

# 错误排查

**MontageJS**应用没有按照预期效果在浏览器里面显示。

如果你在浏览器中测试一个MontageJS应用的时候，发现应用没有显示为预期效果，你可以打开JavaScript控制台查看错误和警告信息。大多数的浏览器包含JavaScript调试工具、控制台或相关的工具来帮助调试web应用。在新版的Chrome和Safari已经内置了这些开发工具。你可以使用浏览器的 `console.log()` 函数向控制台输出信息。

- 打开Chrome控制台的方法是，选择 视图 > 开发者 > JavaScript控制台。
- 打开Safari控制台方法是，首先开启开发者菜单（Safari偏好 > 高级 > 在菜单栏显示开发者菜单），然后选择开发者 > 显示控制台错误信息。

运行MontageJS应用的时候发生“Warning: Element xxx not found in template”错误信息。

这样的错误发生一般是由于DOM元素的ID和对象的 `element` 属性值不匹配。例如，在以下的代码片段中，在对象( `"loginButton"` ) 中引用的DOM元素ID ( `"loginBtn"` ) 与HTML中div元素不匹配。

```
<script type="text/montage-serialization">
{
  "loginButton": {
    "name": "Button",
    "module": "montage/ui/button.reel",
    "properties": {
      "element": { "#": "loginButton" }
    }
  }
}
</script>
<body>
```

```
<div id="loginBtn"/>
</body>
```

在运行时，以上代码会在控制台输出以下错误信息：

```
Warning: Element "#loginButton" not found in template
http://localhost:8081/examples/myapp/mycomponent.reel/mycomponent.h
tml
```

## "Object "xxx" not found at "yyy""错误信息。

如果你设置对象的 `"name"` 属性为一个不合法的值，MontageJS会输入以上错误信息。例如，下面的示例代码会产生这个错误，因为 `"Button"` 被错误拼写为 `"Buttonn"`。

```
{
  "loginButton": {
    "name": "Buttonn",
    "module": "montage/ui/button.reel",
    "properties": {
      "element": {"#": "loginButton"}
    }
  }
}
</script>
```

在控制台中有如下运行时错误：

```
Object "Buttonn" not found at "montage/ui/button.reel" referenced f
rom http://localhost:8081/examples/buttonerror/.
```

## "Can't XHR "http://localhost: ..."错误信息。

如果在组件HTML模板中使用无效的模块ID，在控制台中就有"Can't XHR module-id" 404 错误信息。例如，在下面的代码中定义了一个不存在的组件，但是模块ID错误信息。

错误信息。例如，在下面的代码中定义了一个 `textfield` 组件，但是模块 ID 错误拼写成 `"montage/ui/textfld.reel"`。

```
{
  "emailInput": {
    "name": "Textfield",
    "module": "montage/ui/textfld.reel",
    "properties": {
      "element": { "#": "email" }
    }
  },
}
```

在Chrome控制台中，会出现如下错误信息：

```
GET http://localhost:8081/ui/textfield.reel/textfld.js 404 (Not Found) browser.js:136
Can't XHR "http://localhost:8081/ui/textfield.reel/textfield.js"
```

## "unexpected comma"错误信息。

在对象结尾的逗号在JSON序列化中是不被允许的。JSON序列化必须是合法格式，这样MontageJS才能够成功解析。JSON对象定义的末尾或者是数组的最后一个元素后如果有逗号将产生运行时错误。在下面的例子中，`readyState`属性后面的逗号将产生解析错误：

```
"anObject": {
  "id": "123asd",
  "colors": ["red", "green", "blue"],
  "readystate": false,
}
```

在下面代码中，`"passwordInput"` JSON 对象末尾的逗号也会产生 `"unexpected comma"` 运行时错误。

```
<script type="text/montage-serialization">
{
```

```
"emailInput": {
  "name": "Textfield",
  "module": "montage/ui/textfield.reel",
  "properties": {
    "element": {"#": "email"}
  }
},
"passwordInput": {
  "name": "Textfield",
  "module": "montage/ui/textfield.reel",
  "properties": {
    "element": {"#": "password"}
  }
},
}
</script>
```

Syntax error at line 16 from <http://localhost:8081/examples/errors/>:

```
  },
Unexpected comma.
1
2 {
3   "emailInput": {
4     "name": "Textfield",
5     "module": "montage/ui/textfield.reel",
6     "properties": {
7       "element": {"#": "email"}
8     }
9   },
10  "passwordInput": {
11    "name": "Textfield",
12    "module": "montage/ui/textfield.reel",
13    "properties": {
14      "element": {"#": "password"}
```

```
15         }  
>>>16     },  
17 }  
18
```

移除第16行末尾的逗号可以修复这个错误。