

媒体中心分享

子天

- 图片加水印

- 图片生成

- 断点续传

- 音频转码

- 视频分发

- 接口跨域

- 后续

640x960

@小子欠扁

640x960

@小子欠扁

640x960

@小子欠扁

微博的水印

颜色空间

- RGB
- YUV
- Lab'

RGB

对应于电脑显示器、电视屏幕等显示设备，因此也被称为色光模式。当RGB都为0，即三种色光都同时不发光时就成了黑色。光线越强，颜色越亮，当RGB都为255，即三种色光都同时发光到最亮就成了白色，所以RGB模式被称为加色法。

YUV

Y'UV 的发明是由于彩色电视与黑白电视的过渡时期。黑白视讯只有 Y (Luma, Luminance) 视讯，也就是灰阶值。到了彩色电视规格的制定，是以 YUV/YIQ 的格式来处理彩色电视图像，把 UV 视作表示彩度的 C (Chrominance或 Chroma) ，如果忽略 C 讯号，那么剩下的 Y (Luma) 讯号就跟之前的黑白电视讯号相同，这样一来便解决彩色电视机与黑白电视机的相容问题。

Lab'

Lab 模式是由国际照明委员会(CIE) 与1976年公布的，理论上包括了人眼可见的所有颜色的色彩模式。它不依赖于光线、也不依赖于颜料，弥补了RGB 与CMYK两种色彩模式的不足，是Photoshop 在不同颜色模式之间转换时使用的内部颜色模式，你可以在图像编辑中使用Lab 模式，并且Lab 模式转换为CMYK模式时不会像RGB 转换为CMYK模式时那样丢失色彩，因此，避免色彩丢失的最佳方法是用Lab 模式编辑图像，再转换成CMYK模式打印输出。

颜色空间的转换

- RGB -> YUV

$$Y = R * .229 + G * .587 + B * .114$$

$$U = (B - Y) * .493$$

$$V = (R - Y) * .877$$

- RGB -> XYZ -> Lab'

明暗度对比参考值

- silver: `rgb(192, 192, 192)`

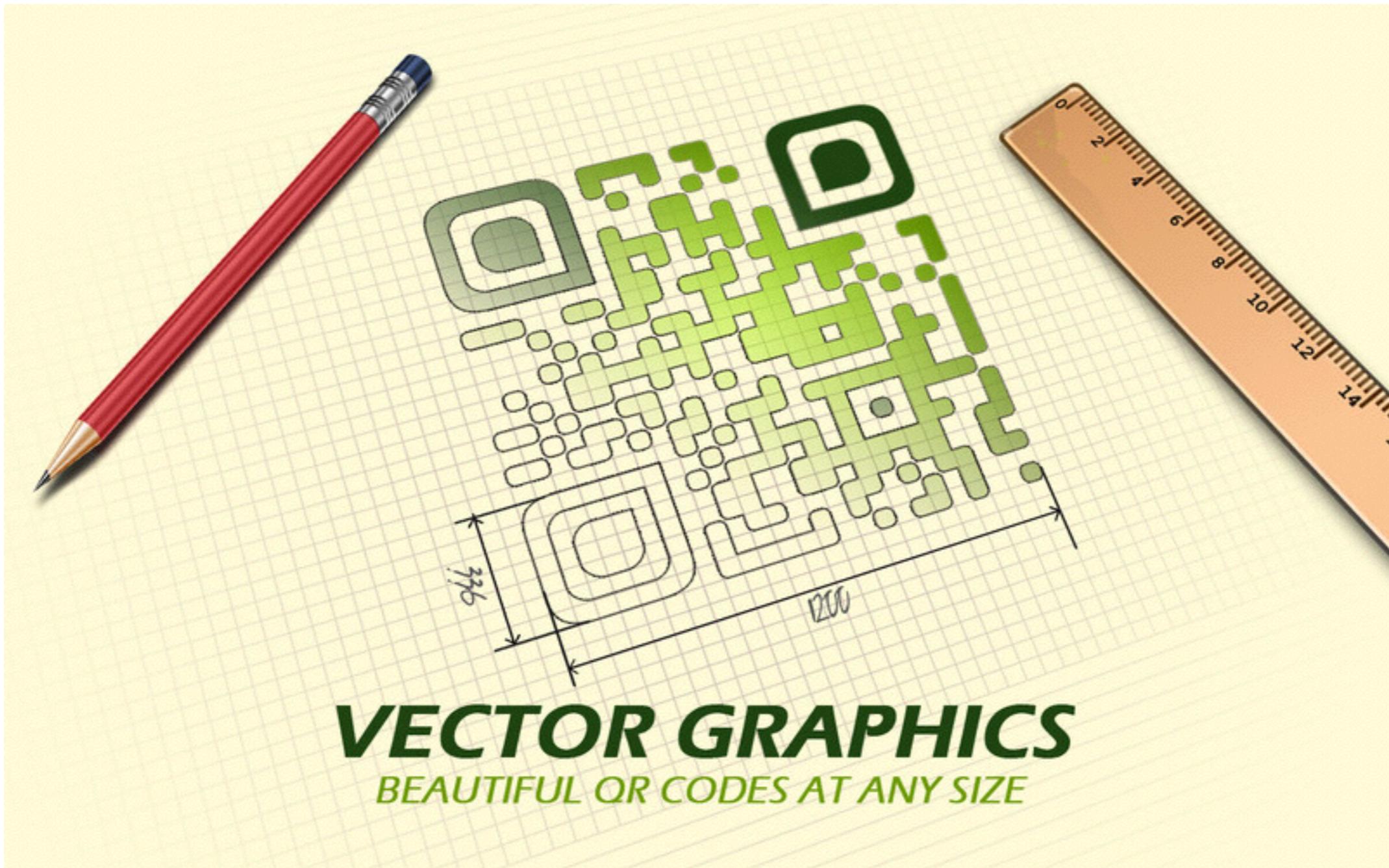
#C0C0C0 (silver)

