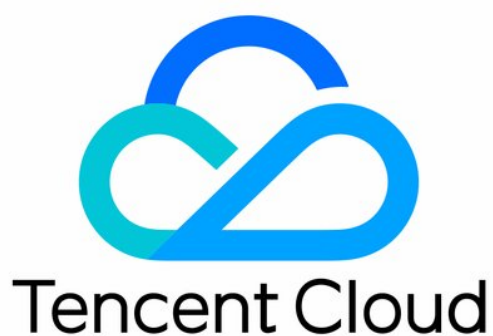


Content Delivery Network

障害処理

製品ドキュメント



Copyright Notice

©2013-2024 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

カタログ：

障害処理

ステータスコードの説明と推奨する対処方法

異なるノードキャッシュのコンテンツが不一致

CDNにアクセスすると、ウェブページのアクセス速度が遅くなります

トラフィックのヒット率が低い場合

CDNドメイン名に突然、404エラーが発生しました

ページ表示の異常-アクセスクロスドメインエラー

リソースキャッシュは有効化されていません

障害処理

ステータスコードの説明と推奨する対処方法

最終更新日：：2021-10-08 10:24:11

以下、CDN内部ステータスコードの意味について説明します。

ステータスコード	意味	推奨する対処方法
400	HTTPリクエストの構文エラー サーバーで解決できません	リクエスト構文が正しいかどうかを確認してください。
403	リクエストが拒否されました	refererのブラックリスト/ホワイトリスト、IPブラックリスト/ホワイトリスト、認証設定などのアクセス制御機能が設定されているかどうかをチェックしてください。
404	サーバーが正確な情報を返しません	オリジンサーバーが正常かどうか、またはオリジンサーバー情報、back-to-origin HOSTの設定が変更されていないかどうかをチェックしてください。
413	POSTの長さが制限を超えています	クライアントPOSTのコンテンツサイズをチェックしてください（デフォルトのサイズ制限は32MBです）。
414	URLの長さが制限を超えています	URLのデフォルトのサイズ制限は2KBです。

ステータスコード	意味	推奨する対処方法
423	ループ バック リクエ スト	back-to-originの301/302追従設定、HTTPS設定のback-to-origin方式、オリジンサーバーrewriteの対処方法をチェックしてください。
499	クライ アント が自主 的に接 続を切 断しま した	クライアントステータスまたはタイムアウト時間の設定をチェックしてください。
502	ゲート ウェイ エラー	ビジネスオリジンサーバーが正常かどうかをチェックしてください。
503	COS周 波数制 御がトリ ガーさ れました	キャッシュ設定またはCOSオリジンサーバーがno-cache/no-storeを返していることをチェックしてください。
509	CC攻撃 のため ロック がトリ ガーさ れました	ロックを解除するため、 お問い合わせ または オンラインサポート までご連絡ください。
514	IPアク セス頻 度制限 を超え ました	CDNコンソールでIPアクセス頻度制限の設定をチェックしてください。

ステータスコード	意味	推奨する対処方法
531	HTTPS リクエ ストの back-to- originド メイン 名解決 エラー	オリジンサーバーのドメイン名解決設定をチェックしてください。
532	HTTPS リクエ ストの back-to- origin サー バーと の接続 の確立 に失敗 しまし た	オリジンサーバー443ポートのステータスおよび証明書の設定またはオリジンサーバーの可用性をチェックしてください。
533	HTTPS リクエ ストの back-to- origin サー バーと の接続 がタイ ムアウ トしま した	オリジンサーバー443ポートのステータスおよび証明書設定またはオリジンサーバーの可用性をチェックしてください。

ステータスコード	意味	推奨する対処方法
537	HTTPS リクエ ストの オリジ ンサー バーの データ 受信が タイム アウト しまし た	ビジネスオリジンサーバーの安定性をチェックしてください。
538	HTTPS リクエ ストの SSLハ ンド シェイ クに失 敗しま した	オリジンサーバープロトコルとアルゴリズムの互換性をチェックしてください。
539	HTTPS リクエ ストの 証明書 の検証 に失敗 しまし た	オリジンサーバーの証明書の設定が正常かどうかをチェックしてください (期限切れではないか、証明書リンクが完全かどうかなど)。
540	HTTPS リクエ ストの 証明書 ドメイ ン名検 証に不 合格と なりま した	オリジンサーバー証明書が正常に設定されているかどうかをチェックしてください。

