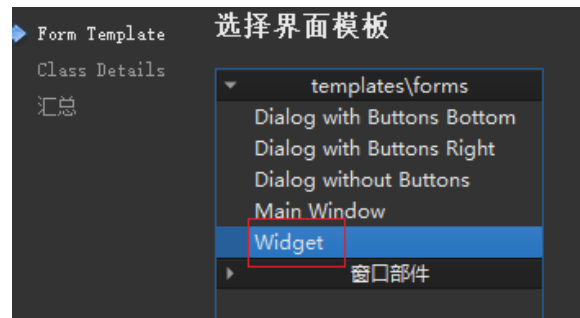
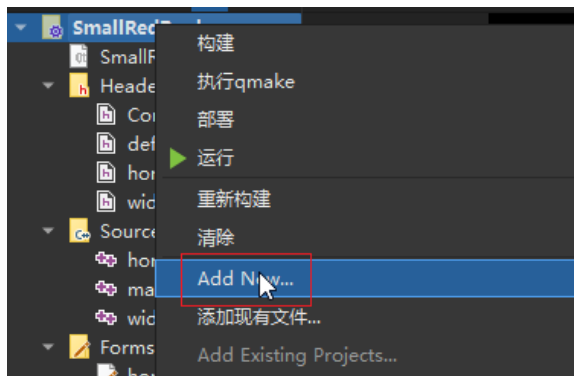


