链滴

使用 JsDeliver CDN 加速 Github 文件

作者：zxniuniu

原文链接：https://ld246.com/article/1608014680326

来源网站：链滴

前言

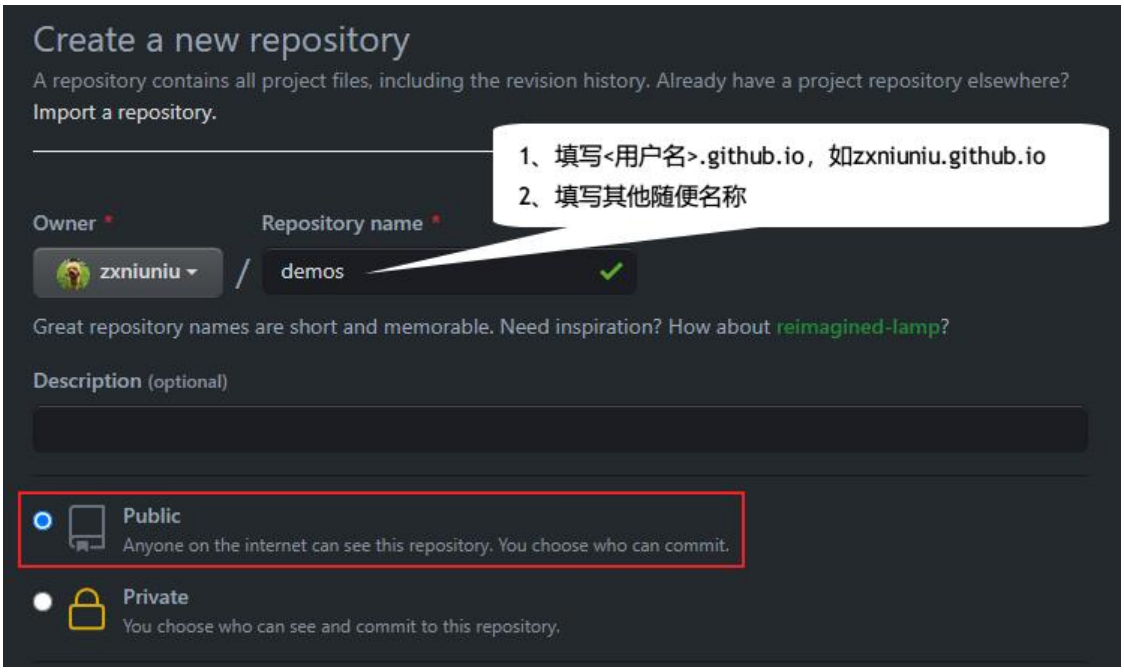
在国内，不使用梯子时github访问速度很慢，甚至有时候会完全中断，无法充分利用github提供的优仓储。在有了jsdelivr后，可以免费为github所有仓库做CDN，连国内都可以极速访问！以下简单说下采用 GitHub+JsDeliver 加速文件的方法。

建立仓库

关于 Github 仓库可以新建，也可以使用 <用户名>.github.io 这个仓库。这里说明一下，仓库名称分两种情况：第一种，名称采用 <用户名>.github.io形式，这种形式最终的访问地址为 用户名>.github.io；另一种是采用其他名字，这时访问地址为：<用户名>.github.io/仓库名称。

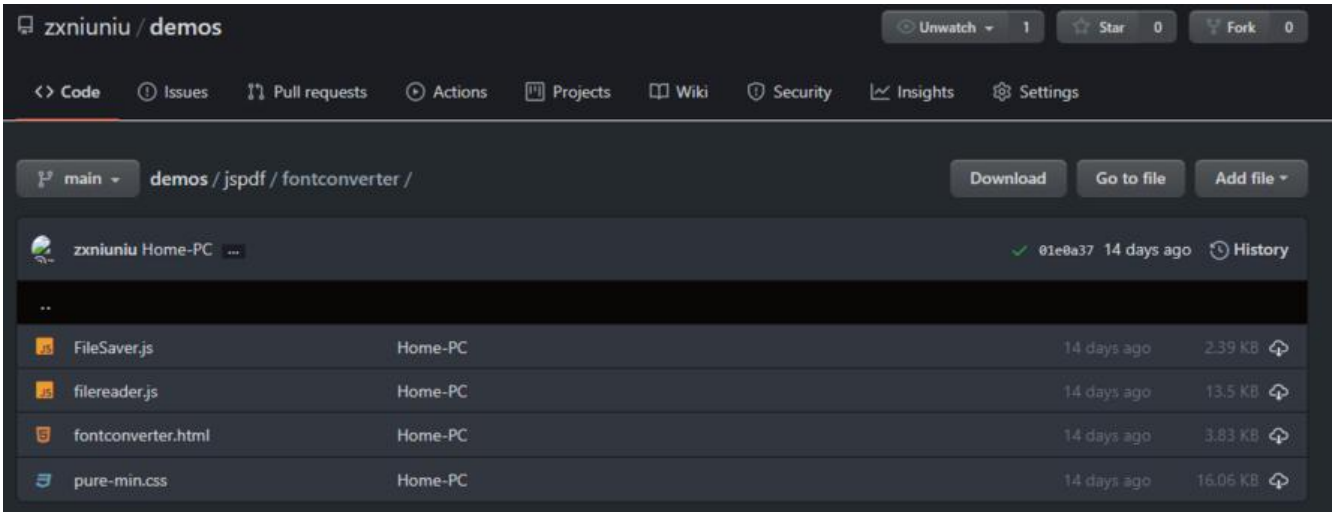
假如 <用户名>为 zxniuniu，当创建时的仓库名称为 zxniuniu.github.io时，部署 Github Pages后终访问地址即为 https://zxniuniu.github.io；当创建时的仓库名称为其他的，比如 demos时，访问址将为 https://zxniuniu.github.io/demos/文件名称。另外，请注意仓库一定要是**公开**，不能是私。

