

内容分发网络 CDN

故障处理



腾讯云

【 版权声明 】

©2013–2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【 商标声明 】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

故障处理

故障概览

状态码说明及处理建议

不同节点缓存内容不一致

接入 CDN 后，网页访问速度慢

域名接入 CDN 后部分参数加载不了

域名接入CDN后，访问返回514

流量命中率偏低

CDN 域名突然出现404状态

CDN 访问报错423

页面展示异常-访问跨域报错

资源缓存未生效

故障处理

故障概览

最近更新时间：2024-08-22 16:08:31

您在使用 CDN 的过程中，可能会遇到一些不符合预期或报错的场景，我们整理了一些常见场景的说明和解决方案供您参考，后续我们会持续补充。

场景	场景描述
状态码说明以及处理意见	返回各类状态码。
不同节点缓存内容不一致	对 CDN 同一个资源 URL，不同地域的终端用户访问到 CDN 节点返回的内容不一致。
接入 CDN 后，网页访问速度慢	如果您使用腾讯云 CDN 后，网页访问速度依然很慢。
域名接入 CDN 后部分参数加载不了	域名接入 CDN 后部分参数加载不了。
流量命中率偏低	流量命中率偏低。
域名接入 CDN 后，访问返回514	域名接入 CDN 后，访问返回514状态码报错。
CDN 域名突然出现404状态	CDN 域名突然出现404状态。
CDN 访问报错423	域名接入 CDN 后，访问返回423状态码报错。

状态码说明及处理建议

最近更新时间：2024-08-22 17:34:14

以下为 CDN 内部状态码含义说明：

状态码	含义	处理建议
0	获取到响应给请求的状态码前，请求结束	请检查客户端是否过早的主动断开请求，或检查回源是否失败。
400	HTTP 请求语法错误 服务器无法解析	请检查请求语法是否正确。
403	拒绝访问	请检查 CDN 控制台 防盗链、鉴权配置、UA 黑白名单。
404	服务器无法返回正确信息	请检查源站是否正常或者源站信息、回源 HOST 配置是否发生变更。详细说明可见 CDN 域名突然出现404状态 。
413	POST 长度超出限制	请检查客户端 POST 内容大小（默认大小限制为 32MB）。
414	URL 长度超出限制	URL 默认大小限制为2KB。
423	回环请求	请检查回源跟随301/302配置，HTTPS 配置回源方式，源站 rewrite 的处理方式。详细说明可见 CDN 访问报错423 。
499	客户端主动断开连接	请检查客户端状态或超时时间设置。
504	网关超时	请与网站官方联系。
502	网关错误	请检查业务源站是否正常。
503	触发 COS 频控	请检查缓存配置或 COS 源站返回 no-cache/no-store。
509	触发 CC 攻击被封禁	域名遭受 CC 攻击已被封禁，可联系人工解封。
514	超出 IP 访问限频、黑名单拒绝访问	请检查 CDN 控制台 IP 访问限频配置、IP 黑白名单配置。详细说明可见 域名接入 CDN 后，访问返回514 。
524	触发平台访问过载	业务请求突发会触发平台过载，请评估业务量级向腾讯云报备，有疑问联系售后。
531	HTTPS 请求回源域名解析错误	请检查源站域名解析配置。
532	HTTPS 请求回源站建连失败	请检查源站443端口状态及证书配置或源站可用性。
533	HTTPS 请求回源站连接超时	请检查源站443端口状态及证书配置或源站可用性。
537	HTTPS 请求接受源站数据超时	请检查业务源站稳定性。
538	HTTPS 请求 SSL 握手失败	请检查源站协议和算法的兼容性。
539	HTTPS 请求证书校验失败	请检查源站证书是否正常配置（是否过期、是否证书链齐全）。
540	HTTPS 请求证书域名校验不通过	请检查源站证书是否正常配置。
562	HTTPS 请求建连失败	请 联系我们 并提供 X-NWS-LOG-UUID 信息或 提交工单 进行排查。
563	HTTPS 请求连接超时	请 联系我们 并提供 X-NWS-LOG-UUID 信息或 提交工单 进行排查。
564	HTTPS 请求回源失败	若配置为 HTTP 回源方式，请检查源站负载及带宽使用率，或源站访问限制。若配置为协议跟随方式，请检查源站443端口状态及证书配置。
567	节点接收文件时，响应超时	请 联系我们 并提供 X-NWS-LOG-UUID 信息或 提交工单 进行排查。

