链滴

使用浏览器原生 API 读写本地文件

作者：Vanessa

原文链接：https://ld246.com/article/1583739236421

来源网站：链滴

2020-03-09

简介

新的文件系统 API 允许 Web 应用读取或修改用户设备中的文件或文件夹。他能让开发者简单的构建个和用户本地文件交互的强大的 Web 应用，如 IDE，照片和视频编辑，文本编辑等。当用户授权给个 Web 应用后，这个 API 就允许该 Web 应用读取或者修改用户设备上的文件。

该 API 不属于 W3C 官方或 WHATWG 规范，目前状态如下：

Specification	Status	Comment
File and Directory Entries API	Draft	raft of proposed API

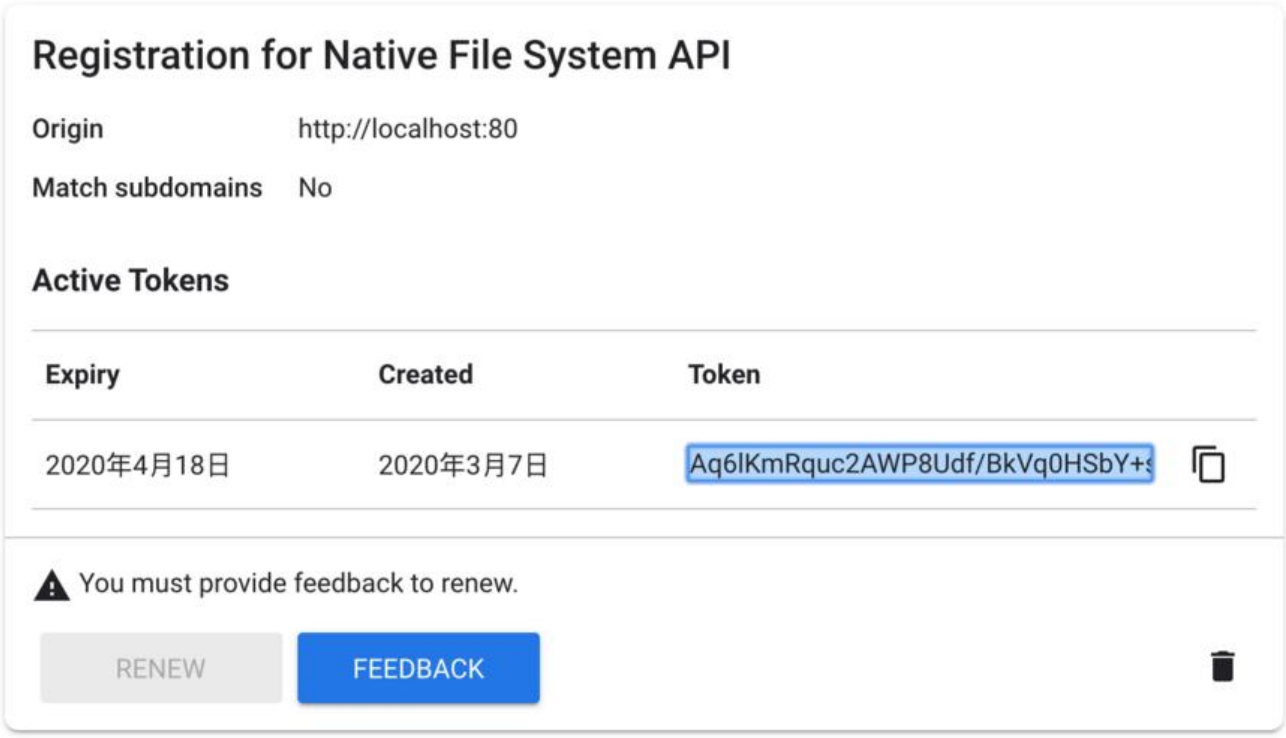
注意事项

- 如果想在本地体验原生文件系统 API，需在浏览器地址中输入 chrome://flags，然后启用 #native-file-system-api
- Token 有一个过期时间
- 写入文件之后必须调用 close 方法，否则无法保证将内容写入磁盘
- 不支持 Safari

步骤

申请

为你的地址申请一个 Token，仅支持 https，chrome-extension，http（仅 localhost）



原文链接：使用浏览器原生 API 读写本地文件

添加 Token

可以通过以下两种方式中的任意一种进行配置：

- 在需要使用到的每一个页面的 head 中添加 meta 标签

```
<meta http-equiv="origin-trial" content="TOKEN_GOES_HERE">
```

- 如果你需要配置你的服务器，你同样可以使用 Origin-Trial HTTP 头来添加 token。返回的头如：

```
Origin-Trial: TOKEN_GOES_HERE
```

选择本地文件

这需要使用原生文件系统 API 中的 window.chooseFileSystemEntries()。当调用该 API 时，将会弹一个文件选择器对话框，提示用户选择一个文件。在选择文件后，该 API 将返回此文件的句柄 FileSysemFileHandle，以便后续操作。

chooseFileSystemEntries 的参数可以设置用户选择文件的行为，如：允许用户选择多个文件或者目，选择不同的文件类型。默认仅允许用户选择一个文件。