$Y = 192$

- 判断逻辑

$Y_1 \geq 192 ? \text{“明亮色”} : \text{“暗色”}$

二维码生成服务



ualqr examples output

KUNG FU PANDA 3

facebook.png 6.4 kB

googleplus.png 33.1 kB

moe91 41.1

orzfly.png 9.3 kB

v2ex.png 57.7 kB

wecha 45.4

weibo.png 50.5 kB

youtube.png 134.1 kB

weiboc

facebook.png 6.4 kB

googleplus.png 33.1 kB

moe91 41.1

orzfly.png 9.3 kB

v2ex.png 57.7 kB

wecha 45.4

weibo.png 50.5 kB

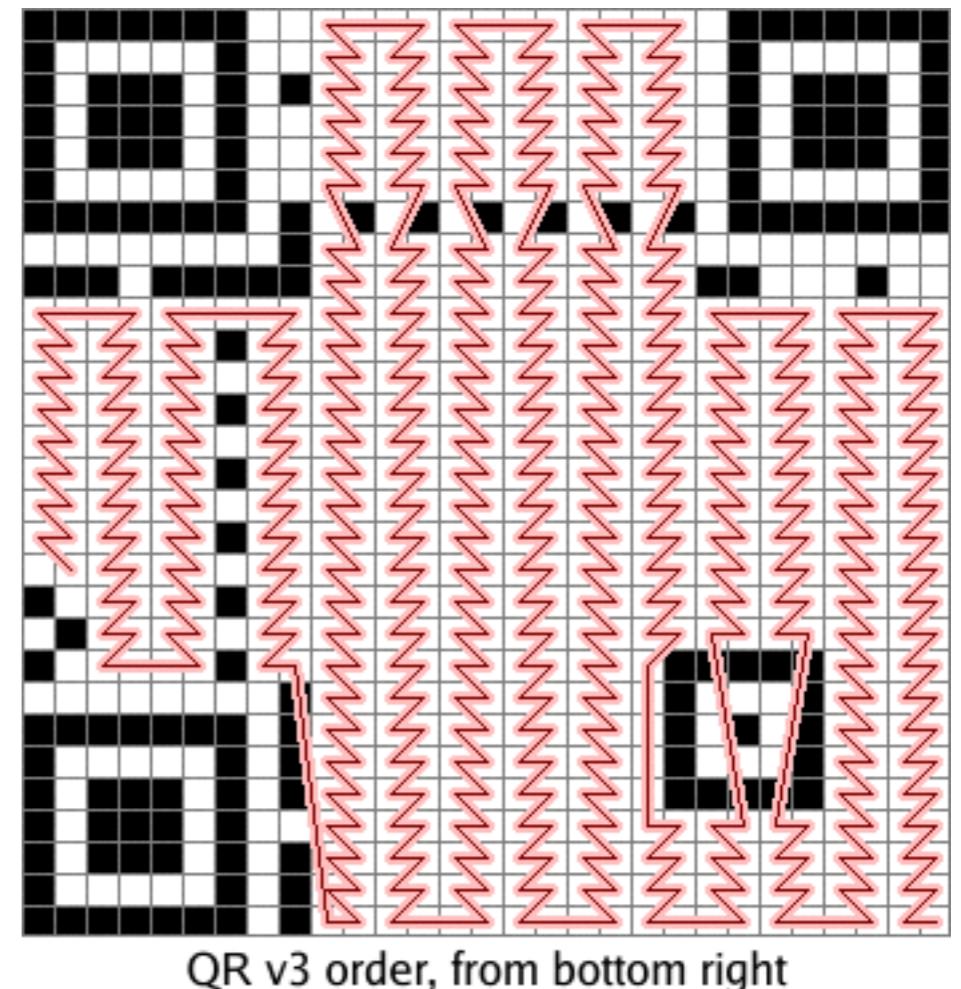
youtube.png 134.1 kB

weiboc

二维码