ステータスコード	意味	推奨する対処方法
562	HTTPS リクエ ストの 接続の 確立に 失敗し ました	お問い合わせ に連絡してX-NWS-LOG-UUID情報を提供するか、または オンラインサポート で原因を調査してください。
563	HTTPS リクエ ストの 接続が タイム アウト しまし た	お問い合わせ に連絡してX-NWS-LOG-UUID情報を提供するか、または オンラインサポート で原因を調査してください。
564	HTTPS リクエ ストの back-to- originに 失敗し ました	HTTP back-to-origin方式に設定している場合は、オリジンサーバーの負荷および帯域幅の使用率をチェックするか、またはオリジンサーバーのアクセス制限をチェックしてください。プロトコル追従方式に設定している場合は、オリジンサーバー443ポートのステータスおよび証明書の設定をチェックしてください。調査の結果、オリジンサーバーに異常がない場合は、 お問い合わせ に連絡してX-NWS-LOG-UUID情報を提供するか、または オンラインサポート で原因を調査してください。

異なるノードキャッシュのコンテンツが不一致

最終更新日：：2021-05-25 16:00:53

現象の説明

CDNの同一リソースURLに対し、リージョンが異なるエンドユーザーがCDNにアクセスした場合、ノードが返す内容が一致しません。

考えられる原因

- 原因1：ドメイン名設定のキャッシュキールール（すべてのパラメータをフィルタリングする）が適用され、同時に、オリジンサーバーがパラメータに応じて異なるリソースを出力するように設定されています。
オリジンサーバーはパラメータが違くと異なるデータを出力しますが、CDNはパラメータを無視してキャッシュを実行しています。そのため、最初に受信した異なるパラメータを持つアクセスには、異なるキャッシュが作成されるため、ノードは一致しくなくなります。次回、同一のリクエストにより異なるノードにアクセスすると、受信したキャッシュが返すデータも異なります。
- 原因2：オリジンサーバーの同一リソースが更新された後にリフレッシュ処理が実行されていません。
CDNはURLに応じてリソースのキャッシュを実行します。オリジンサーバーがファイルを更新した後にURLが変化せず、内容のみに変化が発生した場合で、アクセス時にノードにキャッシュが存在する場合、直接キャッシュを取得します。同時に、各リージョンでのアクセス集中は異なるため、削除の時間も異なります。一部のノードキャッシュは削除済みで、次にアクセスする時は、**back-to-origin**サーバーが取得する新しいリソースとなります。これにより、各ノードのキャッシュには新旧バージョンが同時に存在することになり、異なるノードキャッシュの内容が一致しない状況が発生します。

解決方法

- オリジンサーバーがURLパラメータに基づき異なるリソースと、すべてのパラメータのフィルタリングを同時に使用しないというCDNドメイン名設定のキャッシュキールールを出力していることを確認してください。
- オリジンサーバーが同一URLのリソースを更新した後、一律にリフレッシュ処理することを確認してください。

処理手順

1. 自分の業務状況に応じて、オリジンサーバーがURLパラメータに基づき異なるリソースを出力するかどうか判断します。
 - 出力する場合は、[手順2](#)を実行してください。
 - そうでない場合は、[手順4](#)を実行してください。
2. [CDNコンソール](#)にログインして、【ドメイン名管理】を選択し対応するドメイン名設定を見つけ、【キャッシュ設定】>【キャッシュキールールの設定】の「フィルタリングパラメータ」項目を参照します。CDN設定のドメイン名で、パラメータキャッシュのフィルタリング機能が有効になっているかどうかを確認します。
 - 有効な場合は、[手順3](#)を実行してください。
 - そうでない場合は、[手順4](#)を実行してください。
3. キャッシュキールールの設定内にある対応するルールの操作欄で、【修正】をクリックし、ポップアップの「ルールの修正」ボックスでパラメータのフィルタリング機能をオフにしてから、【保存】をクリックします。

説明：

すべてをオフにすることがユーザーにとって不都合な場合のために、CDNには指定パラメータのフィルタリングを保持する機能が用意されています。ユーザーは実際の業務ニーズに基づき使用を選択することができます。具体的な方法について[キャッシュキールールの設定](#)をご参照ください。

4. 【ページとプリフェッチ】ディレクトリを開き、オリジンサーバーで変更されたリソースに対してリフレッシュを実行します

説明：

ユーザーはAPIを使用してリフレッシュを実行することができます。これにより、そのオリジンサーバーに変更が発生する場合、APIの呼び出しをバインドしてリフレッシュを実行し、ネットワーク全体で変更されたリソースアクセスの内容を即座に一致させることができます。詳細については[URLリフレッシュインターフェース](#)と[ディレクトリリフレッシュインターフェース](#)をご参照ください。

