**MyApp 桌面应用框架完整文档**

**目录**

1. 使用说明
2. 接口文档
3. 开发指南
4. 部署说明
5. 附录

**1. 使用说明**

**1.1 安装依赖**

bash

复制

# 创建虚拟环境（推荐）

python -m venv venv

source venv/bin/activate # Linux/macOS

venv\Scripts\activate # Windows

# 安装依赖

pip install pyqt6 pymongo python-dotenv pyinstaller

**1.2 运行应用**

bash

复制

# 开发模式

python main.py

# 生产模式（需先配置.env）

MONGO\_URI="mongodb://user:pass@host:port" python main.py

**1.3 基本操作**

1. **导航栏**：左侧固定显示三个功能入口
   * Home：显示欢迎信息
   * Data：展示数据表格
   * Settings：系统设置
2. **内容区**：右侧显示当前选中页面的内容
3. **页面切换**：点击导航按钮即时切换，无刷新效果

**1.4 配置管理**

ini

复制

# .env 配置文件示例

MONGO\_URI=mongodb://localhost:27017/

LOG\_LEVEL=DEBUG

APP\_THEME=dark

| **配置项** | **默认值** | **说明** |
| --- | --- | --- |
| MONGO\_URI | mongodb://localhost:27017 | MongoDB连接字符串 |
| LOG\_LEVEL | INFO | 日志级别(DEBUG/INFO/WARN) |
| LOG\_PATH | ./logs/app.log | 本地日志存储路径 |
| PAGE\_TIMEOUT | 300 | 页面无操作超时(秒) |

**2. 接口文档**

**2.1 核心模块**

**MainWindow (core/window.py)**

python

复制

class MainWindow(QMainWindow):

def \_\_init\_\_(self):

"""初始化主窗口布局"""

def switch\_page(self, page\_name: str):

"""

切换当前显示页面

:param page\_name: 页面标识符 (home/data/settings)

"""

**2.2 控制器**

**MainController (controllers/main\_controller.py)**

python

复制

class MainController(QObject):

def setup\_connections(self):

"""建立所有信号与槽的连接"""

def update\_button\_styles(self):

"""更新导航按钮激活状态样式"""

**2.3 数据服务**

**Database (services/database.py)**

python

复制

class Database:

def \_\_init\_\_(self):

"""单例模式数据库连接"""

def get\_db() -> Database:

"""获取数据库实例"""

**2.4 视图组件**

**DataPage (views/data\_page.py)**

python

复制

class DataPage(QWidget):

def init\_table(self):

"""初始化表格列和表头"""

def populate\_sample\_data(self):

"""填充示例数据到表格"""

**2.5 工具模块**

**Logger (utils/logger.py)**

python

复制

def setup\_logging():

"""初始化日志系统（MongoDB+文件双存储）"""

**Styles (utils/styles.py)**

python

复制

def load\_stylesheet() -> str:

"""加载QSS样式表"""

**3. 开发指南**

**3.1 添加新功能**

**添加新页面步骤：**

1. 在views中创建new\_page.py：

python

复制

class NewPage(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setObjectName("newPage")

self.init\_ui()

def init\_ui(self):

# 页面布局实现

1. 在core/window.py注册页面：

python

复制

self.pages["new"] = NewPage()

1. 添加导航按钮：

python

复制

# 在MainWindow的\_\_init\_\_方法中

self.btn\_new = QPushButton("New Page")

self.btn\_new.setObjectName("navButton")

sidebar\_layout.insertWidget(2, self.btn\_new) # 插入到第三个位置

1. 在控制器中连接信号：

python

复制

# controllers/main\_controller.py

self.window.btn\_new.clicked.connect(

lambda: self.window.switch\_page("new")

)

**3.2 数据操作示例**

python

复制

# services/data\_service.py

class DataService:

@staticmethod

def insert\_log(message: str, level: str = "INFO"):

"""

插入日志到数据库

:param message: 日志内容

:param level: 日志级别

"""

db = get\_db()

db.logs.insert\_one({

"timestamp": datetime.now(),

"level": level,

"message": message

})

**3.3 样式定制**

通过修改utils/styles.py实现：

python

复制

def load\_stylesheet():

return """

/\* 自定义按钮 \*/

QPushButton#specialButton {

background: qlineargradient(x1:0, y1:0, x2:1, y2:1,

stop:0 #ff9966, stop:1 #ff5e62);

color: white;

border-radius: 15px;

}

"""

**3.4 代码规范**

1. **模块化设计**：
   * 视图：只包含UI组件
   * 控制器：处理业务逻辑
   * 服务：数据存取操作
2. **命名约定**：
   * 页面组件：[功能]Page
   * 数据库集合：复数形式（如users）
   * 信号方法：on\_[控件]\_[事件]
3. **样式管理**：
   * 使用QSS代替硬编码样式
   * 通过objectName限定样式作用域
   * 颜色值使用HEX格式

**4. 部署说明**

**4.1 打包发布**

使用PyInstaller生成可执行文件：

bash

复制

pyinstaller --name MyApp \

--icon assets/icon.ico \

--add-data "assets;assets" \

--windowed \

--hidden-import=pymongo \

main.py

**4.2 跨平台支持**

| **平台** | **注意事项** |
| --- | --- |
| Windows | 需安装VC++运行库 |
| macOS | 需要签名公证 |
| Linux | 注意字体包依赖 |

**4.3 数据库部署**

1. 安装MongoDB社区版
2. 创建数据目录：

bash

复制

mongod --dbpath /data/db --port 27017

1. 设置用户权限：

javascript

复制

use admin

db.createUser({

user: "admin",

pwd: "securepassword",

roles: ["root"]

})

**5. 附录**

**5.1 依赖列表**

| **包名称** | **版本** | **用途** |
| --- | --- | --- |
| PyQt6 | ≥6.4.0 | GUI框架 |
| pymongo | ≥4.3.3 | MongoDB驱动 |
| python-dotenv | ≥0.21.0 | 配置管理 |

**5.2 示例配置**

yaml

复制

# config.yml（可选）

logging:

file\_max\_size: 10MB

keep\_days: 7

database:

pool\_size: 5

timeout: 5000

**5.3 常见问题**

**Q：无法连接MongoDB？**

* 检查防火墙设置
* 验证连接字符串格式：  
  mongodb://[username:password@]host[:port][/[database][?options]]

**Q：样式不生效？**

1. 确认objectName设置正确
2. 检查QSS选择器语法
3. 使用qsslint工具验证样式表

**Q：打包后资源丢失？**

* 确保assets目录包含在打包参数中
* 使用相对路径访问资源：

python

复制

base\_path = getattr(sys, '\_MEIPASS', os.path.dirname(os.path.abspath(\_\_file\_\_)))