

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，**列表模型**和委托组件会自动添加。

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

如果在 **2D** 视图中打开参照C++模型的文件，则其中看不到任何内容。如果模型中的数据是从互联网上获取的，则无法控制它。为了获得可靠的数据，引入了**虚拟数据**。

例如，下面的代码段描述了包含 **ListView** 的文件 `example.qml`，该文件又指定了一个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 `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模型.qml 文件：

```
import QtQuick 2.0

ListModel {
    ListElement {
        name: "Ariane"
```

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

创建虚拟上下文

下面的示例演示了一个通用模式：

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

这适用于应用程序，但 2D 视图显示零大小的组件。打开的文件的父级不存在，因为缺少上下文。为了解决缺少的上下文，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 ▶



Contact Us



Investors
Newsroom
Careers
Office Locations

Open Source
FAQ

Support

Support Services
Professional Services
Partners
Training

For Customers

Support Center
Downloads
Qt Login
Contact Us
Customer Success

Community

Contribute to Qt
Forum
Wiki
Downloads
Marketplace