CDNにアクセスすると、ウェブページのアクセス速度が遅くなります

最終更新日：：2021-06-09 15:42:37

現象の説明

Tencent Cloud CDNを使用後、ウェブページのアクセス速度が非常に遅いままです。

考えられる原因

- 原因1：接続されているドメイン名のCDNアクセラレーションサービスが有効化されていません。これはドメイン名のDNSプロバイダでCNAMEレコードを設定していないためである可能性があります。[ドメイン名解決の確認](#)を実行してください。
- 原因2：ノードキャッシュの有効期限設定が間違っています。[ノードキャッシュの有効期限設定をチェック](#)を実行してください。
- 原因3：リソースへの初めてのアクセスで、そのリソースに対してプリフェッチ処理を実行していません。[URLのプリフェッチを実行](#)してください。
- 原因4：ウェブページのアーキテクチャ自体に問題があります。[ウェブページのアーキテクチャの最適化](#)を実行してください。

解決方法

ドメイン名解決をチェック

以下の内容は、nslookupコマンドを使用してCDNアクセラレーションドメイン名のDNS解決をクエリーする例です。

```
nslookup アクセラレーションドメイン名
```

```
C:\Users\>nslookup .com
Address:
.com
Address:
Aliases:
.com
.com cdn.dnsv1.com
```

クエリーによるドメイン名解決で、上図の赤いボックスのサフィックスdnsv1.comを持つCNAME解決レコードがない場合は、接続したドメイン名のCDNアクセラレーションサービスが有効になっていないことを意味します。これはドメイン名DNSプロバイダでCNAMEレコードを設定していないためである可能性があります。[CNAMEの設定](#)ドキュメントに従い、ドメイン名のDNSプロバイダに移動してCNAMEレコードを設定することができます。

ノードキャッシュ有効期限設定をチェック

[CDNコンソール](#)にログインし、左側のメニューバーで【ドメイン名管理】を選択し、ドメイン名操作列の【管理】をクリックして、ドメイン名設定画面に入ります。Tabを【キャッシュ設定】に切り替えると、【ノードキャッシュ期限切れ設定】が表示されます。

The screenshot shows the Tencent Cloud CDN console interface. On the left, the 'Domain Management' menu item is highlighted. The main content area shows the 'Cache Configuration' tab selected, with sub-tabs for 'Basic Configuration', 'Access Control', 'Cache Configuration', 'Origin-pull Configuration', 'HTTPS Configuration', and 'Advanced Configuration'. The 'Cache Configuration' sub-tab is active, displaying 'Cache Key Rule Configuration' and 'Node Cache Validity Configuration' sections. The 'Node Cache Validity Configuration' section is highlighted with a red box.

Type	Content	Ignore Query String	Ignore URL Case
All Files	All Files	Not Filter	No

- アクセスしたリソースに対応するノードキャッシュルールをチェックし、ノードキャッシュの有効期限を0にした設定があるか、ノードキャッシュの有効期限が短すぎるか、またはキャッシュがないという状況が存在するかどうか確認します。

CDNノードにキャッシュがない場合は、アクセスリクエストはback-to-originとなり、アクセラレーションの効果を得られなくなります。ユーザーの必要に応じてノードのキャッシュ時間を設定することをお勧めします。