Error Correction Code Level

- L 7%
 - M 15%
 - Q 25%
 - H 30%
-
- Logo宽度为二维宽度1/5的时候， 面积信息损失为4%

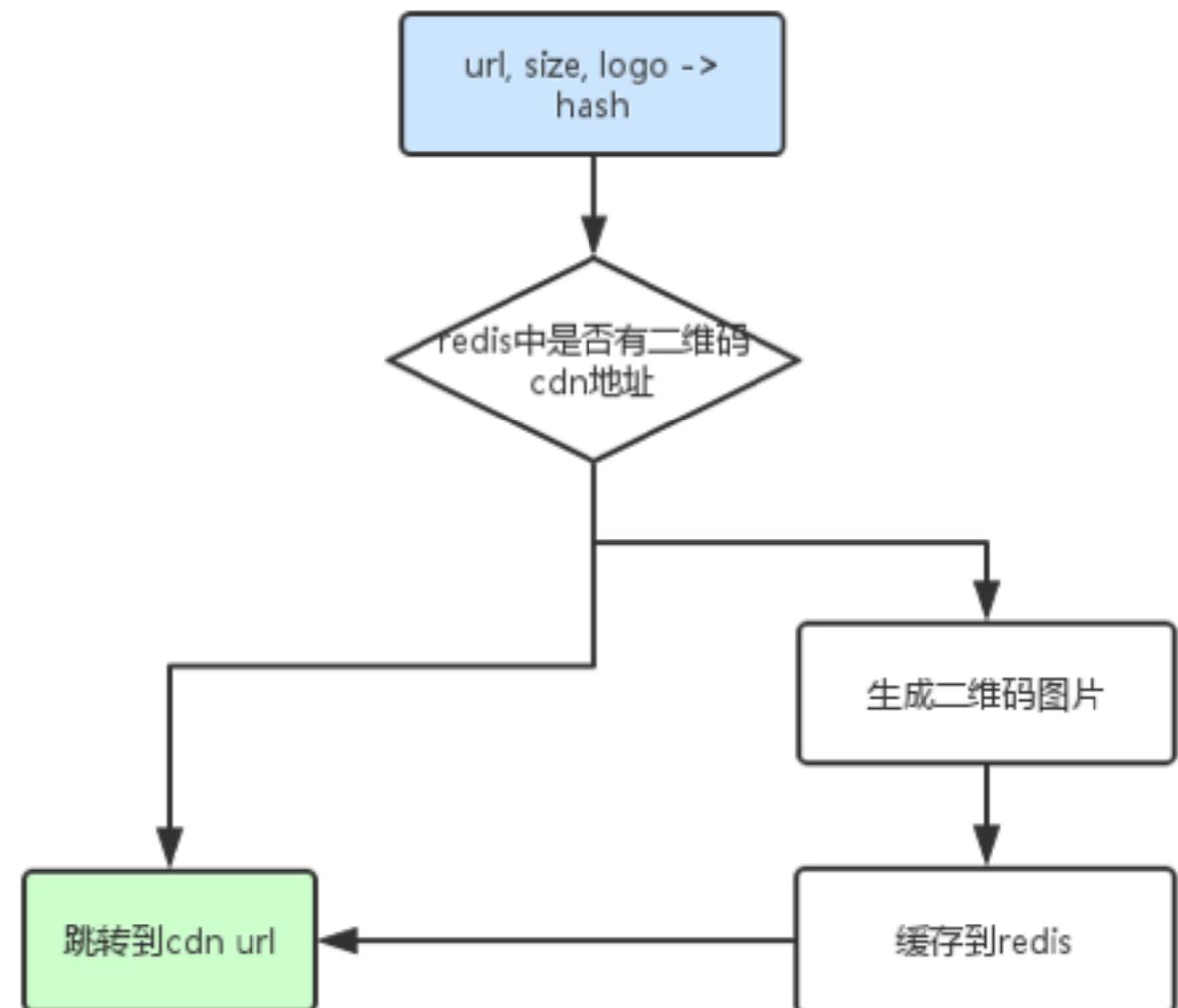


二维码生成服务的优化

- 上传cdn, redis缓存, redirect
- Http缓存
- CDN回源

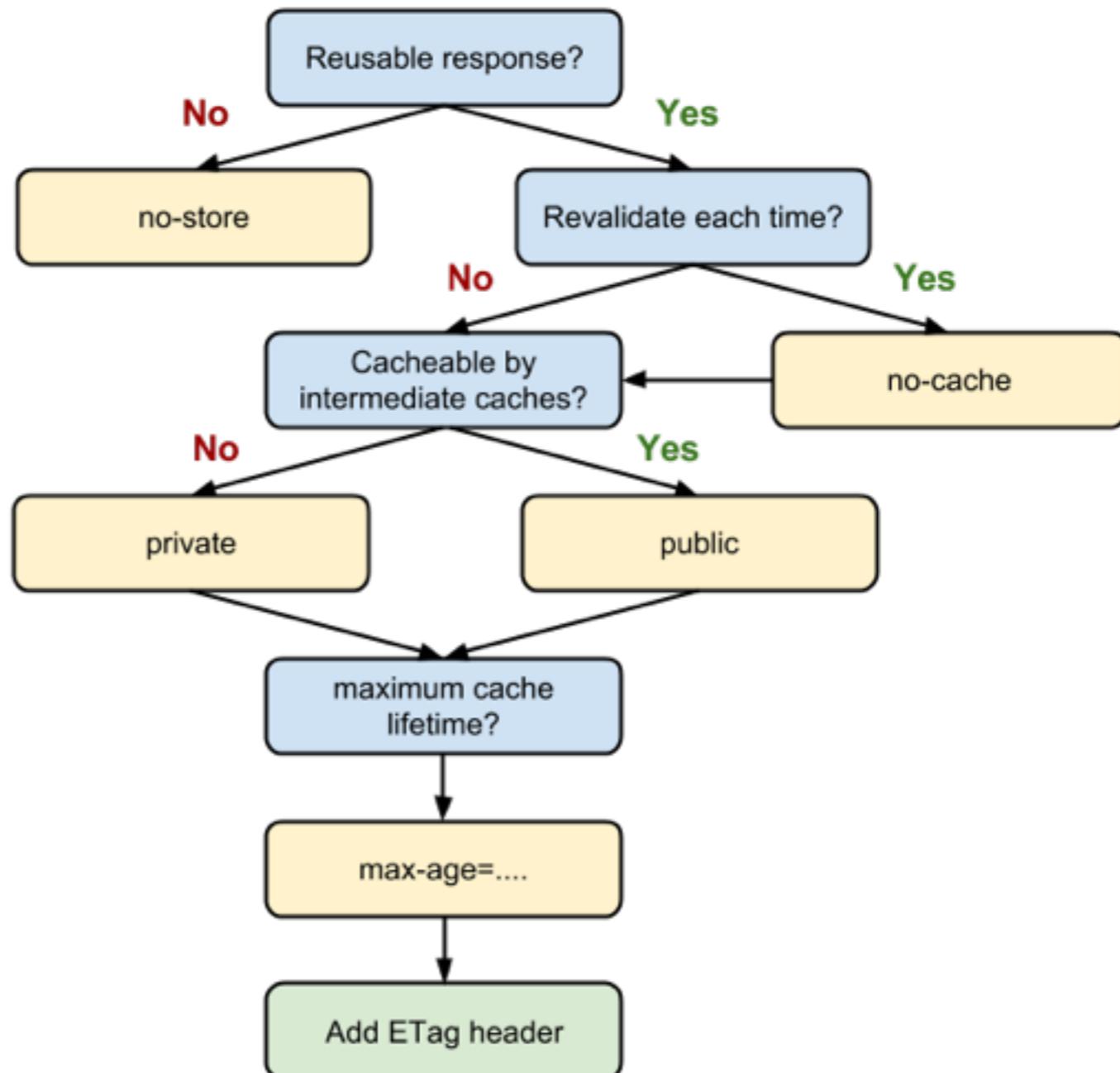
上传cdn, redis缓存, redirect