不同节点缓存内容不一致

最近更新时间：2025-01-13 11:09:52

现象描述

对 CDN 同一个资源 URL，不同地域的终端用户访问到 CDN 节点返回的内容不一致。

可能原因

- 原因一：命中了域名配置的缓存键规则 - 忽略全部参数，同时源站设置了根据参数吐出不同的资源。
由于源站是根据参数不同输出不同数据，而 CDN 是忽略参数进行缓存，这就导致不同的节点可能由于第一次收到的访问带的参数不同建立了不同的缓存。下一次当同一个请求访问到不同节点，收到的缓存返回的数据也是不一样。
- 原因二：源站同一个资源更新后没有做刷新处理。
CDN 是按 URL 进行资源缓存的。如果源站更新文件后，URL 没有变化，只是内容发生变化，访问时如果节点有缓存还是会直接命中缓存。同时，由于各个地域访问热度不一，淘汰时间不一，有的节点缓存已经淘汰，再次访问时，会回源站拉取新的资源，从而这导致各个节点的缓存可能出现新老版本同时存在，不同节点缓存内容不一致的情况。

解决思路

1. 确保源站根据 URL 参数吐出不同的资源和 CDN 域名配置的缓存键规则-忽略全部参数不同时使用。
2. 确保源站同一个 URL 的资源更新以后统一做刷新处理。

处理步骤

1. 根据自身业务情况，判断源站是否根据 URL 参数吐出不同的资源。
 - 是，请执行 [步骤2](#)。
 - 否，请直接执行 [步骤4](#)。
2. 登录 [CDN 控制台](#)，选择[域名管理](#)找到对应的域名配置，查看[缓存配置](#) > [缓存键规则配置](#)的“忽略参数”项：检查 CDN 配置域名是否开启忽略参数缓存功能。



3. 在缓存键规则配置中对应规则的操作栏，单击[修改](#)，在弹出的“修改规则”框关闭忽略参数功能，然后单击[保存](#)。

忽略参数 不忽略 全部忽略 保留指定参数

说明

如果用户不方便全部关闭，这里 CDN 也提供了保留指定参数的忽略功能，用户也可以根据实际的业务需求进行选择使用。具体用法可参见 [缓存键规则配置](#)。

4. 进入刷新预热目录，对源站变更的资源进行刷新。

刷新预热

- URL刷新**
- 目录刷新
- URL预热
- 操作记录

[URL刷新API](#)

URL

请在此输入需要刷新的URL (需要http://或https://), 一行一个, 例如: http://www.test.com/test.html

0/1000

暂不支持提交包含通配符的URL刷新任务
 今日剩余URL刷新量10000个 (境内)
 今日剩余URL刷新量10000个 (境外)

URL Encode 开启后, 将自动对带有特殊字符的url进行urlencode.

提交并刷新

说明

用户也可以采用 API 的方式进行刷新, 这样当源站出现变更时, 绑定调用 API 进行刷新, 可以第一时间保证全网变更资源访问内容的一致性。详细可参见 [URL 刷新接口](#) 和 [目录刷新接口](#)。

接入 CDN 后，网页访问速度慢

最近更新时间：2024-08-22 11:56:01

现象描述

如果您使用腾讯云 CDN 后，网页访问速度依然很慢。

可能原因

- 原因一：您接入域名的 CDN 加速服务未生效，可能原因是您没有在域名 DNS 服务商处配置 CNAME 记录。请执行 [检查域名解析](#)。
- 原因二：节点缓存过期时间配置错误。请执行 [检查节点缓存过期时间配置](#)。
- 原因三：首次访问资源，且之前未对该资源做过预热处理。请执行 [进行 URL 预热](#)。
- 原因四：网页架构模式本身存在缺陷。请执行 [优化网页架构模式](#)。

