Q 搜索

Qt设计工作室手册 > 加载占位符数据

## 加载占位符数据

Qt Design Studio支持视图、模型和委托,因此当您添加网格视图、列表视图或路径视图组件时,会自动添加ListModel和委托组件。

但是,缺少应用程序的上下文会带来挑战。C++中定义的特定模型是最明显的情况。通常,上下文缺少简单属性,这些属性要么在C++中定义,要么在其他组件文件中定义。一个典型的示例是使用其父级属性的组件,例如。parent.width

## 使用虚拟模型

Qt设计工作室手册3.8.0 Topics >

例如,以下代码段描述了包含 ListView 的文件 example.qml,该列表视图又指定了C++模型:

```
ListView {
    model: dataModel
    delegate: ContactDelegate {
        name: name
    }
}
```

在项目的根目录中创建一个名为dummydata的目录,以便它不会部署到设备。在目录中,创建一个与以下值同名的文件(.qml):dummydatamodel

```
qml/exampleapp/example.qml
dummydata/dataModel.qml
```

然后创建包含虚拟数据的 dataModel.qml 文件:

```
import QtQuick 2.0
```



```
}
ListElement {
    name: "Bella"
}
ListElement {
    name: "Corinna"
}
```

## 创建虚拟上下文

以下示例提供了一个常见模式:

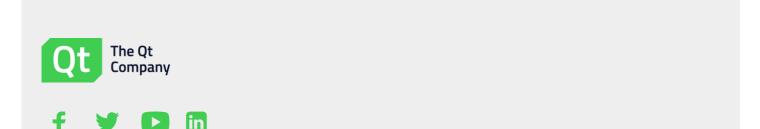
```
Item {
    width: parent.width
    height: parent.height
}
```

这适用于应用程序,但2D视图显示的是零大小的组件。打开的文件的父文件不存在,因为缺少上下文。为了绕过缺失的上下文,引入了*虚拟上下文*的概念。如果在目录中放置与应用程序同名的文件(此处为example.qml),则可以伪造父上下文:dummydata/context

```
import QtQuick 2.0
import QmlDesigner 1.0

DummyContextObject {
    parent: Item {
        width: 640
        height: 300
    }
}
```

〈模拟复杂体验 模拟应用程序逻辑〉





公司

关于我们

投资者

编辑部

职业

办公地点

发牌

条款和条件

开源

常见问题

支持 对于客户

支持服务

专业服务

合作伙伴

训练

支持中心

下载

Qt登录

联系我们

客户成功案例

社区

为Qt做贡献

论坛

维基

下载

市场

反馈 登录