原文链接：使用 JsDeliver CDN 加速 Github 文件



上传文件

更新项目，采用Idea，eclipse，或者直接使用 Github Desktop 更新项目文件。如我上传了jspdf中用的字体ttf转js格式文件。



使用 jsdelivr引用

原始文件路径如下：

https://github.com/zxniuniu/demos/blob/main/jspdf/fontconverter/fontconverter.html

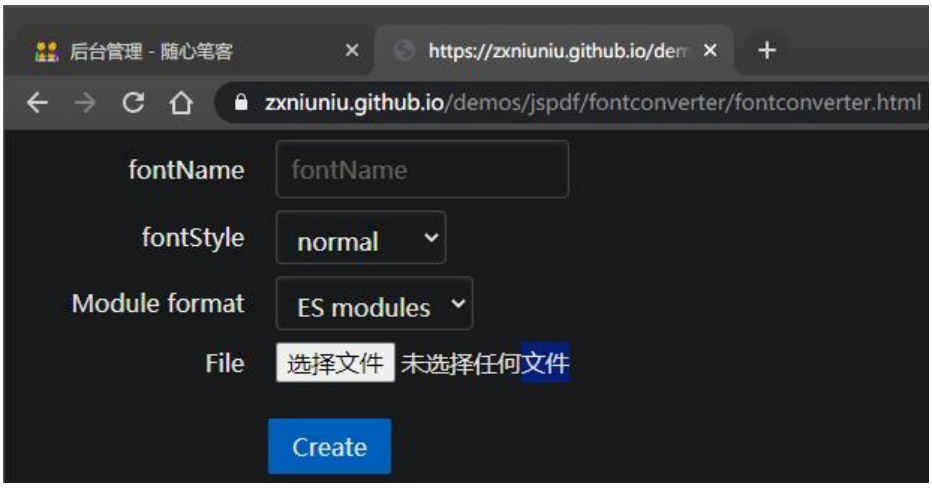
访问的文件路径如下，如果为图片，会直接显示图片，此处会直接显示html文件内容：

https://cdn.jsdelivr.net/gh/zxniuniu/demos/jspdf/fontconverter/fontconverter.html

Github Pages页面内容如下：

http://zxniuniu.github.io/demos/jspdf/fontconverter/fontconverter.html

原文链接：使用 JsDeliver CDN 加速 Github 文件



引用方式

上面的链接为 https://cdn.jsdelivr.net/gh/zxniuniu/demos/jspdf/fontconverter/fontconverter.html，但这并不意味着 jsdelivr只有这一种引用方式。上面的引用方式为直接引用，当然还有按分支及本号引用等。

- 直接引用

这种方式也就是上面的方式，格式为：https://cdn.jsdelivr.net/gh/<用户名>/<仓库名>/<文件及路径>

例如：https://cdn.jsdelivr.net/gh/zxniuniu/demos/jspdf/fontconverter/fontconverter.html 代用户名 zxniuniu 下的 demos 仓库中文件夹 jspdf/fontconverter/ 里的 fontconverter.html 文件。

- 分支及版本号

分支与版本号加到仓库后变，用 @符链接。格式为：https://cdn.jsdelivr.net/gh/<用户名>/<仓库>@[分支/版本号]/<文件及路径>。

使用版本号引用的优点在于：这个链接仅停留在发布版本号的时刻，无论仓库如何变化，这个版本号文件都不会受到影响。同时可以避免 jsdelivr 缓存问题。

关于缓存问题

这个问题跟迷一样，根据实测不仅与分支有关系而且与文件名有关系。文件名为 *.min.* 或者是 *.*，就是带 min 的和不带 min 的。这里我以 index.min.css 和 index.css 为例。

先来看看 index.css

分支	首次上传	能否及时更新	存天数
master	可以被引用	第一次 push 和第一次修改可	更新可能 1 天
latest	可以被引用	与 master 分支几乎一致	能 1 天
版本号	发布版本后引用	发布后及时更新	个版本独立