解决思路

检查域名解析

以下是一个用 nslookup 命令查询 CDN 加速域名 DNS 解析示例：

```
nslookup 加速域名
```

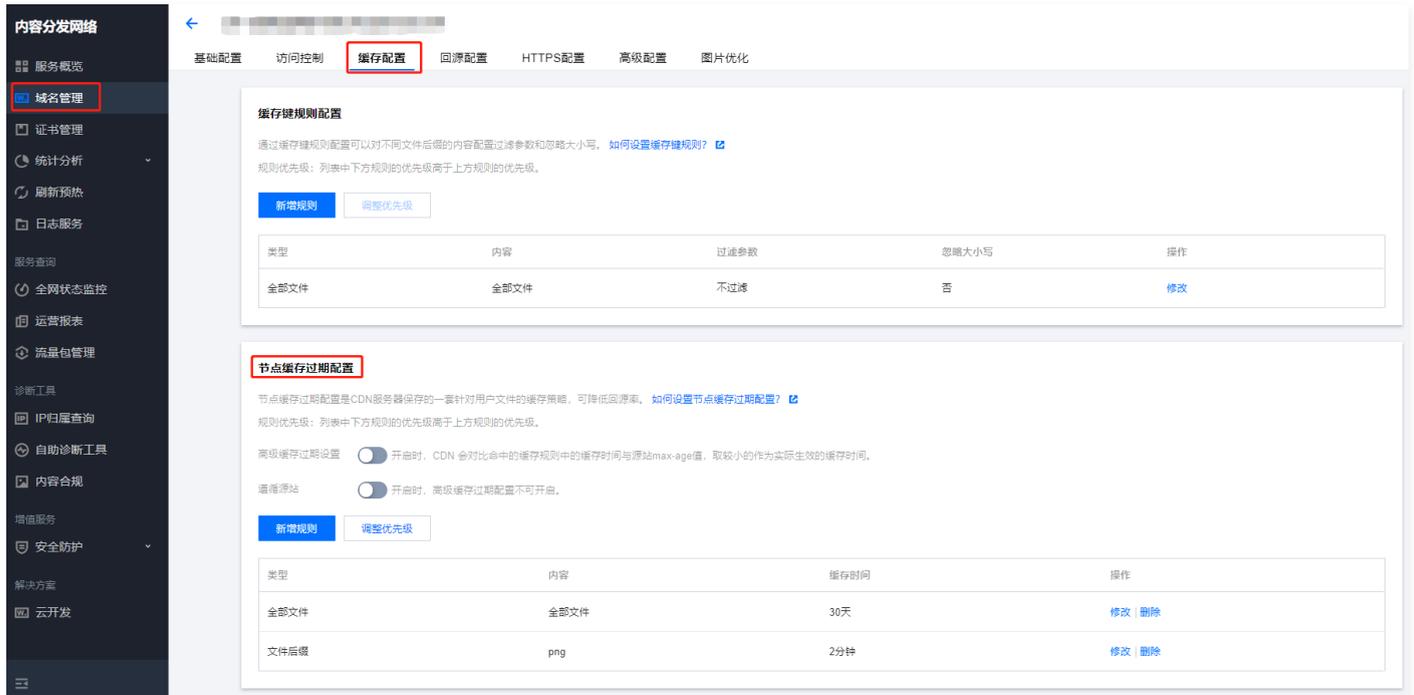
```
C:\Users\...\.com
服务器: ...
Address: ...

非权威应答:
名称: ...
Address: ...
Aliases: ... .com
          ... .com cdn.dnsv1.com
```

若查询的域名解析中没有上图红框后缀为 dnsv1.com 的 CNAME 解析记录，则说明您接入域名的 CDN 加速服务未生效，可能原因是您没有在域名 DNS 服务商处配置 CNAME 记录，可以根据 [配置 CNAME](#) 文档前往您的域名 DNS 服务商处配置 CNAME 记录。

检查节点缓存过期时间配置

登录 [CDN 控制台](#)，在左侧菜单栏选择**域名管理**，单击**域名操作列的管理**，进入域名配置页面，切换 Tab 至**缓存配置**，即可找到**节点缓存过期配置**。



- 检查所访问的资源对应的节点缓存规则，是否存在配置的节点缓存过期时间为0、节点缓存过期时间过短或不缓存的情况。若 CDN 节点没有缓存，访问请求会回源，起不到加速效果。建议用户根据需要配置节点的缓存时间。
- 检查您的源站是否设置了缓存头部 Cache-Control 为 no-store/no-cache/private。
 - 若源站设置了缓存头部 Cache-Control 为 no-store/no-cache/private，此时需同时开启“强制缓存”，CDN 节点才会按照所配置的缓存时间缓存资源。
 - 若未开启“强制缓存”且源站的 Cache-Control 字段为 no-cache/no-store/private，则即使配置了缓存时间，CDN 节点也不会缓存资源。

更多配置规则请前往 [节点缓存过期时间配置](#)。

进行 URL 预热

如果您是首次访问资源，且之前未对该资源做过预热处理，CDN 节点会回源拉取资源，首次访问速度慢属于正常。建议登录 [CDN 控制台](#)，在**刷新预热**中找到 URL 预热功能，进行 [URL 预热](#)。



