Cross-origin resource sharing (CORS)

The same-origin policy (https://www.w3.org/Security/wiki/Same_Origin_Policy) is a security policy enforced on client-side web applications (like web browsers) to prevent interactions between resources from different origins. While useful for preventing malicious behavior, this security measure also prevents legitimate interactions between known origins. For example, a script on a page hosted on App Engine at `example.appspot.com` might need to use resources stored in a Cloud Storage bucket at `example.storage.googleapis.com`. However, because these are two different origins from the perspective of the browser, the browser won't allow a script from `example.appspot.com` to fetch resources from `example.storage.googleapis.com`.

The Cross Origin Resource Sharing (https://www.w3.org/TR/cors/) (CORS) spec was developed by the World Wide Web Consortium (W3C) (https://www.w3.org/) to get around this limitation. Cloud Storage supports this specification by allowing you to configure your buckets to support CORS. Continuing the above example, you can configure the `example.storage.googleapis.com` bucket so that a browser can share its resources with scripts from `example.appspot.com`.

How CORS works

There are two types of CORS requests, simple and preflighted. A simple request can be initiated directly. A preflighted request must send a preliminary, "preflight" request to the server to get permission before the primary request can proceed. A request is preflighted if any of the following circumstances are true:

- It uses methods other than `GET`, `HEAD` or `POST`.
- It uses the `POST` method with a `Content-Type` other than `text/plain`, `application/x-www-form-urlencoded`, or `multipart/form-data`.
- It sets custom headers.

The following process occurs when a browser makes a simple request to Cloud Storage:

1. The browser adds the `Origin` header to the request. The `Origin` header contains the origin of the resource seeking to share the other domain's resources, for example, `Origin:http://www.example.appspot.com`.

2. Cloud Storage compares the HTTP method of the request and the value of the `Origin` header to the **Methods** and **Origins** information in the target bucket's CORS configuration to see if there are matches. If there are, Cloud Storage includes the `Access-Control-Allow-Origin` header in its response. The `Access-Control-Allow-Origin` header contains the value of the `Origin` header from the initial request.
3. The browser receives the response and checks to see if the `Access-Control-Allow-Origin` value matches the domain specified in the original request. If they do match, the request succeeds. If they don't match, or if the `Access-Control-Allow-Origin` header is not present in the response, the request fails.

A preflighted request performs the following steps first. If it is successful, it then follows the same process as a simple request:

1. The browser sends an `OPTIONS` request containing the `Requested Method` and `Requested Headers` of the primary request.
2. The server responds back with the values of the HTTP methods and headers allowed by the targeted resource. If any of the method or header values in the preflight request aren't in the set of methods and headers allowed by the targeted resource, the request fails, and the primary request isn't sent.

This is a very simplified description of CORS. For a more complete description, read the Cross Origin Resource Sharing (https://www.w3.org/TR/cors/) spec.

Cloud Storage CORS support

Note: CORS configuration only affects requests to XML API endpoints (https://cloud.google.com/storage/docs/request-endpoints). JSON API endpoints allow CORS requests, regardless of CORS settings on the target bucket. Requests to the authenticated browser download endpoint `storage.cloud.google.com` do not allow CORS requests.

Cloud Storage allows you to set CORS configuration at the bucket level only. You can set the CORS configuration for a bucket using the gsutil (https://cloud.google.com/storage/docs/gsutil) command-line tool, the XML API (https://cloud.google.com/storage/docs/xml-api/overview), or the JSON API (https://cloud.google.com/storage/docs/json_api/).

For more information about setting CORS configuration on a bucket, see Configuring Cross-Origin Resource Sharing (CORS) (https://cloud.google.com/storage/docs/configuring-cors). For more information about CORS configuration elements, see Set Bucket CORS (https://cloud.google.com/storage/docs/xml-api/put-bucket-cors).

You can use either of the following XML API request URLs to obtain a response from Cloud Storage that contains the CORS headers:

```
storage.googleapis.com/[BUCKET_NAME]
```

```
[BUCKET_NAME].storage.googleapis.com
```

For information about XML API request URLs, see Request Endpoints (https://cloud.google.com/storage/docs/request-endpoints).

Client-side CORS support

Most browsers use the `XMLHttpRequest` object to make a cross-domain request. `XMLHttpRequest` takes care of all the work of inserting the right headers and handling the CORS interaction with the server. You don't have to add any new code to take advantage of CORS support on Cloud Storage buckets.

Last updated April 18, 2019.