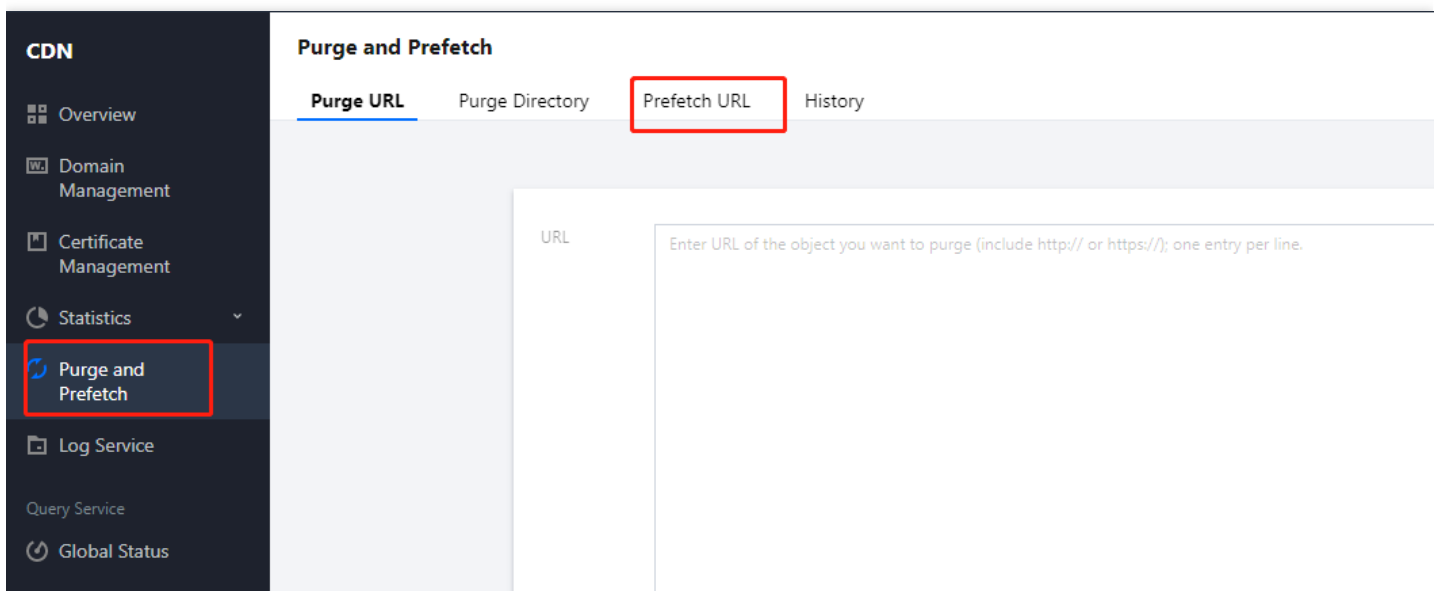
- オリジンサーバーのキャッシュヘッダーCache-Controlがno-store/no-cache/privateに設定されているかどうか確認します。

- オリジンサーバーのキャッシュヘッダーCache-Controlがno-store/no-cache/privateに設定されている場合、同時に「強制キャッシュ」を有効にしないと、CDNノードは設定したキャッシュ時間に基づくリソースのキャッシュを行いません。
- 「強制キャッシュ」を有効にしておらずオリジンサーバーのCache-Controlフィールドがno-cache/no-store/privateの場合は、キャッシュ時間を設定しても、CDNノードもリソースをキャッシュしません。

その他の設定ルールについては、[ノードキャッシュ期限切れ設定](#)に移動してください。

URLプリフェッチの実行

リソースへのアクセスが初めてで、そのリソースがプリフェッチ処理をされていない場合、CDNノードはback-to-originしてリソースをプルするため、初めてのアクセス速度が遅くても正常です。[CDNコンソール](#)にログインして、【ページとプリフェッチ】内でURLプリフェッチ機能を探し、[URLプリフェッチ](#)を実行することを推奨します。



ウェブページアーキテクチャの最適化

ウェブページには動的リソースが比較的多く、アクセスするたびにback-to-originして最新のリソースをプルするため、アクセス速度に影響します。ウェブページの動的リソースの占める割合が多い場合は、オリジンサーバーを最適化して、動的リソースと静的リソースを分離し、CDN配信を使用して静的リソースの速度を上げることを推奨します。

トラフィックのヒット率が低い場合

最終更新日：2021-08-11 14:30:29

現象の説明

リアルタイムモニタリングでのトラフィックヒット率の数値が低く、想定を満たさない。

考えられる原因

- キャッシュ更新が行われた
キャッシュ更新により、指定されたコンテンツがノード上でクリアされ、一時的にトラフィックヒット率が低下する状況が発生します。
- オリジンサーバーに新しいリソースがある
オリジンサーバーに新しいリソースが多くある場合、CDNノードでback-to-originが発生し、トラフィックヒット率が低下します。
- オリジンサーバーの異常
オリジンサーバーに不具合が起こり、5XXまたは4XXエラーが多くなった場合、トラフィックヒット率に影響します。
- キャッシュポリシーの設定が適切でない
実際の業務状況に合わせてキャッシュルールを設定してください。
- Back-to-Origin of Rangeがオフになっている
Back-to-Origin of Rangeがオフの場合、リクエストに応じて分割して引き出すのではなく、back-to-origin時に大容量ファイルをすべて引き出します。これによりback-to-originトラフィックが増加し、トラフィックヒット率に影響します。
- ドメイン名設定のキャッシュキールール（すべてのパラメータを無視）が適用されたが、オリジンサーバーのリソースがパラメータの違いによって異なる
オリジンサーバーのリソースがパラメータの違いによって異なる一方で、CDNはすべてのパラメータを無視してキャッシュを行うため、異なるパラメータのリソースがリクエストされた場合、対応するリソースにマッチせず、トラフィックヒット率に影響します。

