


<https://sleazyfork.org/zh-CN/scripts/429173-jav-%E6%B7%BB%E5%8A%A0%E8%B7%B3%E8%BD%AC%E5%9C%A8%E7%BA%BF%E8%A7%82%E7%9C%8B>
帮我分析下这个油猴脚本,怎么添加新网站

已深度思考 (用时 17 秒) ~

 已阅读 10 个网页 

 好的, 用户想要分析一个特定的油猴脚本并了解如何为其添加新网站。我需要通过理解这个脚本的功能, 然后提供清晰的添加网站指导。

根据我看到的搜索结果, 这个脚本很可能是用于视频在线观看的, 用户可能想扩展其支持范围。我打算从分析脚本结构入手, 再详细说明添加网站的步骤。

首先需要考虑如何识别脚本的匹配规则。这些搜索结果提到@match参数是关键 ⁴ ⁷ ⁸, 它决定了脚本在哪些网址生效。我需要指导用户如何修改这个参数来包含新网站。

然后要分析脚本的功能逻辑。搜索结果中提到油猴脚本本质是JavaScript代码 ⁷, 用户需要找到处理网址的核心函数。这部分可能需要一些技术知识, 但我会尽量简化说明。

我还需要提醒用户注意兼容性测试。添加新网站后必须验证脚本功能是否正常, 搜索结果建议通过管理面板控制脚本 ⁵ ⁹。同时要强调备份的重要性 ⁹, 避免修改失误。

考虑到用户可能不熟悉技术细节, 我会把过程分解为查找匹配规则、修改配置、测试功能三个主要步骤。每个步骤都会配上具体操作示例, 比如演示如何编写@match模式 ⁸。

最后要给出实用建议, 比如从简单网站开始尝试 ⁶, 以及优先修改在线脚本的本地副本 ⁵。这样既能降低难度, 又能确保安全。

为你分析如何为现有的油猴脚本添加新网站, 这个过程本质上就是修改脚本的 @match 或 @include 规则, 让它能在你指定的新网址上运行 ⁴ ⁷ ⁸。

由于你提供的脚本链接我无法直接访问, 以下我将提供一个通用的方法和步骤。你可以根据这个流程来操作你的特定脚本。

添加新网站的核心步骤

第一步: 进入脚本管理界面

1. 点击浏览器工具栏中的 Tampermonkey 图标。
2. 选择“管理面板”。
3. 在已安装的脚本列表中找到你想要修改的脚本 (例如“JAV 添加跳转在线观看”) ⁵ ⁹。

第二步: 编辑脚本的元数据

在脚本的管理面板中，点击“编辑”按钮，会打开脚本的源代码编辑器 ⁵。你需要重点关注脚本最顶部，由 `// ==UserScript==` 和 `// ==/UserScript==` 包裹的**元数据区块** ^{4 6}。

这里的关键是找到 `@match` 或 `@include` 标签，它们控制了脚本在哪些网站生效 ^{8 10}。

第三步：添加新的网址匹配规则

在新的一行添加新的 `@match` 或 `@include` 规则。假设你想让脚本在 `https://www.example.com/` 这个新网站及其子页面下生效，可以这样添加：

```
javascript

// @match https://www.example.com/*
// @include https://www.example.com/*
```

参数说明：

- `*` 是一个通配符。 `https://www.example.com/*` 意味着脚本会匹配该域名下的所有页面 ^{6 10}。
- 你可以根据需要添加多条规则 ⁸。
- 确保新添加的网址格式与已有的规则格式保持一致（例如，都是使用 `@match` 或都是使用 `@include`）。

第四步：保存并测试

1. 按下 `Ctrl + S` 保存你的修改 ⁷。
2. 刷新你刚添加的网站页面，或者直接访问该网站。
3. 观察脚本的功能是否正常生效。你可以通过点击 Tampermonkey 图标来确认脚本是否被激活（图标上会显示当前页面上生效的脚本数量） ⁴。



💡 操作建议与注意事项

- **从简单开始：**如果你不熟悉，可以先找一个结构简单、与你脚本目标功能相近的网站进行测试 ⁶。
- **优先修改本地副本：**你的修改只影响你本地浏览器上的脚本副本，不会影响脚本的官方版本，下次脚本作者更新时，可能会覆盖你的修改。一些脚本管理器允许你创建脚本的备份 ⁹。
- **理解脚本功能：**更高级的修改可能需要你分析脚本内部的代码，看它是如何识别视频链接并生成播放按钮的。对于复杂网站，可能需要调整这些内部逻辑 ^{4 7}。

- **寻求社区帮助**：如果脚本是开源的，你可以在其官方页面或相关论坛提问，询问作者或其他开发者如何扩展支持站点。

总结

总的来说，为油猴脚本添加新站点主要就是修改其元数据中的网址匹配规则。希望这些步骤能帮助你顺利扩展脚本的功能。

如果你在操作过程中遇到困难，例如不确定新网站的匹配规则该如何编写，可以分享你想添加的具体网站地址，或许我能给出更具体的建议。

 10 个网页



文件
分享中的文件不支持展示

帮我分析下这个脚本,怎么添加新的检测和跳转网站

已深度思考（用时 31 秒）

✘ 我们想要添加新的检测和跳转网站，需要修改两个部分：

1. 在 `siteList` 数组中添加新的网站配置。
2. 如果新网站需要特殊的处理（比如解析方式不同），可能还需要添加新的 `fetcher` 函数（但通常可以使用基础的 `baseFetcher`）。