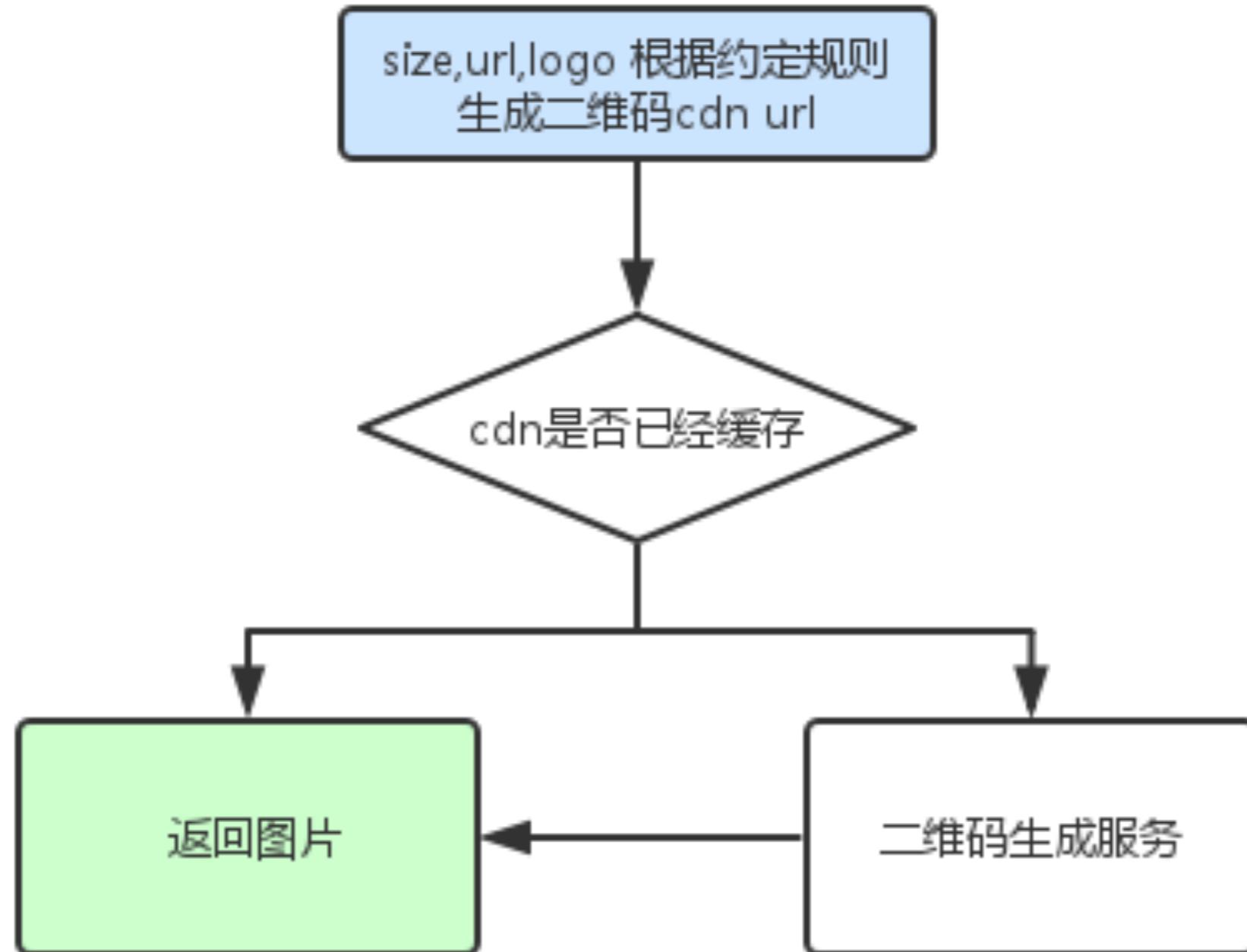
- 需要访问cdn前需要中转后端的接口
- redis缓存一旦失效就会重新生成一张二维码, cdn上存在很多重复的二维码图片



Http缓存 (如果不走cdn)

- Expires:
- Cache-Control: max-age=3600
- Last-Modified:
- If-Modified-Since:
- ETag:
- If-None-Match:



