Anypoint Platform and CORS

Issues with web clients and API testing

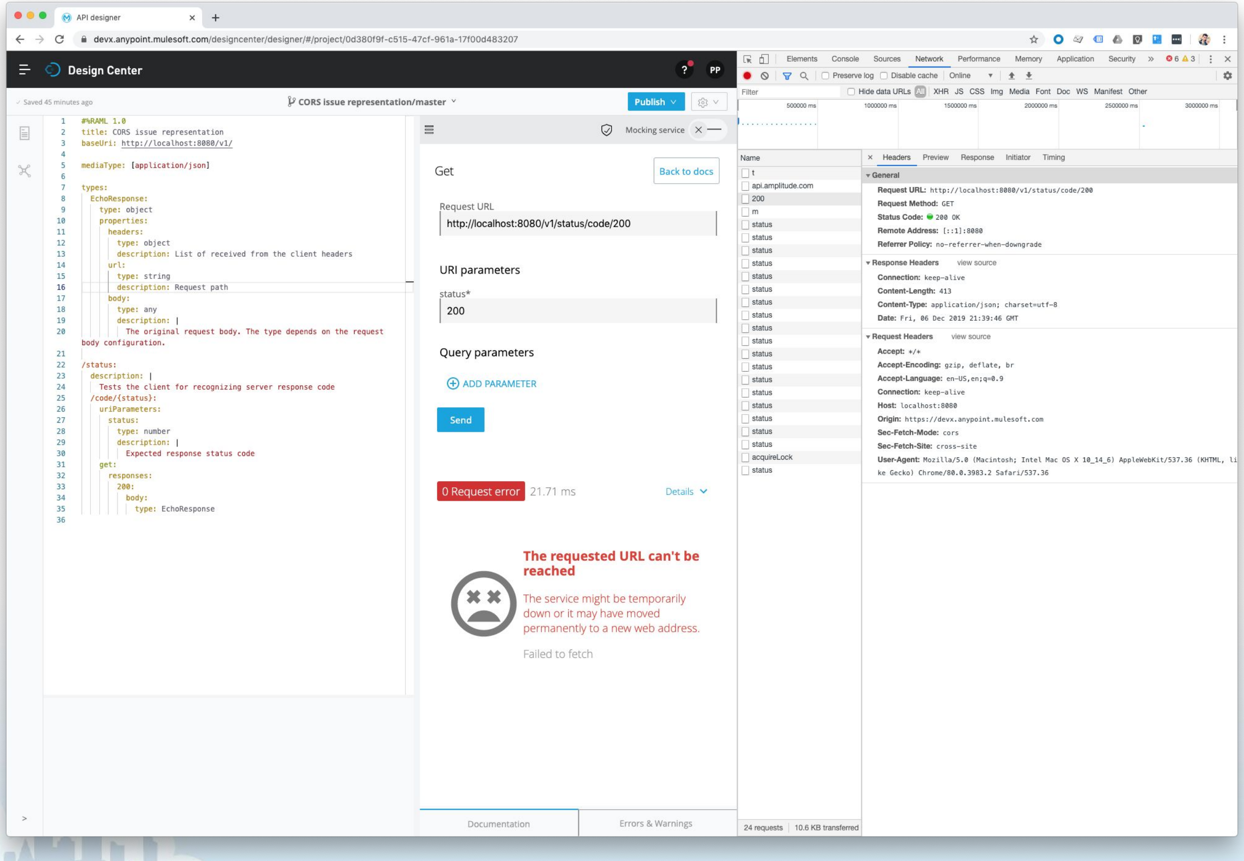
Pawel Psztyc

OCTO, Dec 2019

Problem statement

- Anypoint platform is based on a web platform
- Web platform has an idea of **C**ross **O**rigin **R**esource **S**haring
    - Resources from other origins requires special headers in the response in order to be read by the web client
    - Enterprise customers often has no CORS headers in API response
- This prohibits efficient use of some of Anypoint tooling related to API consumption, testing, and monitoring
    - API Console is unable to make a request to a foreign origin (API Consumer)
    - Testing can only be on a server side, tooling can only work when embedded in the platform or having custom integration
    - API Monitoring tools can only work with a server component

CORS issue representation



```
#%RAML 1.0
title: CORS issue representation
baseUri: http://localhost:8080/v1/

mediaType: [application/json]

types:
  EchoResponse:
    type: object
    properties:
      headers:
        type: object
        description: List of received from the client headers
      url:
        type: string
        description: Request path
      body:
        type: any
        description: |
          The original request body. The type depends on the request
body configuration.

/status:
  description: |
    Tests the client for recognizing server response code
  /code/{status}:
    uriParameters:
      status:
        type: number
        description: |
          Expected response status code
    get:
      responses:
        200:
          body:
            type: EchoResponse
```

Get

Request URL: http://localhost:8080/v1/status/code/200

URI parameters

status*: 200

Query parameters

0 Request error 21.71 ms

The requested URL can't be reached

The service might be temporarily down or it may have moved permanently to a new web address.