这和其他很多强大的 API 一样，chooseFileSystemEntries 必须在安全的上下文中进行调用，并且必通过用户的点击手势才能触发。

FileSystemFileHandle 包含了选中文件的交互属性和方法。保存他的引用，以便后续可继续使用。修改后的内容保存回文件中或执行其他文件操作就需要使用到他。在安装 PWA 后，允许将他保存到 IdexedDB 中，以便刷新后还可以对文件进行持久化的访问。

```
let fileHandle
document.getElementById('openFileBtn').addEventListener('click', async () => {
  fileHandle = await window.chooseFileSystemEntries()
})
```

读取文件内容

当有了一个文件句柄后，你就可以获取该文件的属性，访问文件本身。调用 handle.getFile() 将返回个包含 blob 的 File 对象。调用他自身的方法 slice(), stream(), text(), arrayBuffer() 后即可从 blob 获取数据。

```
const file = await fileHandle.getFile()
const contents = await file.text()
document.getElementById('fileContent').value = contents
```

写文件

- 保存到当前文件

将文件写入磁盘，我们可以调用文件句柄对象的 createWriter() 方法，他将返回一个 FileSystemWritr 对象。当 createWriter() 被调用后，Chrome 会首先检测用户对该文件是否有写的权限。如果用户授权了，浏览器将弹出该权限的信息。如果没有被授权，将会抛出一个 DOMException 异常，应用不能写入文件。

调用 FileSystemWriter.write() 写入你的内容。write() 方法接受一个字符串，当然也可以接受一个 Bu

ferSource 或一个 Blob。最终，通过调用 FileSystemWriter.close() 来完成写入。

```
document.getElementById('saveFileBtn').addEventListener('click', async (e) => {
  const writer = await fileHandle.createWriter()
  await writer.write(0, document.getElementById('fileContent').value)
  await writer.close()
})
```

- 另存为一个新文件

另存为新文件时需要新建一个句柄。可在 chooseFileSystemEntries() 中传入 {type: 'saveFile'} 参数调用后将弹出保存模式，允许用户选择一个文件来进行保存。

```
function getNewFileHandle() {
  const opts = {
    type: 'saveFile',
    accepts: [{
      description: 'Text file',
      extensions: ['txt'],
      mimeTypes: ['text/plain'],
    }],
  };
  const handle = window.chooseFileSystemEntries(opts);
  return handle;
}
```

读取目录

当 chooseFileSystemEntries() 方法中的参数 type 为 openDirectory 时，可枚举出目录中的所有文
。

```
document.getElementById('openDirBtn').addEventListener('click', async (e) => {
  const opts = {type: 'openDirectory'}
  const handle = await window.chooseFileSystemEntries(opts)
  const entries = await handle.getEntries()
  let dirHTML = ''
  for await (const entry of entries) {
    const kind = entry.isFile ? 'File' : 'Directory'
    dirHTML += `<div>${kind} - ${entry.name}</div>`
  }
  document.getElementById('dir').innerHTML = dirHTML
})
```

源码

```
<html>
<head>
  <meta http-equiv="origin-trial"
      content="Aq6lKmRquc2AWP8Udf/BkVq0HSbY+sAsfM/VIwSnvCXZRTA+uegK47AurmB
VaQr2G0vxxFj5z0Kwss7kmzEfA8AAABReyJvcmlnaW4iOiJodHRwOi8vbG9jYWxob3N0OjgwIiwi
mVhdHVyZSI6Ik5hdGl2ZUZpbGVTeXN0ZW0iLCJleHBpcnkiOjE1ODcxNzgzNjd9">
</head>
<body>
<button id="openDirBtn">Open Directory</button>
```

```
<button id="openFileBtn">Open File</button>
<button id="saveFileBtn">Save File</button>
<div style="display: flex;width: 1024px;margin-top: 20px">
    <div id="dir" style="width: 236px;margin-right: 20px"></div>
    <textarea id="fileContent" style="flex:1; height: 300px"></textarea>
</div>
<script>
 let fileHandle
 document.getElementById('openFileBtn').addEventListener('click', async () => {
   fileHandle = await window.chooseFileSystemEntries()
   const file = await fileHandle.getFile()
   const contents = await file.text()
   document.getElementById('fileContent').value = contents
 })

 document.getElementById('saveFileBtn').addEventListener('click', async (e) => {
   const writer = await fileHandle.createWriter()
   await writer.write(0, document.getElementById('fileContent').value)
   await writer.close()
 })
 document.getElementById('openDirBtn').addEventListener('click', async (e) => {
   const opts = {type: 'openDirectory'}
   const handle = await window.chooseFileSystemEntries(opts)
   const entries = await handle.getEntries()
   let dirHTML = ''
   for await (const entry of entries) {
     const kind = entry.isFile ? 'File' : 'Directory'
     dirHTML += `<div>${kind} - ${entry.name}</div>`
   }
   document.getElementById('dir').innerHTML = dirHTML
 })
</script>
</body>
</html>
```

返回总目录

每天 30 秒系列之大前端