

Sciter WebView

Sciter WebView 是 Sciter 行为组件。它可用于将 Web 内容无缝合并到 Sciter 应用程序的 UI 中。

目标是实现控制器（作为 Sciter 的行为），以便 Sciter 能够加载任意网页并与之交互。 `<webview>`

演示



特征

- Windows - 使用 WebView2
- macOS - 使用 WKWebView
- Linux - 使用 WebKitGTK2
- 支持双向 JavaScript 绑定（从 SciterJS 调用 JavaScript，以及从 JavaScript 调用 SciterJS）。
- 支持警报、确认和提示对话框
- 支持导航加载、后退、前进、重新加载、停止

平台

- Windows - i32、i64 和 arm64
- macOS - i64 和 arm64
- Linux 操作系统 - i64

用法

外部行为：

1. 使用 'build' 文件夹中的项目构建 dll/dylib/so 文件；
2. 将 dll/dylib/so 文件放入 sciter 应用程序运行文件夹；
3. 使用 `<webview>` 并在 css 中定义：`webview {behavior: webview library (sciter-webview) }`

内部行为：

1. 将文件夹 [source](#) 中的所有文件复制到 sciter-js-sdk [include/behaviors](#) 中
2. 编译应用程序项目；

3. 使用 <webview> 并在 css 中定义：webview {behavior: webview}

JS API

方法

element.webview.loadUrl

```
element.webview.loadUrl(url:string);
```

导航到 URL

element.webview.loadHtml

```
element.webview.loadHtml(html:string);
```

加载 HTML

element.webview.reload

```
element.webview.reload();
```

重新加载上次加载的文档。

element.webview.goBack返回

```
element.webview.goBack();
```

向后导航。

element.webview.goForward

```
element.webview.goForward();
```

向前导航。

element.webview.evaluateJavaScript

```
element.webview.evaluateJavaScript(jssrc:string);
```

在加载的文档的上下文中评估 *jssrc*。

性能

element.webview.currentSrc

```
const url = element.webview.currentSrc;
```

read-only, string, 报告加载文档的当前 URL

元素.webview.src

```
const url = element.webview.src;
```

read-write、string、要导航的 URL

事件

“webview 就绪”

```
element.on("webview-ready", function() {...});
```

WebView 已初始化并准备好接受方法调用。

“webview-不可用”

```
element.on("webview-unavailable", function() {...});
```

WebView 初始化失败。系统出于某种原因拒绝了 Web 视图创建。

“webview-will-navigate”

```
element.on("webview-will-navigate", function(evt) { const url = evt.data; ...});
```

WebView 将导航到 *URL*。

“webview-did-navigate”

```
element.on("webview-did-navigate", function() {...});
```

WebView 已成功导航到请求的 URL。

“webview-navigate-failure”

```
element.on("webview-navigate-failure", function() {...});
```

由于某些原因，WebView 无法导航到请求的 URL。

“webview-title-change”

```
element.on("webview-title-change", function(evt) { const title = evt.data; ...});
```

WebView 会收到*标题为的新文档*。