pass
export DB_NAME=physicians
export DB_SSLMODE=disable
go run main.go
```

应用将在 <http://localhost:8080> 启动。

## API 接口文档

### 基础信息

- **Base URL:** <http://localhost:8080/api>
- **Content-Type:** [application/json](#)

### 医生信息相关

#### 获取医生信息

```
GET /physician/{npi}
```

#### 响应示例:

```
{
  "id": 1,
  "npi": 1234567890,
  "first_name": "张",
  "last_name": "医生",
  "biography_doc": "<p>医生简介HTML内容</p>",
  "education_doc": "<education>大学，医学院，毕业，2000</education>",
  "reviews": [...]
}
```

### 获取任务信息

```
GET /physician/{npi}/task/{taskID}?username={username}
```

### 标注相关

#### 提交人工标注

```
POST /physician/{npi}/task/{taskID}/trait/{trait}/human-annotation
```

#### 获取机器标注

```
GET /physician/{npi}/task/{taskID}/trait/{trait}/machine-annotations
```

#### 提交机器评估

```
POST /physician/{npi}/task/{taskID}/trait/{trait}/machine-evaluation
```

#### 获取进度

```
GET /physician/{npi}/task/{taskID}/trait/{trait}/progress
```

#### 获取历史记录

```
GET /physician/{npi}/task/{taskID}/trait/{trait}/history
```

## 完成特质评估

```
POST /physician/{npi}/task/{taskID}/trait/{trait}/complete
```

## 数据模型

### 主要结构体

- **Physician**: 医生信息
- **Review**: 患者评论
- **Task**: 标注任务
- **HumanAnnotation**: 人工标注
- **ModelAnnotation**: 模型标注
- **MachineAnnotationEvaluation**: 机器标注评估

详细的数据库结构请参考 [../database/README.md](#)。

## 开发特性

### CORS 支持

API支持跨域请求，允许前端应用从不同端口访问。

### 错误处理

所有API接口都包含适当的错误处理和HTTP状态码。

### 日志记录

使用Gin的内置日志中间件记录请求信息。

## 数据导入

项目包含数据导入工具：

```
cd backend/cmd/import
go run main.go
```

该工具可以从JSON文件导入医生和评论数据。

## 部署注意事项

### 生产环境配置

1. 设置 **GIN\_MODE=release**
2. 使用适当的数据库连接池设置

3. 配置反向代理（如Nginx）
4. 启用HTTPS
5. 设置适当的CORS策略

## Docker 部署

可以创建Dockerfile进行容器化部署：

```
FROM golang:1.16-alpine
WORKDIR /app
COPY . .
RUN go build -o main .
EXPOSE 8080
CMD ["/main"]
```

## 贡献指南

1. Fork 项目
2. 创建功能分支
3. 提交更改
4. 推送到分支
5. 创建 Pull Request

## 许可证

本项目采用 MIT 许可证。