优化网页架构模式

网页动态资源较多，每次访问都会回源拉取最新资源，影响访问速度。若网页动态资源占比多，建议优化源站，将动态资源与静态资源分开，静态资源使用 CDN 分发加速。

域名接入 CDN 后部分参数加载不了

最近更新时间：2024-08-22 17:34:14

现象描述

域名接入 CDN 后部分参数加载不了。

可能原因

CDN 对应的加速域名开启了忽略参数的功能，该功能如果开启则请求到 CDN 节点后，会截取到没有参数的 URL 向源站请求，并且 CDN 节点仅保留一份副本，导致域名接入 CDN 后部分参数加载不了。

如果用户回源的很多资源是通过 URL 中“?”后面的参数来区分和获取，则需要关闭该参数。

解决思路

CDN 对应的加速域名关闭忽略参数的功能，则每个不同的 URL 都缓存不同的副本在 CDN 的节点上。

处理步骤

1. 登录 [CDN 控制台](#)，选择[域名管理](#)找到对应的域名配置，查看[缓存配置](#) > [缓存键规则配置](#)的“忽略参数”项。

The screenshot shows the 'Cache Key Rule Configuration' page in the Tencent Cloud CDN console. The left sidebar contains navigation options like 'Service Overview', 'Domain Management', 'Certificate Management', 'Statistics Analysis', 'Refresh Preheat', 'Log Service', 'Service Query', and 'Global Status Monitoring'. The main content area has tabs for 'Basic Configuration', 'Access Control', 'Cache Configuration', 'Origin Configuration', 'HTTPS Configuration', 'Advanced Configuration', and 'Image Optimization'. Under 'Cache Configuration', there is a 'Cache Key Rule Configuration' section. It includes a description, a link to 'How to Set Cache Key Rules?', and a note about rule priority. Below this are two buttons: 'Add Rule' and 'Adjust Priority'. A table lists the current rule with columns for 'Type', 'Content', 'Ignore Parameters', 'Ignore Case', and 'Action'. The 'Ignore Parameters' column is highlighted with a red box, and the 'Action' column contains a 'Modify' link.

类型	内容	忽略参数	忽略大小写	操作
全部文件	全部文件	不忽略	否	修改

2. 在[缓存键规则配置](#)对应规则的操作栏，单击[修改](#)，在弹出的“修改规则”框关闭忽略参数功能，然后单击[保存](#)。

忽略参数 不忽略 全部忽略 保留指定参数

说明

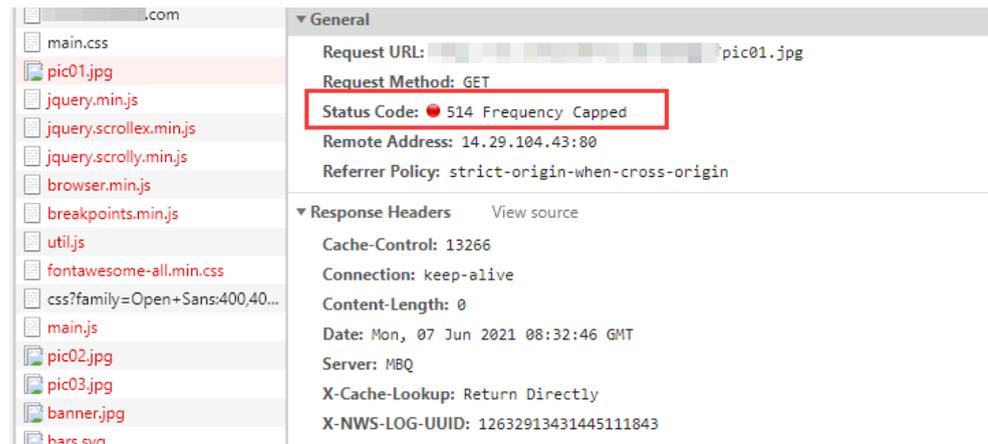
CDN 也提供了保留指定参数的忽略功能，用户可以根据实际的业务需求选择使用。具体用法可参见 [缓存键规则配置](#)。

域名接入CDN后，访问返回514

最近更新时间：2024-08-22 17:34:14

现象描述

域名接入 CDN 后，访问返回514状态码报错。



可能原因

1. 没有启用 HTTPS 服务，使用了 HTTPS 访问导致返回514；
2. 命中了 IP 访问限频配置，IP 访问限频设置针对单 IP 单节点每秒访问次数进行了限制，若超出限制，则会直接返回514；
3. 命中了 IP 黑白名单配置，若用户端访问的 IP 命中配置黑名单内 IP 或白名单外 IP 均返回514。