解決方法

1. オリジンサーバーに異常がないかどうか確認します。

2. キャッシュ更新が行われたり、オリジンサーバーに新しいリソースが多かったりする場合、これは正常な現象です。
3. オリジンサーバーが、URLパラメータに基づき異なるリソースを出力するのと、CDNドメイン名設定のキャッシュキールール（すべてのパラメータを無視）が同時に使用されていないことを確認します。
4. 実際の業務状況に合わせてキャッシュルールを設定します。

処理手順

1. オリジンサーバーに異常がないか、またはキャッシュ更新が行われていないか確認します。
 - YESの場合、ヒット率の低下は正常な現象です。
 - NOの場合は、[手順2](#)を実行してください。
2. 自分の業務状況に応じて、オリジンサーバーがURLパラメータに基づき異なるリソースを出力するかどうか判断します。
 - YESの場合は、[手順3](#)を実行してください。
 - NOの場合は、[手順5](#)を実行してください。
3. [CDNコンソール](#)にログインし、【ドメイン名管理】を選択して対応するドメイン名設定を見つけ、【キャッシュ設定】>【キャッシュキールールの設定】の「パラメータを無視」の項目を確認します。CDN設定のドメイン名で、パラメータキャッシュを無視する機能が有効になっているかどうか確認します。
 - YESの場合は、[手順4](#)を実行してください。
 - NOの場合は、[手順5](#)を実行してください。
4. キャッシュキールール設定内の対応するルールの操作欄で、【変更】をクリックし、ポップアップ表示された「ルールの修正」ボックスでパラメータの無視機能をオフにしてから、【保存】をクリックします。

説明：

すべてをオフにすることがユーザーにとって不都合な場合のために、CDNには指定パラメータの無視を保持する機能が用意されています。ユーザーは実際の業務ニーズに基づき使用を選択することができます。具体的な方法については[キャッシュキールールの設定](#)をご参照ください。

5. [CDNコンソール](#)にログインし、【ドメイン名管理】を選択して対応するドメイン名設定を見つけ、【キャッ

シュ設定】>【ノードキャッシュ期限切れ設定】を確認します。キャッシュルールが自身の業務やオリジンサーバーの実際の状況に適合しているか確認してください。

- YESの場合は、[手順5](#)を実行してください。
- NOの場合は、[ノードキャッシュ期限切れ設定](#)を参照し、キャッシュルールを調整してください。

CDNドメイン名に突然、404エラーが発生しました

最終更新日：：2021-11-08 12:18:54

##故障について

CDNドメイン名に突然、404エラーが発生しました。

考えられる原因

1. オリジンサーバーの異常。
2. コンソールのオリジンサーバー情報とback-to-origin HOSTの設定に変更があった。

ソリューション

1. オリジンサーバーに異常がないかどうか確認します。
2. コンソールのオリジンサーバー情報とback-to-origin HOST設定をチェックして、関連設定に異常がないことを確認します。

処理手順

1. オリジンサーバーに異常があるかどうかチェックします。
 - YESの場合は、オリジンサーバーを修復します。
 - NOの場合は、[手順2](#)を実行してください。
2. コンソールのオリジンサーバー情報とback-to-origin HOSTの設定をチェックします。

[CDNコンソール](#)にログインし、[ドメイン名管理](#)を選択して対応するドメイン名設定を見つけ、[基本設定](#)>[オリジンサーバー情報](#)の「オリジンサーバーアドレス」、「back-to-origin HOST」を表示して、ここの設定が正しいことを確認します。

 - オリジンサーバータイプ：

ユーザー 保有 オリジン サーバー	ユーザー保有オリジンサーバーを選択する場合は、正常にアクセスできるビジネスサーバーのIPアドレスまたはドメイン名を提供する必要があります
COS オリジン サーバー	Tencent Cloud COSのうち1つのバケットをオリジンサーバーとして選択する場合は、バケットの設定に応じてデフォルトのドメイン名または静的ウェブサイトのシナリオを選択します。バケットがプライベートバケットの場合は、CDNを承認してback-to-origin認証を有効にし、プライベートバケットアクセスを有効にしてください
サード パーティ COS	サードパーティCOSを選択する場合は、オリジンサーバーとして有効なバケットアクセスアドレスを入力してください。現在サポートされているサードパーティは、AWS S3とAlibaba Cloud OSSです。サードパーティのプライベートバケットにback-to-originするには、有効なキーを入力し、back-to-origin認証を有効にして、プライベートバケットアクセスを有効にする必要があります

