

# 模拟应用程序逻辑

您可以使用 JavaScript 来模拟应用程序逻辑，使您的 UI 栩栩如生。

您将需要以下文件：

- 将指定 UI 的 API 的组件文件（.qml）
- 为 UI 生成模拟数据的脚本文件。有关使用脚本的更多信息，请参阅[集成 QML 和脚本](#)。
- 模块定义文件（*qmlDir*），用于声明您在 UI 文件中指定的组件（QML 类型）。有关更多信息，请参见[模块定义 qmlDir 文件](#)。

在这里，您将基于将注册为单例类型的 `QObject` 类创建一个组件。这允许在 UI 中使用全局属性值。

要创建必要的文件：

- 在文件资源管理器中，为项目文件夹中的导入文件夹（例如“数据”）中的模拟数据创建一个新文件夹。

**注意：** 请确保将数据文件夹名称大写，因为稍后需要将其作为组件导入，并且组件名称必须大写。

**注意：** 如果将此文件夹放在项目中的其他位置，则需要将路径添加到导入列表中。为此，在 Qt 设计工作室中，打开项目文件（.qmlproject）以将路径添加到传递给 QML 运行时的插件目录列表中。例如，如果将 *Data* 文件夹放在项目根目录中另一个名为 *后端* 的文件夹中，则可以添加以下内容：

```
importPaths: [ "imports", "backend" ]
```

- 选择“文件>新文件>Qt 快速文件”>Qt 快速文件>选择添加将指定 UI API 的 Qt 快速文件。
- 按照向导的说明在数据文件夹中创建 Qt 快速文件。在这些说明中，该文件称为 *值.qml*。

**注意：** 确保将文件名大写，因为它将成为自定义组件。

- 选择“文件>新文件”>“>>”选择，以创建为 UI 生成模拟数据的 JavaScript 文件。
- 按照向导的说明在“数据”文件夹中创建 JavaScript 文件。在这些说明中，该文件称为 *模拟.js*。
- 删除脚本文件中的模板文本并保存该文件。
- 在文本编辑器（如记事本）中，创建一个名为 *qmlDir* 的模块定义文件，其中包含以下内容，并将其放在 *数据* 文件夹中。

singleton-values-1.0-values.qml

8. 在“代码”视图中打开“值”.qml 进行编辑。
9. 将以下代码添加到文件顶部，以注册将用于将全局属性公开为单一实例类型的 `QObject` 派生类：

```
pragma Singleton
```

10. 添加以下 import 语句以导入 `模拟.js` 文件以使用其提供的功能：

```
import "simulation.js" as JS
```

11. 将默认的 `Item` 声明替换为以下代码，以创建一个 `QObject` 派生类，该类将列出要模拟的全局属性及其默认值：

```
QtObject {
    id: values

    // property values to simulate
    property int name1: 5
    property string name2: "foo"
    property real name3: 2.5
}
```


12. 添加以下代码以使用 `Timer` 组件为属性指定值范围：

```
property Timer name1Timer: Timer{
    running: true
    repeat: true
    onTriggered: JS.name1Timer()
    interval: 10
}
```

This will execute the function defined for every 10 ms. Within your javascript functions you can perform the necessary actions to simulate the behavior you need. Review [Importing JavaScript Resources in QML](#) for more details.

**Note:** You must add the JavaScript method to the JavaScript file. You have the option of adding this JavaScript code directly within the handler as well.

13. Open the .ui.qml file of the Component that will use the simulated data and add the following code to the top of the file in order to import the Data folder as a QML module:

14. Returning to the **2D** view, locate the property that should be bound to the simulated values. Select  and **Set Binding** for the property and enter the simulated Value property. For example, you would set the following expression to bind to the example property:name1

Values.name1

< Loading Placeholder Data

Simulating Dynamic Systems >



Contact Us

Company

About Us  
Investors  
Newsroom  
Careers  
Office Locations

Support

Support Services  
Professional Services  
Partners  
Training

Community

Contribute to Qt  
Forum

Licensing

Terms & Conditions  
Open Source  
FAQ

For Customers

Support Center  
Downloads  
Qt Login  
Contact Us  
Customer Success



Marketplace

© 2022 The Qt Company

[Feedback](#) [Sign In](#)