处理步骤

1. 若使用了 HTTPS 访问可先在 CDN 控制台确认是否启用 HTTPS 服务，若没有开启可将服务打开并完成配置证书。也可参考 [HTTPS 配置指南](#) 进行配置。



2. 依次确认控制台 IP 访问限频配置、IP 黑白名单配置规则，若命中其中一项您可根据业务情况进行合理调整。

流量命中率偏低

最近更新时间：2025-01-13 11:09:52

现象描述

实时监控中流量命中率的数值偏低，不符合预期。

可能原因

- 进行了缓存刷新
缓存刷新会清空节点上指定内容，短时间会出现命中率下降的情况。
- 源站含新资源
源站新资源较多，会引起 CDN 节点回源，流量命中率出现下降趋势。
- 源站异常
若源站出现异常，5XX或4XX较多时，也会影响流量命中率。
- 缓存策略配置不当
请根据您的实际业务情况配置缓存规则。
- 关闭分片回源
若关闭了分片回源，导致回源时拉取整个大文件，而不是按照请求时分片拉取，会拉高回源流量，从而影响流量命中率。
- 域名配置的缓存键规则-忽略参数为：不忽略
若在您的业务场景下，资源 URL 路径中问号后的参数对资源内容有影响，则需配置忽略参数为不忽略；反之，若参数对资源内容没有影响，则需配置忽略参数为全部忽略。

解决思路

1. 检查您的源站，确保源站无异常。
2. 若您进行了缓存刷新或源站新资源较多，此为正常现象。
3. 确保源站根据 URL 参数吐出不同的资源和 CDN 域名配置的缓存键规则 - 忽略全部参数不同时使用。
4. 根据实际业务情况配置缓存规则。

处理步骤

1. 检查您的源站是否异常或是否进行了缓存刷新。
 - 是，命中率为正常现象。
 - 否，请执行 [步骤2](#)。
2. 根据自身业务情况，判断源站是否根据 URL 参数吐出不同的资源。
 - 是，请执行 [步骤3](#)。
 - 否，请执行 [步骤5](#)。
3. 登录 [CDN 控制台](#)，选择[域名管理](#)找到对应的域名配置，查看[缓存配置](#) > [缓存键规则配置](#)的“忽略参数”项：检查 CDN 配置域名是否开启忽略参数缓存功能。

基础配置 访问控制 **缓存配置** 回源配置 HTTPS配置 高级配置 图片优化

缓存键规则配置

通过缓存键规则配置可以对不同文件类型的内容配置过滤参数和忽略大小写。 [如何设置缓存键规则?](#)

规则优先级：列表中下方规则的优先级高于上方规则的优先级。

[新增规则](#) [调整优先级](#)

类型	内容	忽略参数	忽略大小写	操作
全部文件	全部文件	不忽略	否	修改

- 是，请执行 [步骤4](#)。

否，请执行 [步骤5](#)。

4. 在缓存键规则配置中对应规则的操作栏，单击**修改**，忽略参数配置选择：**全部忽略**，然后单击**保存**。



说明

如果用户不方便全部关闭，这里 CDN 也提供了保留指定参数的忽略功能，用户也可以根据实际的业务需求进行选择使用。具体用法可参见 [缓存键规则配置](#)。

5. 登录 [CDN 控制台](#)，选择**域名管理**找到对应的域名配置，查看**缓存配置 > 节点缓存过期配置**，请您查看缓存规则是否符合自身业务和源站的实际情况。

是，请执行 [步骤5](#)。

否，请参考 [节点缓存过期配置](#) 调整您的缓存规则。

CDN 域名突然出现404状态

最近更新时间：2024-08-22 17:34:15

现象描述

访问 CDN 域名突然出现404状态。

可能原因

1. 源站异常。
2. 控制台源站信息、回源 HOST 配置发生了改动。

解决思路

1. 检查您的源站，确保源站无异常。
2. 检查控制台源站信息、回源 HOST 配置，确保相关配置无异常。

处理步骤

1. 检查源站是否出现异常。

- 是，修复源站。
- 否，请执行 [步骤2](#)。

2. 检查控制台源站信息以及回源 HOST 配置。

登录 [CDN 控制台](#)，选择[域名管理](#)找到相应的域名，查看[基础配置](#) > [源站信息](#)的“源站地址”、“回源 HOST”，确保该处配置正确。