原文链接：使用 JsDeliver CDN 加速 Github 文件

再来看看 index.min.css

分支	首次上传	能否及时更新	存天数
master	可以被引用	第一次 push 可以更新	能 1 天
latest	可以被引用	第一次 push 和第一次修改可	更新可能 1 天
版本号	发布版本后引用	发布后及时更新	个版本独立

因此总结一下使用方式就是：

- 图床，或不需要修改文件，可以直接使用 latest 分支或者 master 分支即可
- 静态文件仓库，或经常需要改动的文件，建议使用版本号方式

文件是否有限制

- GitHub 公开仓库大小为 100G，并且可以创建无数个仓库哦！但仓库超过 1G 后会有人工审核仓内容，如果发现用来做图床～～～smirk
轻则删库，重则封号。因此为了安全建议在 1G 之前就换个仓库，反正可以创建无数个仓库嘛。
- Github 单文件上传限制为 100M，但是 jsdelivr 加速的单文件大小为 50M，因此也就意味着要加的单文件大小限制为 50M。
- 文件类型方面，基本的图片、视频、静态文件等都可以。

查看仓库文件

查看仓库文件有大小限制，因此当仓库文件大于 50M时，就无法通过 jsdelivr 查看了，只能在 GitHub 仓库查看。jsdelivr 查看仓库文件有两种方式：

- 查看仓库版本号

格式：https://www.jsdelivr.com/package/gh/+ 用户名 +/+ 仓库名

例如：https://www.jsdelivr.com/package/gh/zxniuniu/demos

- 查看仓库文件

格式https://cdn.jsdelivr.net/gh/+ 用户名 +/+ 仓库名 +/ 仓库名 [@分支]

例如仓库名@master，默认为 master 分支，https://cdn.jsdelivr.net/gh/zxniuniu/demos@maste/

jsDelivr API

API地址如下：https://data.jsdelivr.com/v1

版本查看

/package/npm/:name 或 /package/gh/:user/:repo

```
https://data.jsdelivr.com/v1/package/gh/zxniuniu/demos
// => {
    "tags": [],
    "versions": [
        "1.0.0"
    ]
}
```

文件查看

```
/package/npm/:name@:version/:structure? 或 /package/gh/:user/:repo@:version/:structure?
// 其中：structure: tree or flat; 默认值为tree

https://data.jsdelivr.com/v1/package/gh/zxniuniu/demos@1.0.0
// =>
{
    "default": null,
    "files": [
        {
            "type": "directory",
            "name": "jspdf",
            "files": [
                {
                    "type": "directory",
                    "name": "fontconverter",
                    "files": [
                        {
                            "type": "file",
                            "name": "fontconverter.html",
                            "hash": "GQY2zOPzMHl5LBQPGngKPiIFnQcTlGRAWymjCawb88o=",
                            "time": "2020-11-30T15:00:21.000Z",
                            "size": 3924
                        }
                    ]
                },
                {
                    "type": "file",
                    "name": "jspdf-chinese-export.html",
                    "hash": "QrrKYVre6pPmR2H0IctW5ydJ7i477WjgYeHrFDOaZvI=",
                    "time": "2020-11-30T15:00:21.000Z",
                    "size": 4446
                }
            ]
        }
    ]
}
```

查看版本范围

```
/package/resolve/npm/:name@:range 或  /package/resolve/gh/:user/:repo@:range
https://data.jsdelivr.com/v1/package/resolve/gh/zxniuniu/demos@1
// =>{
    "version": "1.0.0"
```

```
}
```

使用情况统计

```
/package/npm/:name/stats/:groupBy?/:period? 或 /package/gh/:user/:repo/stats/:groupBy?/:
eriod?
// groupBy可用值：version or date，默认为version
// period可用值： day or week or month or year，默认值month
https://data.jsdelivr.com/v1/package/gh/zxniuniu/demos/stats
// =>{
    "rank": null,
    "total": 0,
    "versions": {},
    "commits": {}
}
```

更多API参见 jsDelivr API

清理 jsdelivr 缓存

把链接地址中的 cdn 换成 purge 即可清除指定文件的缓存，但经过测试，这个方法也是有时候好用时候不好用。如清理zxniuniu/demos/jspdf/fontconverter/fontconverter.html 文件缓存：

```
https://purge.jsdelivr.net/gh/zxniuniu/demos/jspdf/fontconverter/fontconverter.html
// => {
    "fastly": [
        {
            "status": "ok",
            "id": "20777-1607803511-297900"
        },
        {
            "status": "ok",
            "id": "20774-1607805727-283092"
        }
    ],
    "maxcdn": {
        "code": 200
    },
    "cloudflare": true,
    "quantil": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<response><message>success
/message></response>"
}
```

参见:

优雅使用JsDeliver加速文件

一个帮你实时刷新jsdelivr CDN缓存的小工具

解决jsdelivr缓存问题的几个办法