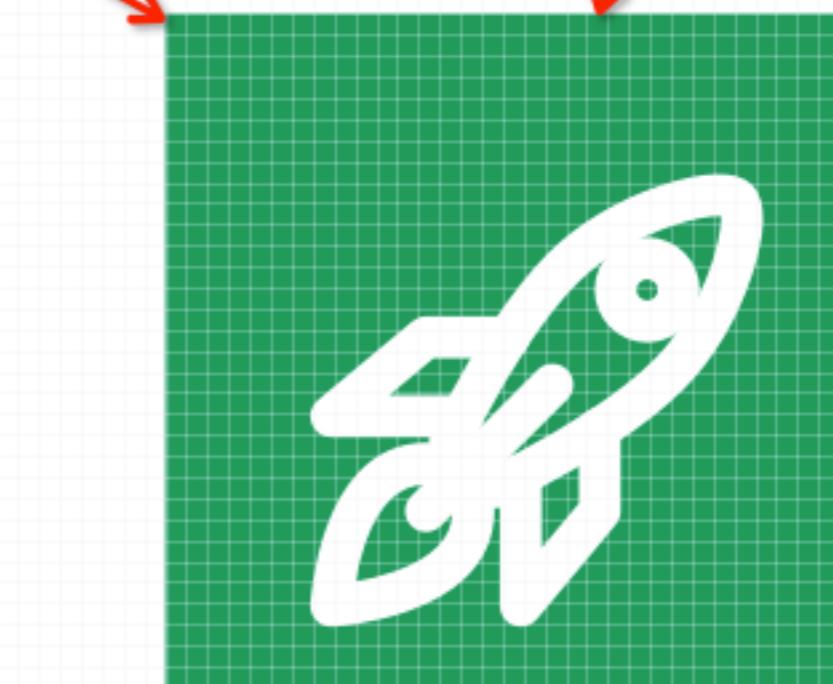


CDN回源

裁图服务

- 识别图片中心 (边界识别)

background pixel (n) diff



- 识别图片主要内容 (OpenCV)

上传服务

- 断点续传
- 文件识别 (Magic Bytes), 防隐写
- 上传(次数/速率)控制
- 事件处理
- 异常处理

```
34     upload_pass @backend;~  
35     # upload_store /tmp 1;~  
36     upload_store /tmp/upload;~  
37     upload_store_access user:r;~  
38     ~  
39     set $upload_field_name "image";~  
40     upload_resumable on;~  
41     ~  
42     upload_max_file_size 0;~  
43     upload_limit_rate 0; # 512k~  
44     ~  
45     upload_set_form_field $upload_field_name.name "$upload_file_name";~  
46     upload_set_form_field $upload_field_name.content_type "$upload_content_type";~  
47     upload_set_form_field $upload_field_name.path "$upload_tmp_path";~  
48     ~  
49     upload_aggregate_form_field "$upload_field_name.md5" "$upload_file_md5";~  
50     upload_aggregate_form_field "$upload_field_name.size" "$upload_file_size";~  
51     ~  
52     upload_pass_args on;~  
53     ~  
54     upload_cleanup 400 404 499 500-505;~  
55     }~  
56     ~  
57     location @backend {~  
58         return 200;~  
59     }~
```