- 源站类型：

自有源站	若您选择自有源站，需要提供可正常访问的业务服务器的 IP 地址或域名
COS 源	若您选择腾讯云对象存储中的一个存储桶作为源站，根据存储桶处的配置选择默认域名或静态网站场景；若您的存储桶为私有桶，请授权 CDN 并开启回源鉴权，即开启私有存储桶访问
第三方对象存储	若您选择第三方对象存储，请输入有效的存储桶访问地址作为源站，当前支持的第三方为 AWS S3 和阿里云 OSS。回源至第三方私有存储桶，需填写有效密钥并开启回源鉴权，即开启私有存储桶访问

- 回源 HOST：即回源域名，CDN 节点在回源时，访问的源站 IP 地址下具体的站点域名。默认为当前加速域名，若接入泛域名，则默认为泛域名，且实际回源 HOST 为访问域名。您可根据实际业务情况自行修改（注：源站类型为 COS 源、第三方对象存储时不可修改）。

更多源站配置说明，详情请参见 [源站配置](#)。

CDN 访问报错423

最近更新时间：2024-08-22 17:08:17

现象描述

域名接入 CDN 后，访问返回423状态码和 Forward Loop Detect 报错。

可能原因

- 源站地址为已接入的 CDN 加速域名，造成了循环解析，无法正常回源。
- 源站配置了 HTTP 请求到 HTTPS 301/302 跳转，且 CDN 控制台开启了回源跟随 301/302 配置，可能会造成访问423。

解决思路

1. 确保您的源站地址与腾讯云 CDN 加速域名不相同。
2. 若源站使用 HTTP 请求到 HTTPS 301/302 跳转，且 CDN 控制台开启了回源跟随 301/302 配置，需控制台开启 HTTPS 配置，强制跳转 HTTPS，且回源方式修改为协议跟随。
3. 域名访问/后源站301/302到/index.html，CDN 控制台同时开启了回源跟随301/302配置，导致302跳转死循环，需关闭回源跟随 301/302 配置。

处理步骤

1. 登录 [CDN 控制台](#)，选择[域名管理](#)找到对应的域名配置，查看[基础配置 > 源站信息](#)，检查您的源站地址是否与该账户下其他加速域名相同。



- 是，则单击[编辑](#)，将源站地址替换为非腾讯云 CDN 加速域名。



- 否，请执行 [步骤2](#)。

2. 如果源站配置了 HTTP 请求到 HTTPS 301/302 跳转，且 CDN 控制台开启了回源跟随 301/302 配置，则建议开启 HTTPS 配置，强制跳转 HTTPS，且回源方式修改为协议跟随。
 - 开启 HTTPS 配置，强制跳转 HTTPS：登录 [CDN 控制台](#)，选择[域名管理](#)找到对应的域名，查看 [HTTPS 配置 > 强制跳转](#)，启用该功能。



- **修改回源方式为协议跟随**：登录 [CDN 控制台](#)，选择**域名管理**找到对应的域名，查看**基础配置 > 源站信息**，单击**编辑**，修改回源协议为“协议跟随”。

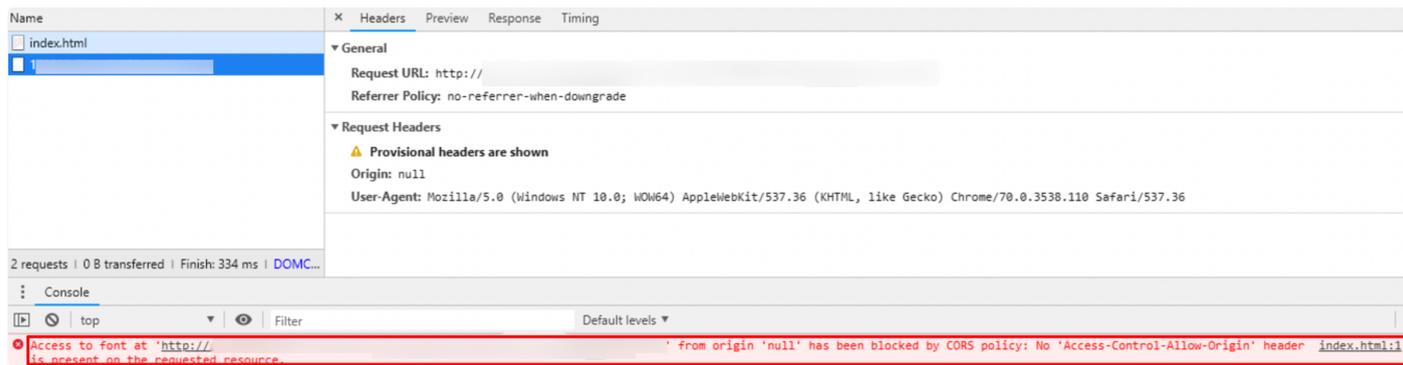


页面展示异常-访问跨域报错

最近更新时间：2024-11-15 16:22:03

现象描述

前端报跨域错误，导致页面错误或展示异常等问题，如下图：



可能原因

跨域是由于浏览器的同源策略限制，同时也为了网页的安全性考虑，当通过脚本向不同的来源发送请求时，这个请求的响应会被浏览器拦截，从而导致前端报错或页面无法正常展示。而当一个请求URL的协议、域名、端口三者之间任意一个与当前页面URL不同即为跨域。