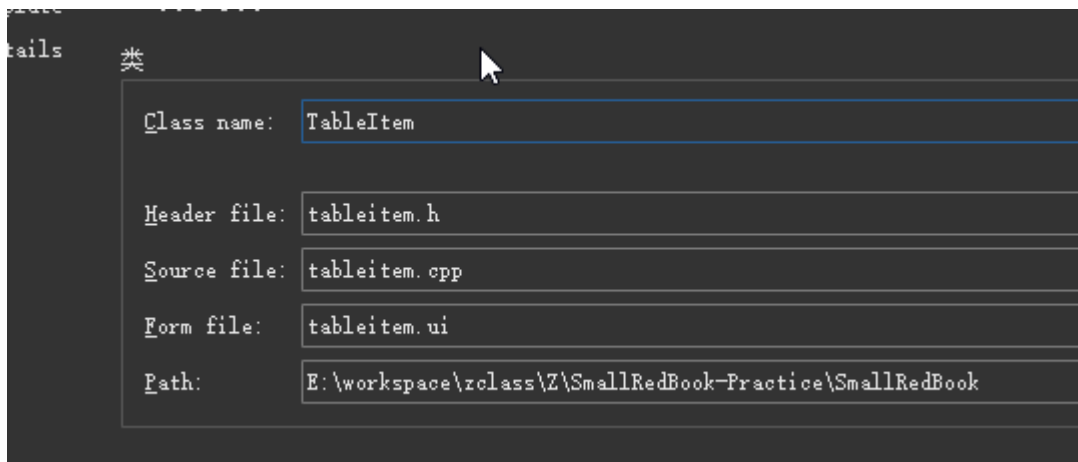


# 202506-电子信息工程生产实习-day4

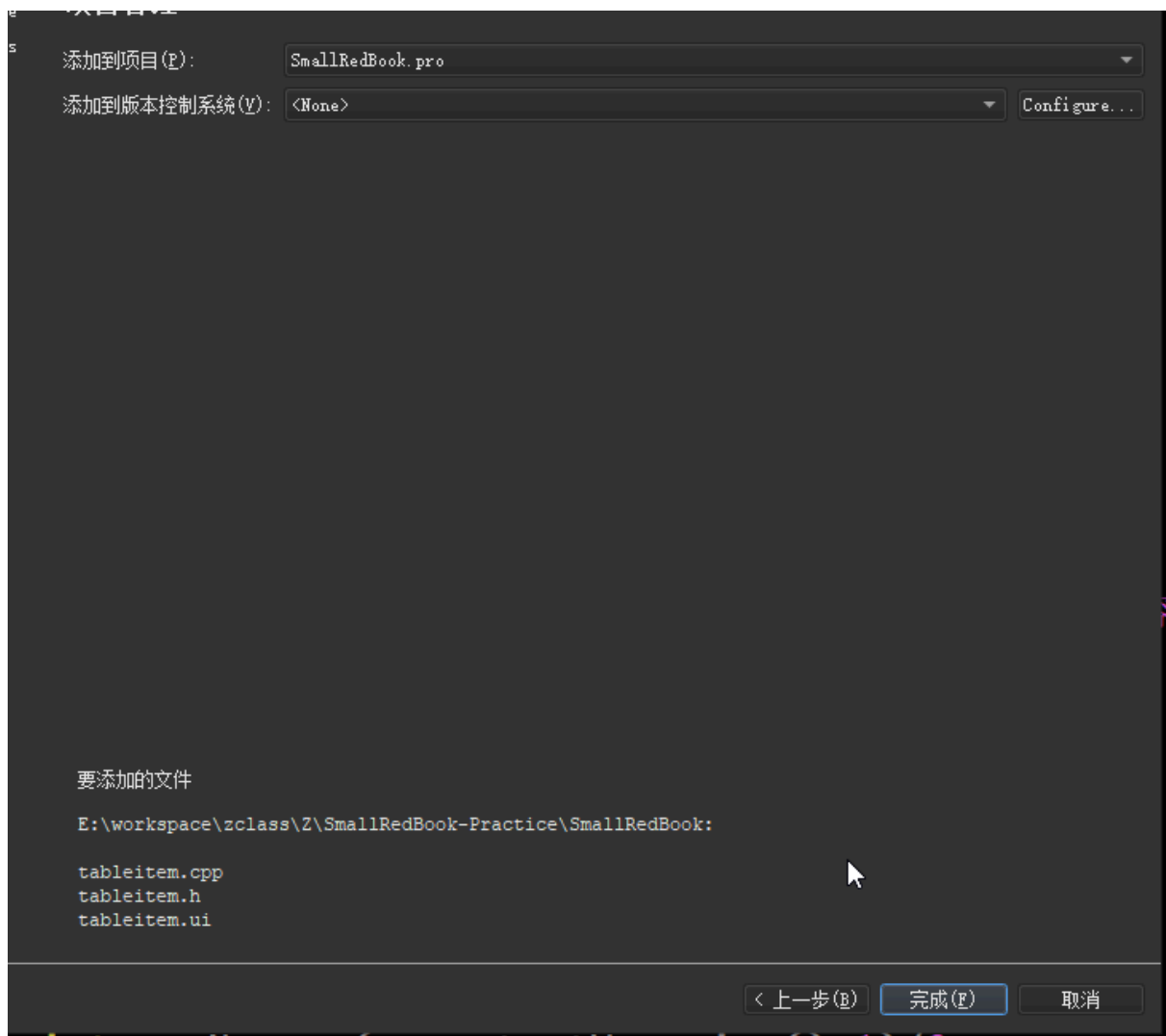
## 1. 创建表格项TableWidgetItem



命名为TableWidgetItem



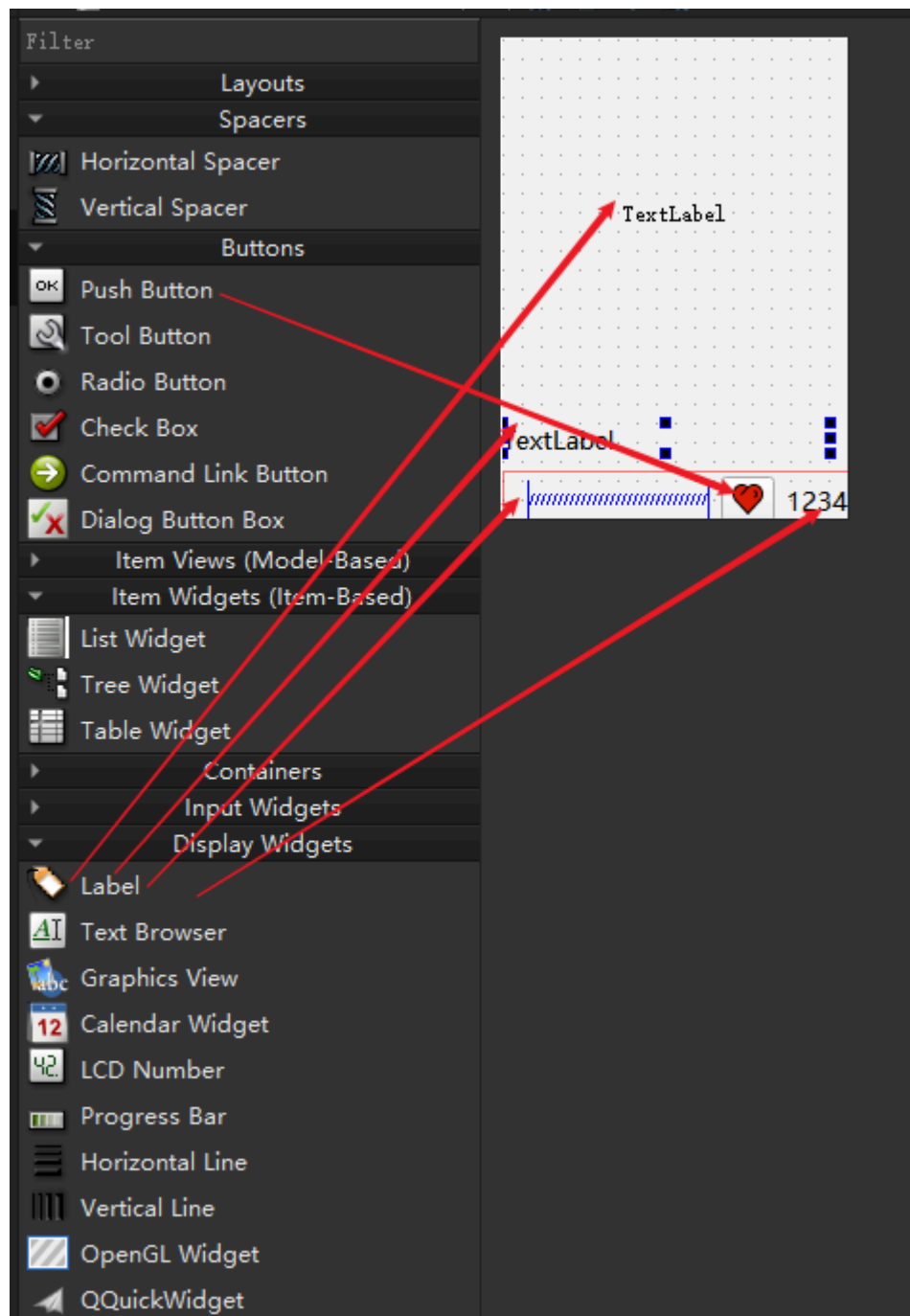
最后点【完成】



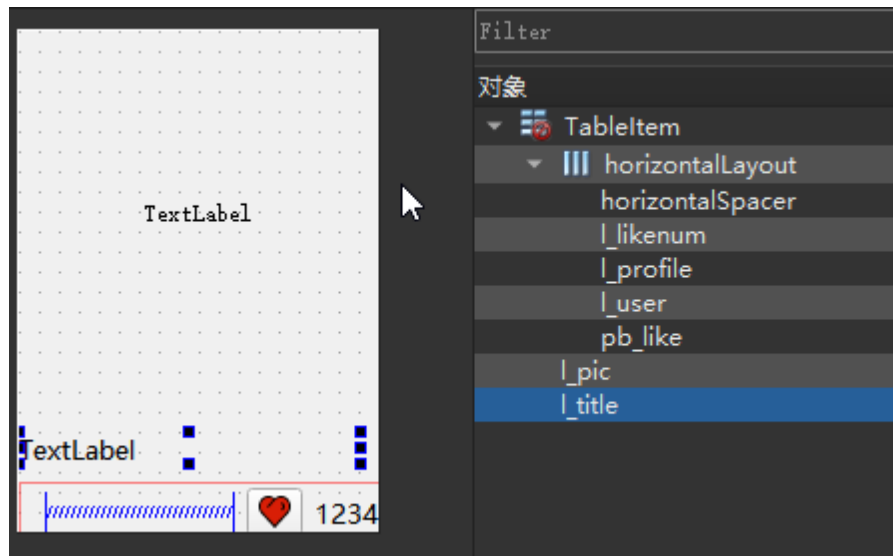
- 设计表格项目界面



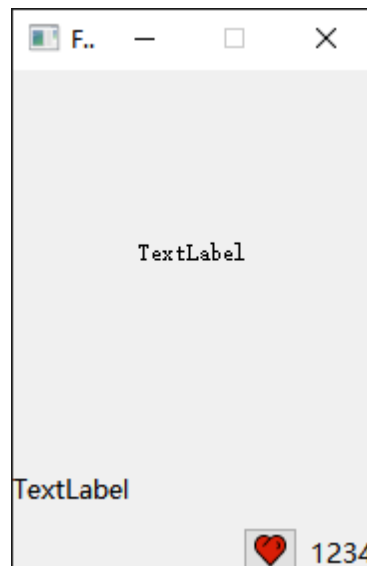