◦ back-to-origin HOST :

back-to-originドメイン名です。CDNノードがback-to-originのときは、アクセスしたオリジンサーバーのIPアドレスの下にある具体的なサイトドメイン名です。デフォルトは、現在のアクセラレーションドメイン名です。汎用ドメイン名にアクセスする場合、デフォルトは汎用ドメイン名であり、実際のback-to-origin HOSTはアクセスドメイン名です。実際の業務状況に応じて変更できます（注：オリジンサーバータイプがCOSオリジンサーバーやサードパーティCOSの場合、変更できません）。

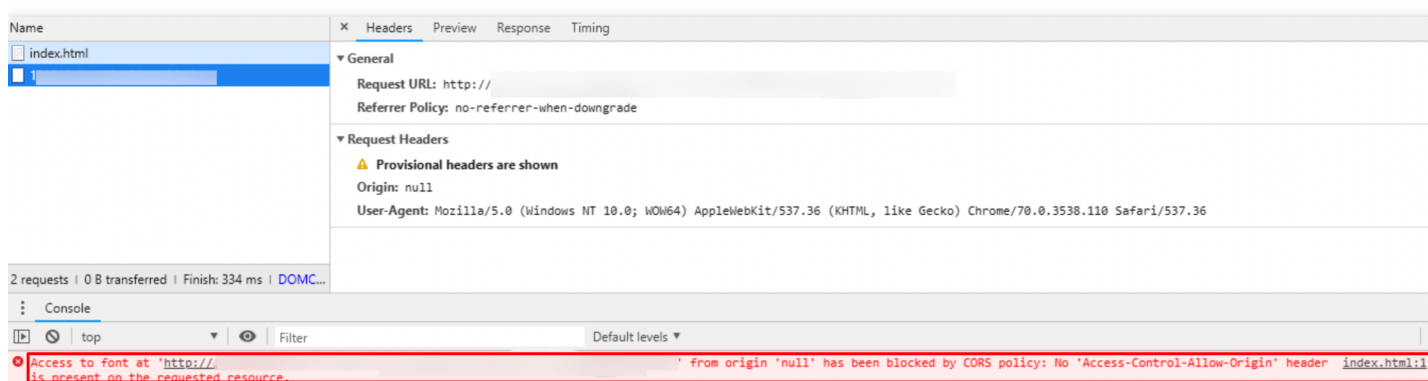
オリジンサーバー設定の詳細については、[オリジンサーバー設定](#)をご参照ください。

ページ表示の異常-アクセスクロスドメインエラー

最終更新日：：2021-10-08 10:24:11

現象の説明

下図に示すように、フロントエンドはページエラーや表示異常などの問題を引き起こすクロスドメインエラーを報告します。

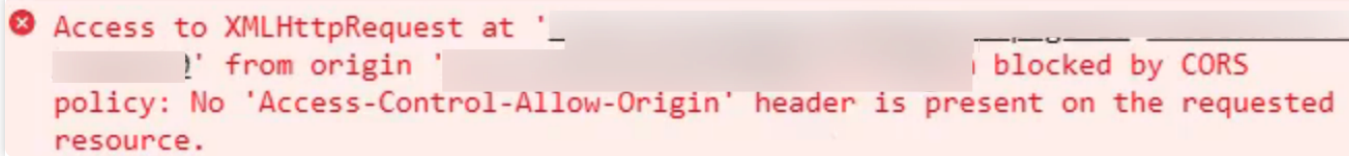


考えられる原因

クロスドメインでは、ブラウザの同一オリジンポリシーの制限により、またウェブページのセキュリティを考慮する観点から、スクリプトを介してリクエストが各種ソースに送信されると、このリクエストの応答がブラウザによってブロックされるため、フロントエンドがエラーまたはページが正常に表示されない事象を報告します。リクエストURLの**プロトコル**、**ドメイン名**、**ポート**のいずれかが現在のページのURLと異なる場合はクロスドメインです。

解決方法

1. 次に示すように、ページ異常エラーがクロスドメインが原因であるかどうかを確認します。



× Access to XMLHttpRequest at '...' from origin '...' blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