解决思路

1. 确认页面异常报错是否是由跨域造成的，如下所示。



2. 在 CDN 控制台配置对应的 HTTP 响应头部，定义允许访问该资源的域。

处理步骤

1. 登录 [CDN 控制台](#)，在对应的域名管理-高级配置-HTTP 响应头配置下，设置好 Access-Control-Allow-Origin 头部参数即可。如下图，即允许所有域名发起的跨域请求。详情请见 [Access-Control-Allow-Origin 匹配模式介绍](#)。



2. 或针对性的设置为单个或多个已知的允许发起跨域请求的域名/IP，如下所示。

同时也可以根据业务需要，加上如 Access-Control-Request-Method、Access-Control-Request-Headers、Access-Control-Max-

Age 等头部参数，来限制浏览器所能接受的请求的方法、可携带的头、预请求的有效时间等等。详情请见 [参数支持列表](#)。

HTTP响应头配置

HTTP响应头配置会影响客户程序（浏览器）的响应行为。 [什么是HTTP响应头配置？](#)

配置状态 关闭状态下仍可修改下方配置，但不会发布至现网，仅当开启此开关时，进行现网配置下发

[新增规则](#) [调整优先级](#)

头部操作	头部参数	头部取值	操作
设置	Access-Control-Allow-Origin	http://test.com,http://1.1.1.1	修改 删除

注意
若您已在 COS 侧存储桶配置了跨域访问，为保证正常访问，请在 CDN 控制台 [HTTP 响应头](#) 同步配置跨域规则。

参数支持列表

头部参数	说明
Access-Control-Allow-Origin	用于解决资源的跨域权限问题，域值定义了允许访问该资源的域。若来源请求 Host 在域名配置列表之内，则直接填充对应值在返回头部中。也可以设置通配符“*”，允许被所有域请求。更多说明请见 Access-Control-Allow-Origin 匹配模式介绍 。支持输入“*”，或多个域名 / IP / 域名与 IP 混填（必须包含 http:// 或 https://，填写示例： <code>http://test.com,http://1.1.1.1</code> ，逗号隔开）（注意：输入框最多可输入1000字符）。
Access-Control-Allow-Methods	用于设置跨域允许的 HTTP 请求方法，可同时设置多个方法，如下： <code>Access-Control-Allow-Methods: POST, GET, OPTIONS</code> 。
Access-Control-Max-Age	用于指定预请求的有效时间，单位为秒。非简单的跨域请求，在正式通信之前，需要增加一次 HTTP 查询请求，称为“预请求”，用来查明这个跨域请求是不是安全可以接受的，如下请求会被视为非简单的跨域请求：以 GET、HEAD 或者 POST 以外的方式发起，或者使用 POST，但是请求数据类型为 application / x-www-form-urlencoded、multipart / form-data、text / plain 以外的数据类型，如 application / xml 或者 text / xml。使用自定义请求头为： <code>Access-Control-Max-Age: 1728000</code> ，表明在 1728000秒（20天）内，对该资源的跨域访问不再发送另外一条预请求。
Access-Control-Expose-Headers	用于指定哪些头部可以作为响应的一部分暴露给客户端。默认情况下，只有6种头部可以暴露给客户端： <code>Cache-Control</code> 、 <code>Content-Language</code> 、 <code>Content-Type</code> 、 <code>Expires</code> 、 <code>Last-Modified</code> 、 <code>Pragma</code> 。如果想让客户访问到其他的头部信息，可以进行如下设置，当输入多个头部时，需用“,”隔开，如： <code>Access-Control-Expose-Headers: Content-Length,X-My-Header</code> ，表明客户端可以访问到 <code>Content-Length</code> 和 <code>X-My-Header</code> 这两个头部信息。