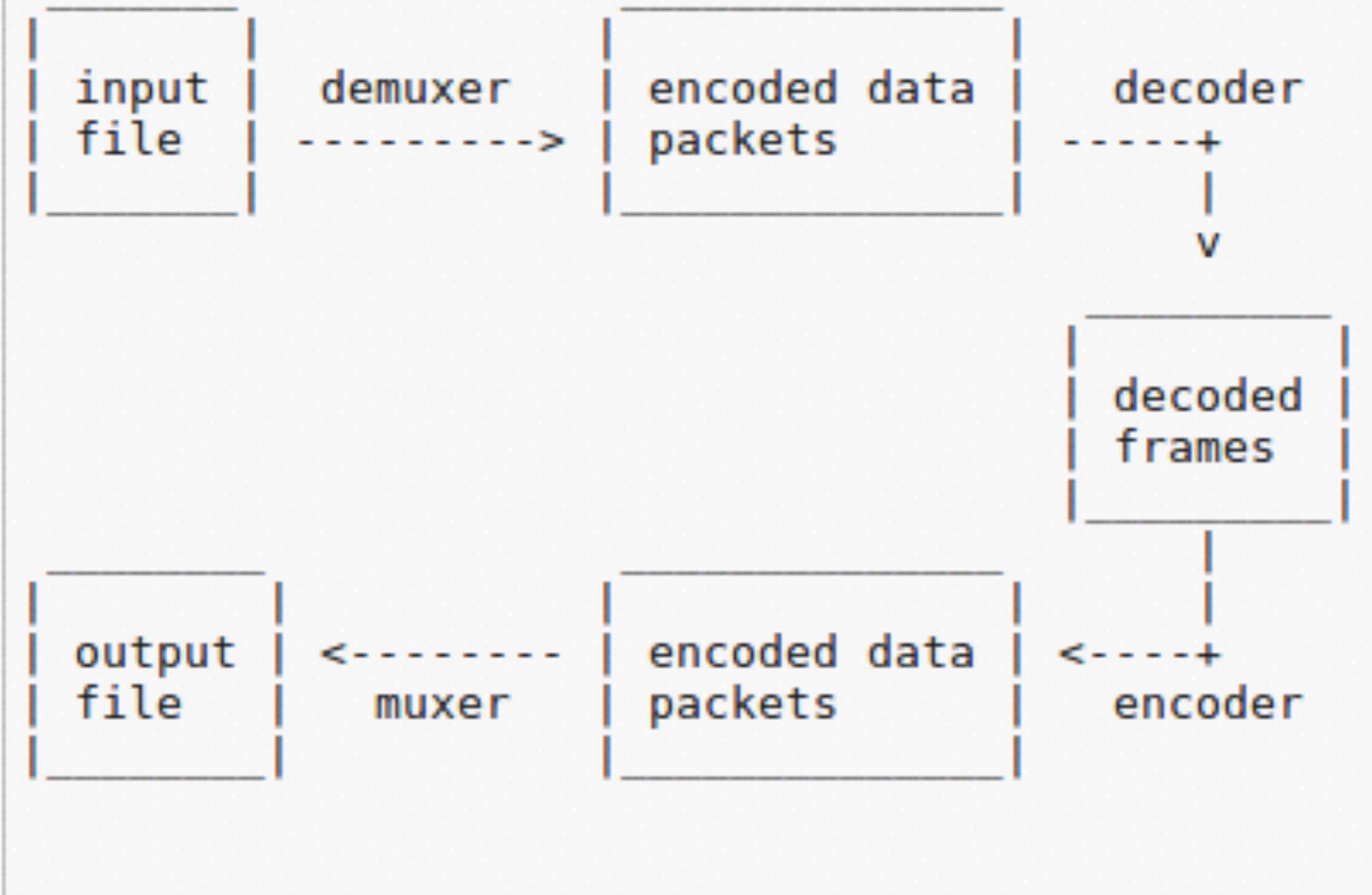
断点续传

断点续传的实现

- 分段上传
- 速率控制
- 残留文件清理

视频分发服务

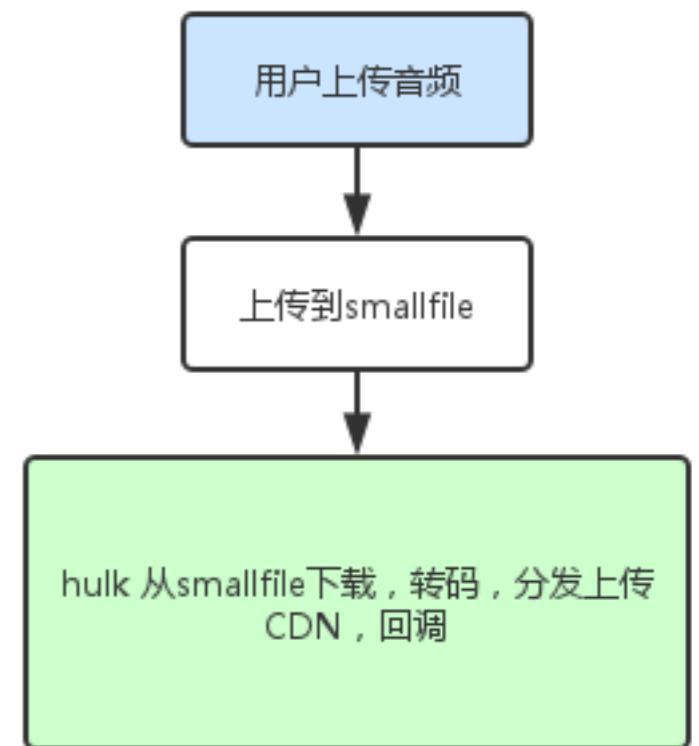
- 并发分发上传? 顺序分发上传? (带宽)
- 上传速率控制 (带宽)
- 视频搬家
- CDN回源



FFMPEG处理流程

音频转码服务

- ffmpeg (lame)



API的提供

- 实现跨域的3种方式 (jsonp, xhr2, iframe)
- 友好api url的实现
- 流程的可配置化

JSONP

实际上是在页面中动态插入一个`<script>`标签，然后在被加载的脚本中执行当前网页里的函数；只支持GET请求。

XHR2 [HTTP access control (CORS)]

- 简单请求
- 预检请求

http请求头

```
Origin: <origin>
Access-Control-Request-Method: <method>
Access-Control-Request-Headers: <field-name>[, <field-name>]*
```

http返回头

```
Access-Control-Allow-Origin: <origin> | *
Access-Control-Expose-Headers: X-My-Custom-Header, X-Another-Custom-Header
Access-Control-Max-Age: <delta-seconds>
Access-Control-Allow-Credentials: true | false
Access-Control-Allow-Methods: <method>[, <method>]*
```

```
static const auto maxPreflightCacheTimeout = std::chrono::seconds(600);
```