参考宽度高度：180,250，中间可以水平分割器撑开组件。



修改对象名如图



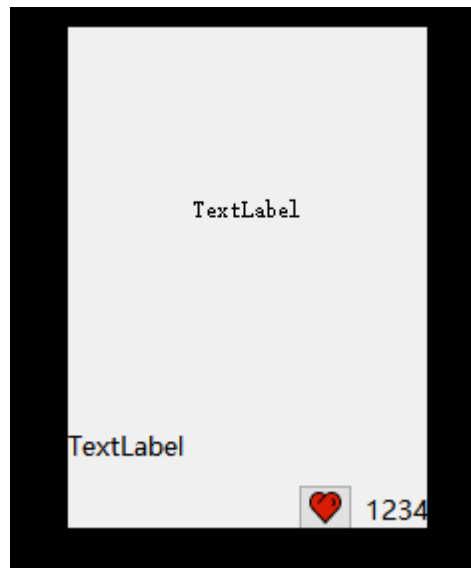
先进行测试，查看窗口



去除标题栏和边框。注意，如果是多个样式的话，则用位或 | 连接

```
this->setWindowFlags(Qt::CustomizeWindowHint | Qt::FramelessWindowHint);
```

如图：



设置各种初始内容，帖子图片、标题、头像图片、用户名、是否喜欢、点赞数量。

构造函数增加参数，将item内容传递过来

```
1c:
explicit TableItem(const Content& content, QWidget *parent =
```

```
//设置图片，默认显示第0张，如 0-0, 1-0, 2-0...., 注意保持宽高比
ui->l_pic->setPixmap(QPixmap(CONTENT_PIC(content.id)).scaled(ui->l_pic->s

//设置标题
ui->l_title->setText(content.title);

//设置头像id
ui->l_profile->setStyleSheet(PROFILE_STYLE(content.id));

//设置用户名
ui->l_user->setText(content.user);

//设置是否喜欢, true : 点赞, false : 未点赞
if(content.like){
    ui->pb_like->setIcon(LIKE_ICON);
}else{
    ui->pb_like->setIcon(UNLIKE_ICON);
}
//设置按钮为平的
```

```

ui->pb_like->setFlat(true);
//设置图标大小为16*16
ui->pb_like->setIconSize(QSize(16,16));

//点赞数量
ui->l_likenum->setText(QString::number(content.like_num));

