# 软件开发文档

目录

[软件开发文档 1](#_Toc28956091)

[一、整体架构 1](#_Toc28956092)

[1、app/ 1](#_Toc28956093)

[2、docs/ 1](#_Toc28956094)

[3、images/ 2](#_Toc28956095)

[4、lib/ 2](#_Toc28956096)

[5、qss/ 2](#_Toc28956097)

[二、新增不同类别的Widget步骤 2](#_Toc28956098)

[三、Widget Id 2](#_Toc28956099)

[1、what 2](#_Toc28956100)

[2、how 2](#_Toc28956101)

[四、Widget 2](#_Toc28956102)

[1、创建widget 2](#_Toc28956103)

[2、widget的必要信号连接 3](#_Toc28956104)

[3、timeline 3](#_Toc28956105)

## 一、整体架构

### 1、app/

Info (app/info.py) 类存放配置信息

Kernel (app/kernel.py) 类存放数据信息

Func (app/func.py) 类存放通用函数

其余文件夹按模块划分

### 2、**docs**/

存放文档

### 3、images/

按类别存放图片

### 4、lib/

存放项目通用模块

### 5、qss/

存放qss文件

## 二、新增不同类别的Widget步骤

1、继承lib.TabItemWidget类，完成对应模块

2、 在Info中增加改类别xxx = int\_value，并新增WidgetType[int\_value] = “type\_name”

3、在Kernel中修改WidgetTypeCount和WidgetNameCount

4、在Func的createWidget中导入相应类

5、在images/widgets中放入图标，格式为png

## 三、Widget Id

### 1、what

Widget id是用来和相应widget进行绑定的关键字，通过Func.generateWidgetId (widget\_type)来生成。

### 2、how

在timeline控件中，从上方的icon bar拖入下方区域便会生成对于widget id并创建widget。

当发生复制行为时，生成对于widget id并复制控件（控件内部提供复制函数）

当发生引用行为时，生成对于widget id并引用控件（通过将不同的widget id指向同一widget实现）

## 四、Widget

### 1、创建widget

调用Func类的createWidget

### 2、widget的必要信号连接

调用Func类的linkWidgetSignals，这个函数实际是调用了Psy（主窗口类）的linkWidgetSignals

### 3、timeline

Timeline首先继承lib.TabItemWidget，包括icon bar和timeline area

1. Icon bar是用来选择icon并可以拖入timeline area的控件
2. Timeline area使用了graphics中的功能:

分两层，为graphics view套着graphics scene

对graphics scene进行布局是通过grid layout进行管理，为使用layout，timeline item继承了graphics layout item。