2. CDNコンソールで対応するHTTPレスポンスヘッダーを設定し、このリソースへのアクセスを許可するドメインを定義します。

処理手順

1. [CDNコンソール](#)にログインし、対応するドメイン名管理-高度な設定-HTTPレスポンスヘッダーの設定で **Access-Control-Allow-Origin**ヘッダーパラメータを設定します。下図に示すように、すべてのドメイン名によって開始されたクロスドメインリクエストが許可されます。詳細については、[Access-Control-Allow-Originマッチングモードの説明](#)をご参照ください。
2. または次に示すように、クロスドメインリクエストの開始を許可する単一または複数の既知のドメイン名/IPに的を絞って設定します。
また業務上の必要性に応じて、**Access-Control-Request-Method**、**Access-Control-Request-Headers**、**Access-Control-Max-Age**などのヘッダーパラメータを追加することで、ブラウザが受信可能なリクエストの方法、追加できるヘッダー、事前リクエストの有効時間などを制限することもできます。詳細については、[パラメータサポートリスト](#)をご参照ください。

注意：

COS側のバケットでクロスドメインアクセスを設定済みである場合は、正常なアクセスを保証するため、CDNコンソールの[HTTPレスポンスヘッダー](#)でクロスドメインルールを同期的に設定してください。

パラメータサポートリスト

ヘッダーパラメータ	説明
-----------	----

ヘッダーパラメータ	説明
Access-Control-Allow-Origin	リソースのクロスドメインの権限問題を解決するために使用します。ドメイン値はそのリソースドメインへのアクセスを定義しています。ソースリクエストHostがドメイン名の設定リストにある場合は、対応する値を戻りヘッダーに直接入力します。ワイルドカード <code>*</code> を設定して、すべてのドメインによるリクエストを許可することもできます。詳細は Access-Control-Allow-Originマッチングモードの説明 をご参照ください。「 <code>*</code> 」の入力、または複数のドメイン名/IP/ドメイン名とIPの組み合わせ入力（「 <code>http://</code> 」または「 <code>https://</code> 」を必ず含める必要があります。入力見本：「 <code>http://test.com,http://1.1.1.1</code> 」、カンマ区切り）（注意：最大1000文字まで入力可能）をサポートしています。
Access-Control-Allow-Methods	クロスドメインが許可するHTTPリクエスト方法の設定に使用します。同時に複数の方法を設定することができます。例：Access-Control-Allow-Methods: <code>POST, GET, OPTIONS</code> 。
Access-Control-Max-Age	プレリクエストする有効時間の指定に使用します。単位は秒。非シンプルクロスドメインリクエストは、正式に通信する前に、HTTP照会リクエストを一度追加する必要があります。これを「プレリクエスト」と呼びます。このクロスドメインリクエストが安全に受信可能かを確認するために使用します。以下のリクエストが非シンプルクロスドメインリクエストと見なされた場合：GET、HEADまたはPOST以外の方式で起動するか、またはPOSTを使用しますが、リクエストデータのタイプはapplication/x-www-form-urlencoded、multipart/form-data、text/plain以外のデータタイプになります。例えば、application/xml またはtext/xmlになります。カスタマイズしたリクエストヘッドを使用して：Access-Control-Max-Age: <code>1728000</code> にすると、1728000秒（20日）以内に、そのリソースのクロスドメインアクセスに対してその他のプレリクエストを再発信しなくなることを示します。
Access-Control-Expose-Headers	どのヘッダーが応答する一部としてクライアントに公開できるかを指定するために使用します。デフォルトでは、Cache-Control、Content-Language、Content-Type、Expires、Last-Modified、Pragmaの6種類のヘッダーのみクライアントに公開できます。クライアントをその他のヘッダー情報にアクセスさせたい場合は、以下の設定を行うことができます。複数のヘッダーを入力する場合は、「 <code>,</code> 」を用いて区切ります。例：Access-Control-Expose-Headers: Content-Length,X-My-Headerは、クライアントがContent-LengthおよびX-My-Headerの2つのヘッダー情報にアクセスできることを示しています。

Access-Control-Allow-Origin一致パターンの概要

一致パターン	フィールド値	説明
完全一致	<code>*</code>	<code>*</code> を設定する時、応答してヘッダーを追加します：Access-Control-Allow-Origin: <code>*</code>