Failed to fetch

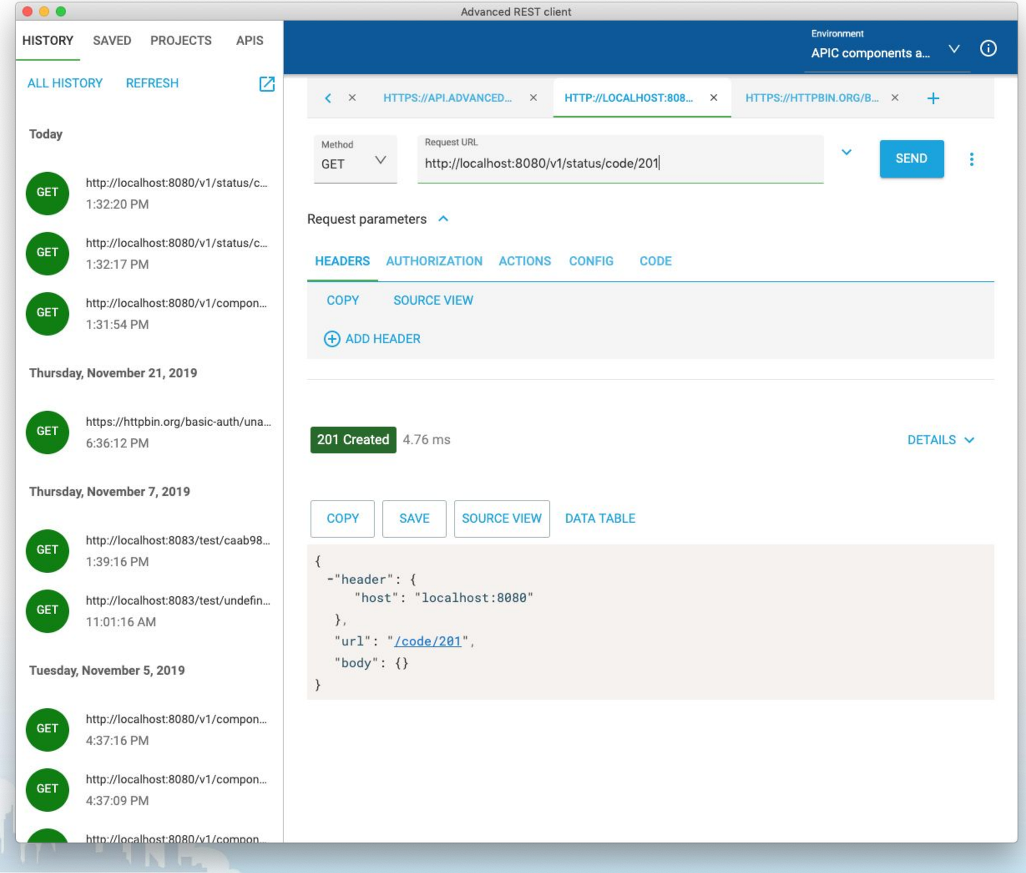
```
▾ General
Request URL: http://localhost:8080/v1/status/code/200
Request Method: GET
Status Code: ● 200 OK
Remote Address: [::1]:8080
Referrer Policy: no-referrer-when-downgrade
▾ Response Headers
Connection: keep-alive
Content-Length: 413
Content-Type: application/json; charset=utf-8
Date: Fri, 06 Dec 2019 21:39:46 GMT
▾ Request Headers
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: keep-alive
Host: localhost:8080
Origin: https://devx.anypoint.mulesoft.com
Sec-Fetch-Mode: cors
Sec-Fetch-Site: cross-site
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, li
ke Gecko) Chrome/80.0.3983.2 Safari/537.36
```

API Designer is unable to make a request to the API's endpoint due to CORS limitation.

Even though it has a proxy service it won't be possible to access an API behind a firewall.

Only way to overcome this limitation is to ask the client to install additional software inside their network ⚠️

CORS issue representation



Today

GET http://localhost:8080/v1/status/c... 1:32:20 PM

GET http://localhost:8080/v1/status/c... 1:32:17 PM

GET http://localhost:8080/v1/compon... 1:31:54 PM

Thursday, November 21, 2019

GET https://httpbin.org/basic-auth/una... 6:36:12 PM

Thursday, November 7, 2019

GET http://localhost:8083/test/caab98... 1:39:16 PM

GET http://localhost:8083/test/undefin... 11:01:16 AM

Tuesday, November 5, 2019

GET http://localhost:8080/v1/compon... 4:37:16 PM

GET http://localhost:8080/v1/compon... 4:37:09 PM

GET http://localhost:8080/v1/compon...

Method: GET

Request URL: http://localhost:8080/v1/status/code/201

201 Created 4.76 ms

```
{
 -"header": {
     "host": "localhost:8080"
  },
  "url": "/code/201",
  "body": {}
}
```

Other clients does not have this limitation.

This includes desktop clients, CLI tools, browser extensions*, mobile applications**.

*Browser extensions require permission from the user to access foreign origin

**Mobile applications require permission from the user to access internet

Inconsistency across platforms

- Web applications are the only affected by the limitation
  - CORS was introduced to protect user data by prohibiting unauthorized access to a resource in a session based APIs so the existence of CORS is not in question here
- Other platforms has no such limitation
  - Again, on other platforms the security model disallow an evil script to be executed in a context that would allow to access user data in a remote machine
- There's no way for a web application to overcome this limitation
  - Except for setting up CORS headers in a server response. This, however, sometimes is beyond the control of a web developer.

CORS permission API

The solution is to introduce new Permission API for web developers to request from the user access to a remote resource, effectively disabling CORS rules for given URL pattern.

Security consideration

- When permission is granted any XHR/Fetch call must not include “cookie” header in any request. This provides a compatibility layer with current state and prohibits new security issues
- API calls must be made with the “authorization” header (if authentication is even required) with a token (JWT).
- API providers have an option to opt-out from Permission based requests by setting “Access-Control-Allow-Origin” header which takes precedence over the permission API
    - Although it would be inconsistent with other platforms which ignores this header

Code example

```
const result = await navigator.permissions.query({
  name: 'https://api.domain.com/v1/*'
});

if (result.state == 'granted') {
  runApiRequest();
} else if (result.state == 'prompt') {
  requestPermission();
} else if (result.state == 'denied') {
  renderInstallProxyInfo();
}
```