Preserve log Disable cache | No throttling

File data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other

[X](#) Headers Preview Response Timing

General

Request URL: http://media.mogujie.com/image/put?appKey=11m
Request Method: OPTIONS
Status Code: 200 OK
Remote Address: 221.228.216.89:80

Response Headers [view source](#)

Access-Control-Allow-Credentials: true
Access-Control-Allow-Headers: content-type
Access-Control-Allow-Methods: POST
Access-Control-Allow-Origin: http://www.xiaodian.com
Access-Control-Max-Age: 1800
Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH
Connection: keep-alive
Content-Length: 0
Date: Sat, 07 May 2016 10:21:16 GMT
Server: JuanNiuX/12.8.12
Vary: Origin
Z-Proxy: guomai013042
Z-Server: mofa130153

Request Headers [view source](#)

Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8,ja;q=0.6,nl;q=0.4,vi;q=0.2,zh-TW;q=0.2
Access-Control-Request-Headers: content-type
Access-Control-Request-Method: POST
Cache-Control: no-cache
Connection: keep-alive
Host: media.mogujie.com
Origin: http://www.xiaodian.com
Pragma: no-cache
Referer: http://www.xiaodian.com/pc/itemmanager/item/add?cid=11u4
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36

Query String Parameters [view source](#) [view URL encoded](#)

appKey: 11m

Chrome代码中写死了缓存的时间

Preserve log Disable cache | No throttling

Selected data URLs All | XHR JS CSS Img Media Font Doc WS Manifest Other

[×](#) Headers Preview Response Cookies Timing**General**

Request URL: <http://media.mogujie.com/image/put?appKey=11m>
Request Method: POST
Status Code: 200 OK
Remote Address: 221.228.216.89:80

Response Headers [view source](#)

Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: <http://www.xiaodian.com>

Cache-Control: no-store
Connection: keep-alive
Content-Encoding: gzip
Content-Language: zh-CN
Content-Type: application/json; charset=UTF-8
Date: Sat, 07 May 2016 10:21:16 GMT
Server: JuanNiuX/12.8.12
Transfer-Encoding: chunked
Vary: Origin
Vary: Accept-Encoding
Z-Proxy: guomai013042
Z-Server: mofa130153

Request Headers [view source](#)

Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN, zh;q=0.8, ja;q=0.6, nl;q=0.4, vi;q=0.2, zh-TW;q=0.2
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 83053
Content-Type: multipart/form-data; boundary=----WebKitFormBoundarybhyArj1ABbugxocN
Cookie: __mogujie=%2F2r%2BSDd%2FCsnqwWqSR7JwZTPKXpR0%2BDsaoClh8WUPzXMoIkjvYMgAInwlqzdVqHJCebQdgzBydYNGp5K%2FDtf06w%3D%3D; __ud_=12edk28; FRMS_FINGERPRINT=122144968aece94f667e01011991502f1bd97b0e7d802b01053aea91b53779013910000000000000012013d6a92ba82ba439d1371cce16c45e622d9ce1281d3d9446802a442591341c9f0f895fb98ab911401b326b5062b2f0e69132168934a3e9455fa7213451211731c83db8d2ff01b00912899e06ebc13bd881271d3d9446802a44259136100000000000000142100000000000000012511ff1de774005f8da130187374a5fff147c456124144968aece94f667e1311539da4c097a00000001381b326b5062b2f0e69; _OkLJ_%UJ=X3EKVZIXXQE BN6NW; _mg_tk=ebf1abe414; __mgjuuid=64c86b21-53bf-06c2-0385-87157b0bcc80
Host: media.mogujie.com
Origin: http://www.xiaodian.com

Pragma: no-cache
Referer: <http://www.xiaodian.com/pc/itemmanager/item/add?cid=11u4>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36

Query String Parameters [view source](#) [view URL encoded](#)

appKey: 11m

Request Payload

-----WebKitFormBoundarybhyArj1ABbugxocN
Content-Disposition: form-data; name="image"; filename="720x1080.jpg"
Content-Type: image/jpeg

-----WebKitFormBoundarybhyArj1ABbugxocN--

iframe

```
<iframe name="upload" frameborder="0" ...></iframe>
```

```
<form target="upload" action="/upload" ...></form>
```

- 在post的返回值中返回一段js代码，调用父页面的函数
- 限制：同源策略，`document.domain = tool.lu`

还能做什么？

- 图片的色彩聚类分析
- 相似图片的识别
- 根据图片识别商品标准类目