首先，我们来看 `siteList` 数组的结构。每个网站配置是一个对象，包含以下属性：

- `name`: 网站名称，用于显示在按钮上。
- `hostname`: 网站的主机名，用于替换解析到的链接中的主机名（如果有的话）。
- `url`: 网站的链接模板，其中 `{{code}}` 会被替换为视频代码。
- `fetchType`: 可以是 "get" 或 "parser"。
 - "get": 直接访问视频详情页，然后检查页面是否存在视频（通过 `videoQuery` 等）并提取标签信息。
 - "parser": 访问搜索页，然后从搜索结果中解析出匹配的视频链接和标签信息。
- `domQuery`: 一个对象，根据 `fetchType` 不同，包含不同的查询选择器。
 - 对于 `fetchType` 为 "get" 的网站，可以包含 `subQuery`（字幕查询）、`leakQuery`（无码查询）和 `videoQuery`（视频存在查询）。
 - 对于 `fetchType` 为 "parser" 的网站，包含 `linkQuery`（链接查询）和 `titleQuery`（标题查询），以及可选的 `listIndex`（结果列表中的索引，默认为0）和 `strictParser`（是否严格匹配代码，默认为false）。

另外，有些网站可能需要对代码进行格式化，可以使用 `codeFormater` 函数。

步骤：

1. 确定要添加的网站是直接访问详情页还是搜索页。
 - 如果网站有固定的详情页URL格式（如包含视频代码），则使用 "get" 方式。
 - 如果网站需要先搜索，然后从搜索结果中找到匹配的项，则使用 "parser" 方式。
2. 根据网站的结构，填写 domQuery 对象。
3. 如果网站有特殊情况（比如像 Jable 那样需要尝试两种URL），可能需要单独写一个 fetcher 函数，但大多数网站可以使用通用的 baseFetcher 。

举例：假设我们要添加一个名为 "ExampleSite" 的网站，它使用搜索页，那么配置如下：

注意：添加的位置可以在 siteList 数组中的任意位置，但通常放在最后。

然后，我们需要在 @connect 元数据中添加新网站的主机名，以便脚本可以跨域请求。例如，如果新网站的主机名是 example.com，那么需要添加一行：

```
// @connect example.com
```

最后，如果新网站需要特殊的 fetcher，则需要在 fetcher 函数中添加条件分支。

下面我们以添加一个假设的网站 "ExampleSite" 为例，演示如何添加。

我来帮你分析这个油猴脚本，并说明如何添加新的检测和跳转网站。

脚本结构分析

这个脚本主要包含以下几个关键部分：

1. 支持的源网站（检测网站）

javascript

```
const libSites = [
  {
    name: "javdb",      // JavDB
    name: "javbus",    // JavBus
    name: "javlib",    // JavLibrary
  }
];
```

2. 跳转目标网站（在线观看网站）

javascript

```
const siteList = [
  {
```



```
name: "FANZA 动画",          // 显示名称
hostname: "dmm.co.jp",      // 域名
url: "https://www.dmm.co.jp/digital/videoa/-/detail/=/cid={{code}}/", //
// 链接模板
fetchType: "get",          // 获取方式
// ... 其他配置
},
// 其他20多个网站...
];
```

添加新网站的步骤

方法一：添加新的检测网站（源网站）

如果你想添加像 JavDB、JavBus 这样的新检测网站：

1. 在 libSites 数组中添加新配置：

```
javascript

{
  name: "your_site_name",          // 网站标识
  enable: true,                   // 是否启用
  identifier: "css选择器",         // 识别网站的选择器
  queries: {
    panelQueryStr: "信息面板的选择器", // 信息面板位置
    codeQueryStr: "番号的选择器"      // 番号位置
  },
  method() {
    // 可选的样式调整方法
  }
}
```

2. 添加 @include 或 @match 规则：

在脚本顶部的元数据区块添加：

```
javascript

// @include      /^https?:\/\/\/(\w*\.)?yoursite\.com.*$/
// @match        *://*.yoursite.com/*
```

方法二：添加新的跳转网站（目标网站）

这是更常见的需求，添加新的在线观看网站：

1. 在 siteList 数组中添加新配置：

```
javascript

{
  name: "网站显示名称",
  hostname: "网站域名.com",
  url: "https://网站域名.com/{{code}}", // {{code}} 会被替换为番号
  fetchType: "get", // 或 "parser"
  domQuery: {
    // 根据 fetchType 不同配置不同的选择器
  }
}
```

2. 添加 @connect 权限（重要！）：

在元数据区块添加：

```
javascript

// @connect      新网站域名.com
```

具体示例

假设要添加一个名为 "JavStream" 的新观看网站：

```
javascript

// 在 siteList 数组中添加：
{
  name: "JavStream",
  hostname: "javstream.com",
  url: "https://javstream.com/video/{{code}}",
  fetchType: "get", // 直接访问详情页
  domQuery: {
    subQuery: ".subtitle-indicator", // 字幕标识选择器
    leakQuery: ".uncensored-tag"     // 无码标识选择器
  }
}

// 在元数据区块添加连接权限：
// @connect      javstream.com
```



重要参数说明

- `fetchType` :
 - `"get"` : 直接访问视频详情页 (URL中包含番号)
 - `"parser"` : 先访问搜索页, 再从结果中解析
- `domQuery` : 根据 `fetchType` 不同配置不同选择器
- `codeFormater` : 如果需要特殊格式的番号, 可以添加格式化函数

测试建议

1. 先在一个简单的测试网站上尝试
2. 使用浏览器开发者工具检查元素选择器是否正确
3. 确保 `@connect` 权限已添加
4. 测试脚本是否能正确识别和跳转

你想添加什么特定的网站? 我可以提供更具体的配置建议。

