

图片采集上传插件使用文档

一、插件简介

图片采集上传插件是一款高效的浏览器扩展工具，支持网页图片快速采集、自定义服务器存储及 Markdown 格式一键复制等功能，适用于内容创作、素材收集、团队协作等场景，尤其能突破微信公众号等平台的图片防盗链限制。

二、安装方法

1. 准备工作：

- 下载插件源码包并解压到本地文件夹
- 推荐使用 Chrome、Edge 等基于 Chromium 内核的浏览器

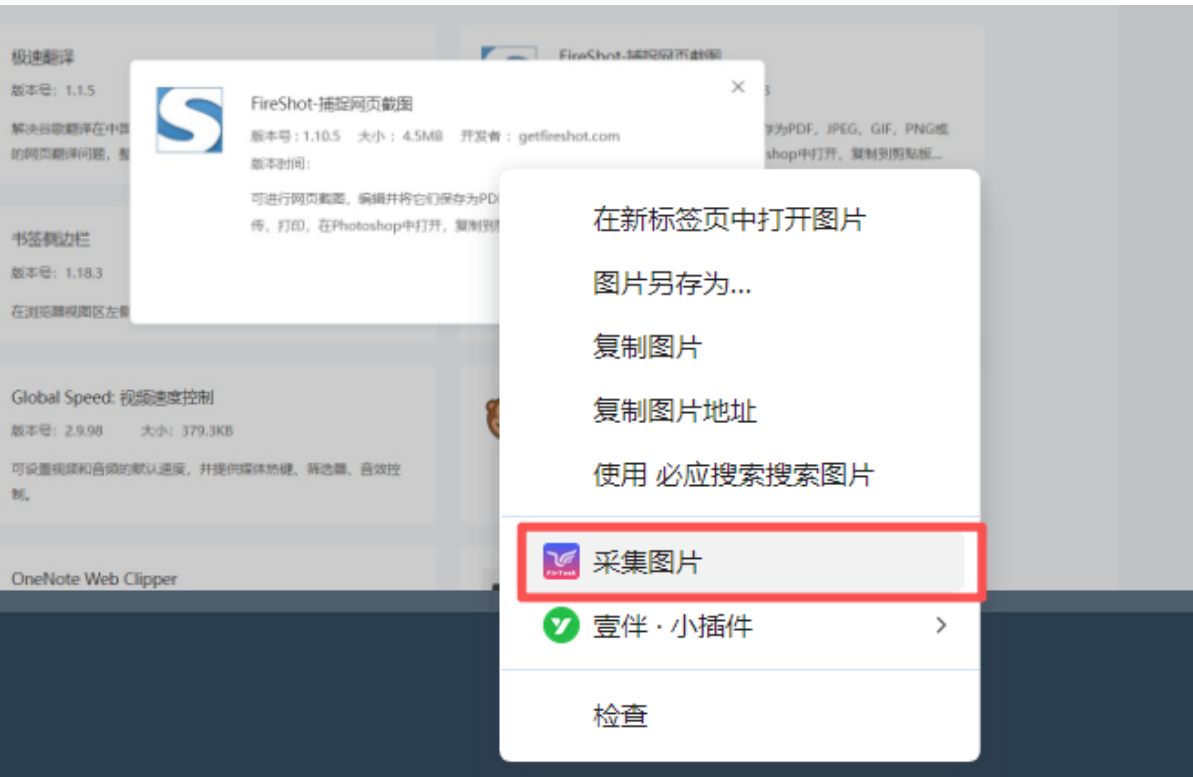
1. 安装步骤：

- 打开浏览器，在地址栏输入 `chrome://extensions/` 进入扩展程序页面
- 开启右上角「开发者模式」开关
- 点击「加载已解压的扩展程序」，选择插件源码所在文件夹
- 插件安装成功后，会在浏览器工具栏显示图标

三、基本功能使用

1. 采集图片

- **单张采集：**在网页中右键点击目标图片，选择「采集图片」，自动唤起图片编辑页面



- **采集说明：**支持各类网页图片（包括动态加载、隐藏图片），兼容微信公众号、知乎、博客等平台

2. 图片编辑页面功能

打开编辑页面后，可进行以下操作：

功能按钮	作用说明
上传到服务器	将采集的图片上传至已配置的自定义服务器
保存到本地	将图片直接下载到电脑本地文件夹
复制地址	复制图片在服务器的直接访问链接
复制 Markdown	生成 <code>![图片描述](图片URL)</code> 格式并复制
服务器设置	进入自定义服务器配置页面