Access-Control-Allow-Origin 匹配模式介绍

匹配模式	域值	说明
全匹配	*	设置为 * 时，则响应添加头部： <code>Access-Control-Allow-Origin: *</code>
固定匹配	<code>http://cloud.tencent.com</code> <code>https://cloud.tencent.com</code> <code>http://www.b.com</code>	来源 <code>https://cloud.tencent.com</code> ，命中列表，则响应添加头部： <code>Access-Control-Allow-Origin: https://cloud.tencent.com</code> 来源为 <code>https://www.qq.com</code> ，未命中列表，响应无变化。
二级泛域名匹配	<code>https://*.tencent.com</code>	来源 <code>https://cloud.tencent.com</code> ，命中列表，则响应添加头部： <code>Access-Control-Allow-Origin: https://cloud.tencent.com</code> 来源为 <code>https://cloud.qq.com</code> ，未命中列表，响应无变化。
端口匹配	<code>https://cloud.tencent.com:8080</code>	来源为 <code>https://cloud.tencent.com:8080</code> ，命中列表，则响应添加头部： <code>Access-Control-Allow-Origin:https://cloud.tencent.com:8080</code> 来源为 <code>https://cloud.tencent.com</code> ，未命中列表，响应无变化。

⚠ 注意

若存在特殊端口，则需要在列表中填写相关信息，不支持任意端口匹配，必须指定。

资源缓存未生效

最近更新時間：2024-08-22 14:42:47

現象描述

設置完節點緩存過期時間，預熱完成后，請求依然未能命中節點緩存。

可能原因

1. 設置有多條緩存配置，但不清楚其優先級。
2. 配置了遵循源站的緩存策略，但源站的 Cache-Control 字段為 `no-cache/no-store/private`。

解決思路

1. 正確設置緩存優先級
CDN 緩存規則可以設置多條，並且底部優先，這裡需要確保用戶預期和優先級保持一致，才能保證客戶預期的規則生效。
2. 正確設置緩存時間
檢查控制台的緩存時間是否過小。

⚠ 注意

文件訪問頻率低，熱度不夠，不經常被用戶訪問到的 URL，即使符合所有緩存規則，但是也有被節點去除緩存的風險。

3. 檢查緩存設置規則是否符合預期
 - 檢查 CDN 緩存鍵規則是否設置參數緩存規則導致的節點未緩存。
 - 檢查 CDN 節點過期緩存設置是否設置了強制不緩存。
 - 檢查 CDN 節點過期緩存設置遵循源站時，回源時源站的頭部是否返回了 `no-cache/no-store/private`。

處理步驟

1. 檢查緩存設置優先（底部優先）

登錄 [CDN 控制台](#)，在左側菜單欄選擇**域名管理**，單擊**域名操作列的管理**，進入域名配置頁面，切換 Tab 至**緩存配置**，即可找到**緩存鍵規則配置**。如下圖所示，jpg 不忽略參數緩存優先級高於全部文件忽略參數緩存，需要確保業務緩存策略符合優先級設置。

緩存鍵規則配置

通過緩存鍵規則配置可以對不同文件後綴的內容配置忽略參數和忽略大小寫。[如何設置緩存鍵規則？](#)

規則優先級：列表中下方規則的優先級高於上方規則的優先級。

[新增規則](#) [調整優先級](#)

類型	內容	忽略參數	忽略大小寫	操作
全部文件	全部文件	全部忽略	否	修改
文件後綴	jpg	不忽略	否	修改 刪除

2. 檢查緩存時間

登錄 [CDN 控制台](#)，在左側菜單欄選擇**域名管理**，單擊**域名操作列的管理**，進入域名配置頁面，切換 Tab 至**緩存配置**，即可找到**節點緩存過期配置**。如下圖所示如果緩存設置時間過小，可能會誤以為緩存設置沒有生效，確保符合業務緩存策略。

節點緩存過期配置

節點緩存過期配置是 CDN 服務器保存的一套針對用戶文件的緩存策略，可降低回源率。[如何設置節點緩存過期配置？](#)

規則優先級：列表中下方規則的優先級高於上方規則的優先級。

[新增規則](#) [調整優先級](#)

類型	內容	緩存行為	操作
文件後綴	jpg	遵循源站	修改 刪除
全部文件	全部文件	緩存 1 秒	修改 刪除

共 2 條 10 條 / 頁 [◀](#) [▶](#) 1 / 1 頁 [▶](#)

3. 检查缓存策略

在缓存键规则配置和节点缓存过期配置里对策略进行检查，确保符合预期。

如果设置了遵循源站，确保源站的 Cache-Control 字段不为 `no-cache/no-store/private`。

4. 将需要缓存的资源重新预热一遍，等待预热完成后，再次请求即可。