一致パターン	フィールド値	説明
固定一致	<pre>http://cloud.tencent.com https://cloud.tencent.com http://www.b.com</pre>	ソース <code>https://cloud.tencent.com</code> がリストにヒットした場合、レスポンスしてヘッダー追加します: <code>Access-Control-Allow-Origin: https://cloud.tencent.com</code> ですが <code>https://www.qq.com</code> でリストにヒットしない場合、レスポンスに変化はありません。
セカンダリ汎用ドメイン名に一致	<pre>https://*.tencent.com</pre>	ソース <code>https://cloud.tencent.com</code> がリストにヒットした場合、応答してヘッダーを追加します: <code>Access-Control-Allow-Origin: https://cloud.tencent.com</code> ソースが <code>https://cloud.qq.com</code> でリストにヒットしない場合、レスポンスに変化はありません。
ポートに一致	<pre>https://cloud.tencent.com:8080</pre>	ソースが <code>https://cloud.tencent.com:8080</code> でリストにヒットした場合、応答してヘッダーを追加します: <code>Access-Control-Allow-Origin: https://cloud.tencent.com:8080</code> ソースが <code>https://cloud.tencent.com</code> でリストにヒットしない場合、レスポンスに変化はありません。

注意：

特別なポートがある場合は、リストに関連情報を入力する必要があります。任意のポートへの対応はサポートしないため、指定する必要があります。

リソースキャッシュは有効化されていません

最終更新日：2021-11-08 12:17:45

##故障について

ノードキャッシュの有効期限を設定し、ウォームアップが完了した後も、リクエストがノードキャッシュにヒットしません。

考えられる原因

1. 複数のキャッシュを設定しているが、優先度が明確ではない。
2. オリジンサーバーのキャッシュポリシーに従うように設定されているが、オリジンサーバーのCache-Controlフィールドが `no-cache/no-store/private` になっている。

ソリューション

1. キャッシュの優先度を正しく設定します
CDNキャッシュルールは複数設定でき、最下部が優先されます。ここでは、お客様が想定しているルールが有効になるように、ユーザーの想定と優先度が一致していることを確認する必要があります。
2. キャッシュ時間を正しく設定します
コンソールのキャッシュ時間が短すぎないかチェックします。

注意：

ファイルアクセス頻度が低く、話題性に欠け、ユーザーが頻繁にアクセスしないURLは、すべてのキャッシュルールを満たしていても、ノードによってキャッシュから削除されるリスクがあります。

3. キャッシュ設定ルールが想定に合致しているかチェックします。
 - CDNキャッシュルールがパラメータキャッシュルールに設定されて、ノードがキャッシュされないようにしているかチェックします。
 - CDNノードの期限切れのキャッシュ設定が強制的にキャッシュされないかチェックします。
 - CDNノードの期限切れキャッシュ設定がオリジンサーバーに従っている場合、**back-to-origin**時のオリジンサーバーのヘッダーが `no-cache/no-store/private` を返すかどうかチェックします。

処理手順

1. キャッシュ設定の優先度（最下部が優先）をチェックします

[CDNコンソール](#)にログインし、左側メニューバーで**ドメイン名管理**を選択し、ドメイン名操作列の**管理**をクリックして、ドメイン名の設定ページに進み、Tabを**キャッシュ設定**に切り替えると、**キャッシュキールール設定**が見つかります。下図に示すように、**jpg**の無視されないパラメータキャッシュの優先度は、パラメータキャッシュを無視するすべてのファイルの優先度よりも高くなります。業務のキャッシュポリシーが優先度設定に適合していることを確認する必要があります。

2. キャッシュ時間をチェックします

[CDNコンソール](#)にログインし、左側メニューバーで**ドメイン名管理**を選択し、ドメイン名操作列の**管理**をクリックして、ドメイン名の設定ページに進み、Tabを**キャッシュ設定**に切り替えると、**ノードキャッシュの有効期限の設定**が見つかります。下図に示すように、キャッシュ設定時間が短すぎると、ビジネスキャッシュポリシーへの適合を確保するため、キャッシュの設定が有効になっていないと誤解する可能性があります。

3. キャッシュポリシーをチェックします

キャッシュキールール設定とノードキャッシュ有効期限設定のポリシーをチェックして、想定と合致していることを確認します。

オリジンサーバーに従うように設定されている場合は、オリジンサーバーの**Cache-Control**フィールドが `no-cache/no-store/private` でないことを確認してください。

4. キャッシュする必要のあるリソースを再度プリフェッチし、プリフェッチが完了するのを待ってから、再度リクエストすれば完了です。