四、自定义服务器配置（重点）

1. 配置步骤



服务器设置

图片上传接口地址：

例如: `https://www.xxx.cn/imgapi`

保存设置

- 点击编辑页面的「服务器设置」进入配置页
- 输入你的服务器上接口地址（如 `https://www.example.com/upload.php`）
- 点击「保存设置」完成配置

2. 服务器要求

- 需支持 `multipart/form-data` 格式的 POST 请求
- 接口需返回包含图片 URL 的 JSON 响应（示例：`{"url": "https://xxx.com/img/123.jpg"}`）
- 建议配置跨域许可（如 PHP 中添加 `header("Access-Control-Allow-Origin: *");`）
- 图片存储目录要求为 `uploads`，如果目录不同，则需要修改插件里「`image-editor.js`」的 URL 生成地址的目录。就是下面代码中的 `uploads` 改为你的图片目录。

```
if (result.filename || result.path || result.url) {  
    // 关键修复：给serverImageUrl赋值，供Markdown使用  
    serverImageUrl = result.url ||  
    `${serverConfig.url.replace('imgapi.php', '')}uploads/${result.filename}`;  
}
```

3. 上传接口示例

PHPdemo

```
<?php  
// 允许插件的域名访问（*表示允许所有，生产环境建议指定具体域名）  
header("Access-Control-Allow-Origin: *");  
header("Access-Control-Allow-Methods: POST, OPTIONS");  
header("Access-Control-Allow-Headers: Content-Type");  
  
// 处理预检请求  
if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {  
    exit;  
}
```

```
// 配置
$uploadDir = 'uploads/';
$allowedTypes = ['image/jpeg', 'image/png', 'image/gif', 'image/webp'];
$maxFileSize = 5 * 1024 * 1024; // 5MB

// 确保上传目录存在
if (!is_dir($uploadDir)) {
    mkdir($uploadDir, 0755, true);
}

// 检查是否有文件上传
if (!isset($_FILES['image'])) {
    http_response_code(400);
    echo json_encode(['error' => '未上传图片']);
    exit;
}

$file = $_FILES['image'];

// 检查上传错误
if ($file['error'] !== UPLOAD_ERR_OK) {
    http_response_code(400);
    echo json_encode(['error' => '上传失败，错误代码：' . $file['error']]);
    exit;
}

// 检查文件类型
if (!in_array($file['type'], $allowedTypes)) {
    http_response_code(400);
    echo json_encode(['error' => '不支持的文件类型']);
    exit;
}

// 检查文件大小
if ($file['size'] > $maxFileSize) {
    http_response_code(400);
    echo json_encode(['error' => '文件过大，最大支持5MB']);
    exit;
}

// 生成安全的文件名
$filename = pathinfo($file['name'], PATHINFO_FILENAME);
$extension = pathinfo($file['name'], PATHINFO_EXTENSION);
$safeFilename = preg_replace('/[^\a-zA-Z0-9_]/', '', $filename);
$timestamp = time();
$targetFilename = "{$timestamp}-{$safeFilename}.{$extension}";
$targetPath = $uploadDir . $targetFilename;

// 移动上传的文件
if (move_uploaded_file($file['tmp_name'], $targetPath)) {
    http_response_code(200);
    echo json_encode([
        'message' => '图片上传成功',
        'filename' => $targetFilename,
        'path' => $targetPath
    ]);
}
```

```
} else {  
    http_response_code(500);  
    echo json_encode(['error' => '保存文件失败']);  
}  
?>
```

五、常见问题解决

1. 图片采集失败：

- 原因：部分网站有严格的防盗链机制
- 解决：插件会自动尝试绕过，若失败可刷新页面重试

1. 上传失败（提示跨域错误）：

- 原因：自定义服务器未配置跨域许可
- 解决：在服务器端添加跨域头（如 PHP 中添加 `header("Access-Control-Allow-Origin: *");`）

1. 上传成功但无法访问图片：

- 检查服务器返回的图片 URL 是否正确
- 确认服务器存储目录是否有公开访问权限

1. Markdown 复制无效：

- 确保图片已成功上传（需先点击「上传到服务器」）
- 若浏览器不支持剪贴板 API，插件会自动降级使用备用方案

通过以上功能，图片采集上传插件可大幅简化图片管理流程，尤其适合自媒体创作者、设计师、研究人员等高频使用图片的场景。如有其他问题，可查看插件内相关提示或检查浏览器开发者工具中的错误信息。