```

帖子id、是否点赞、点赞数量三个变量需要存储起来。方便后面去用到。  
类中增加属性，并在构造函数中存储

```

int m_id;
bool m_isLike;
int m_likeNum;

```

```

m_id = content.id;
m_isLike = content.like;
m_likeNum = content.like_num;

```

为点赞按钮增加槽函数



点赞 和 取消点赞的 标志通过 m\_isLike 来判断。

```

void TableItem::slots_setLike(){
    isLike = !isLike;

    if(m_isLike){
        ui->pb_like->setIcon(LIKE_ICON);
        ui->l_likenum->setText(QString::number(++m_likeNum));
    }else{
        ui->pb_like->setIcon(UNLIKE_ICON);
        ui->l_likenum->setText(QString::number(--m_likeNum));
    }

    //告诉主页，是否取消点赞了
    emit signals_isLike(id,isLike );
}

```

其中 signals\_isLike 信号，用于通知Home中的m\_contentVec 更改数据的信号。参数2个：帖子id和是否点赞。

```

signals:
    void signals_isLike(int id,bool islike);

```

其绑定的槽函数，后面会讲述。

## 2. 初始化主页-发现

在上一步解析了json内容后，根据 `m_contentVec` 的内容进行加载数据。

增加函数： `void initFindContent ()`

1. 对 `m_contentVec` 进行判空。
2. 加载之前清空表格数据。
3. 计算并设置行数。
4. 设置单元格组件，
  - a. 创建 `TableItem` 对象（180x250），并指定Content内容为参数。
  - b. 通过 `setCellWidget` 设置单元格组件，其中行列都是从0开始的，行：除以2，列：取余2
  - c. 设定表格的行高为250



代码如下：

```
void Home::initFindContent(){
    //1.判空
    if(m_contentVec.size()<=0){
        QMessageBox::warning(this,"WARNING","暂无内容");
        return ;
    }

    //清空表格tablewidget
    ui->tw_content->clear();

    //计算行数
    int rowNum = (m_contentVec.size()+1)/2;
    ui->tw_content->setRowCount(rowNum);

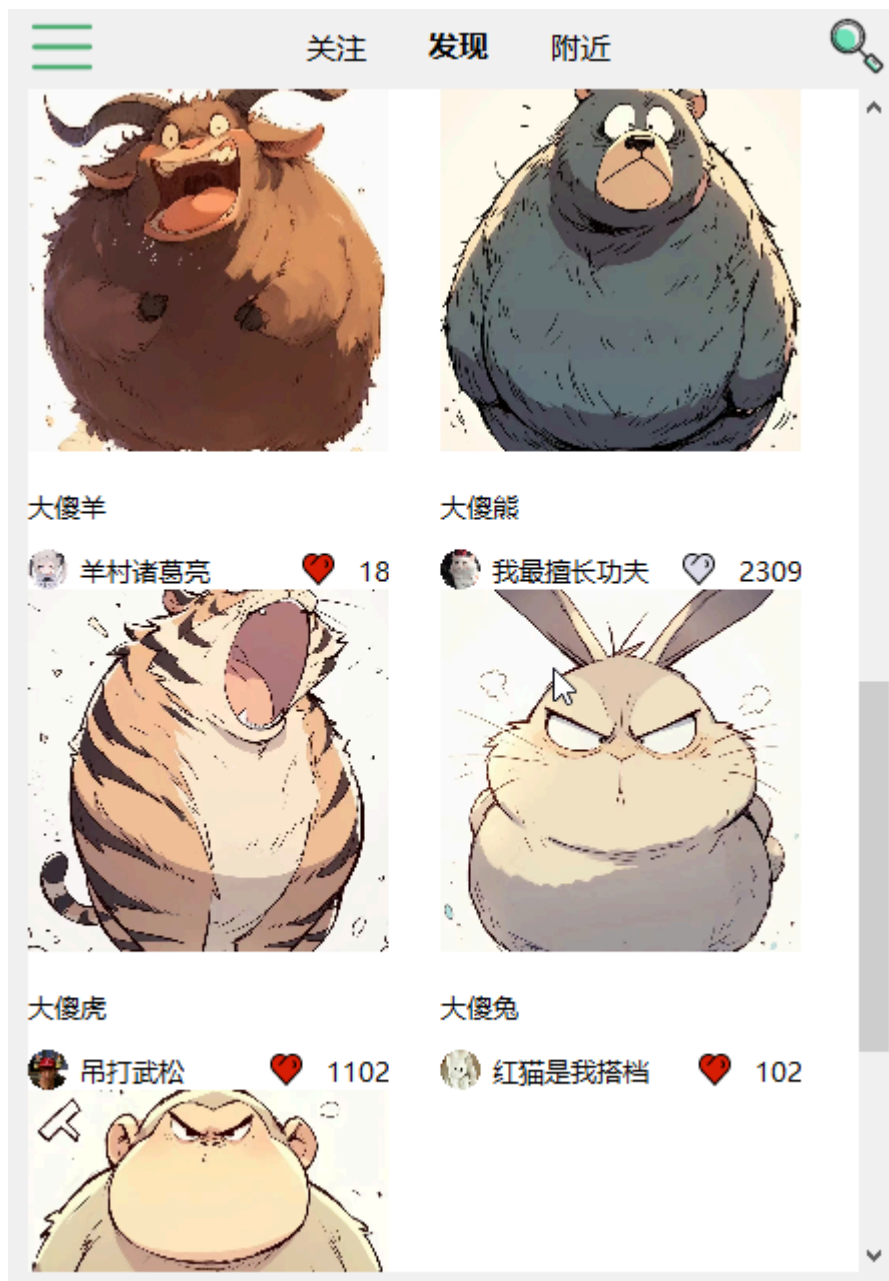
    //遍历json配置的内容
    int size = m_contentVec.size();

    for( int i=0;i<size;++i){
        //item 大小： 180,250
        TableItem* ptableItem= new TableItem(m_contentVec[i]);
        ui->tw_content->setCellWidget(i/2,i%2,ptableItem);
        ui->tw_content->setRowHeight(i/2,250);
    }
}
```

在构造函数中，增加初始化发现内容

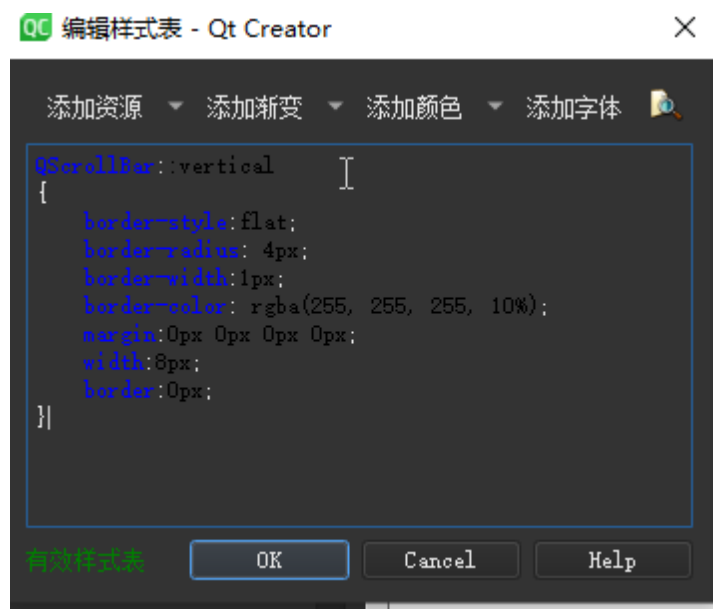
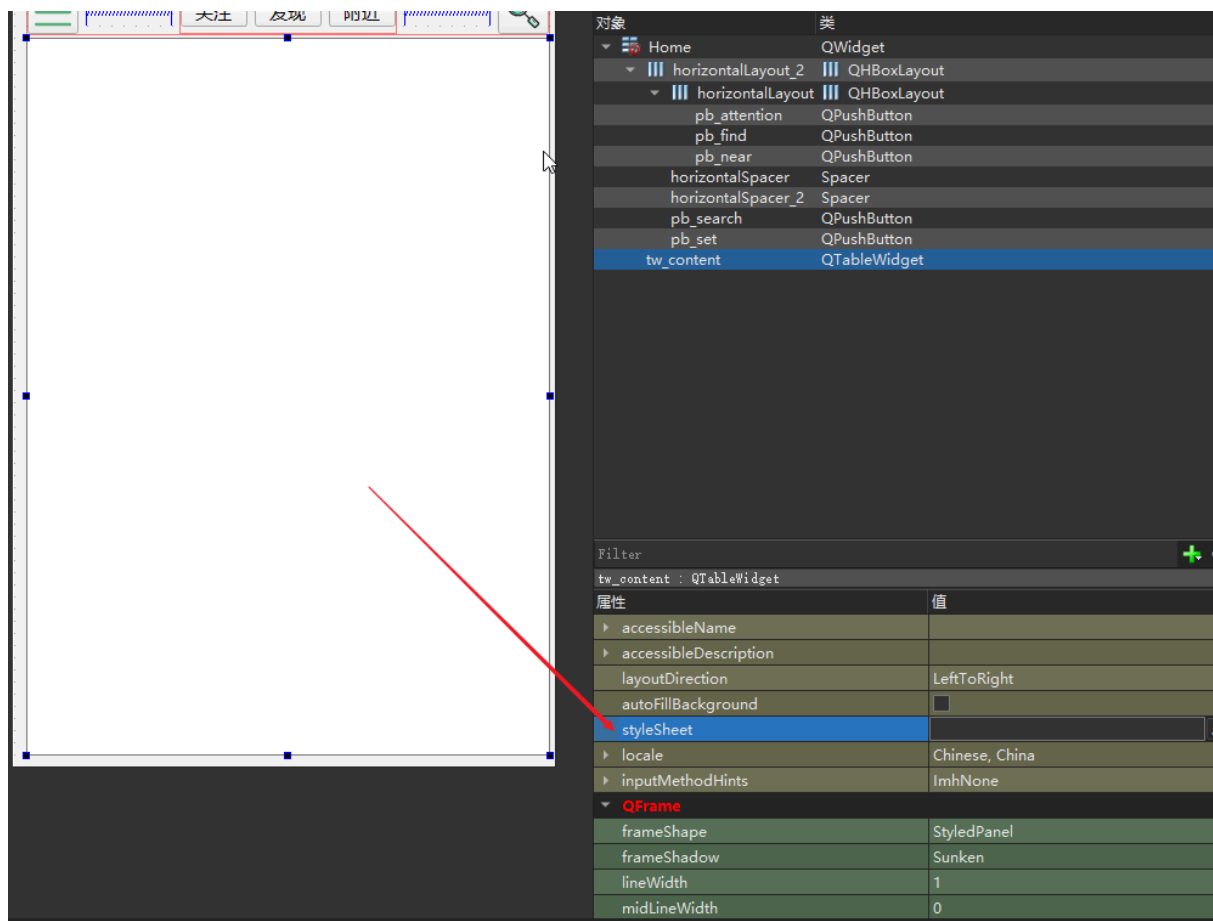
```
//默认初始化发现内容
initFindContent();
```

运行结果如图：



发现其右面的滚动条太粗了，一点都不美观。

对 QTableWidget 组件加样式表，除了用代码加样式表，还可以通过属性直接加。



样式表代码如下：

```
QScrollBar::vertical
{
    border-style: flat;
```

```
border-radius: 4px;  
border-width:1px;  
border-color: rgba(255, 255, 255, 10%);  
margin:0px 0px 0px 0px;  
width:8px;  
border:0px;  
}
```

修改后，效果如下图：

