CORS Policy Usage Scenarios

SOA | software™

Copyright

Trademarks

SOA Software, Inc.

SOA Software, Inc.
12100 Wilshire Blvd, Suite 1800
Los Angeles, CA 90025
(866) SOA-9876
www.soa.com
info@soa.com

Disclaimer

Contents

CORS Policy Usage Scenarios

This document provides a list of common CORS Policy usage scenarios.

Scenario 1: Send Preflight and CORS Response Headers

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS -** REST Borrower).

3 Virtualize **PS** on Network Director (Example service name: **VS1**).

4 Create CORS Policy with the following configuration:

- Max Age: 0
- Allow Credentials: true
- Allow Origins: http://client.cors-api.appspot.com
- Allow Headers:
- Expose Headers: x-response-for-cors-pass2

5 Attach the policy to **VS1**.

6 Navigate to client: http://client.cors-api.appspot.com/client .

7 Provide URL of the VS1 resource for DELETE method.

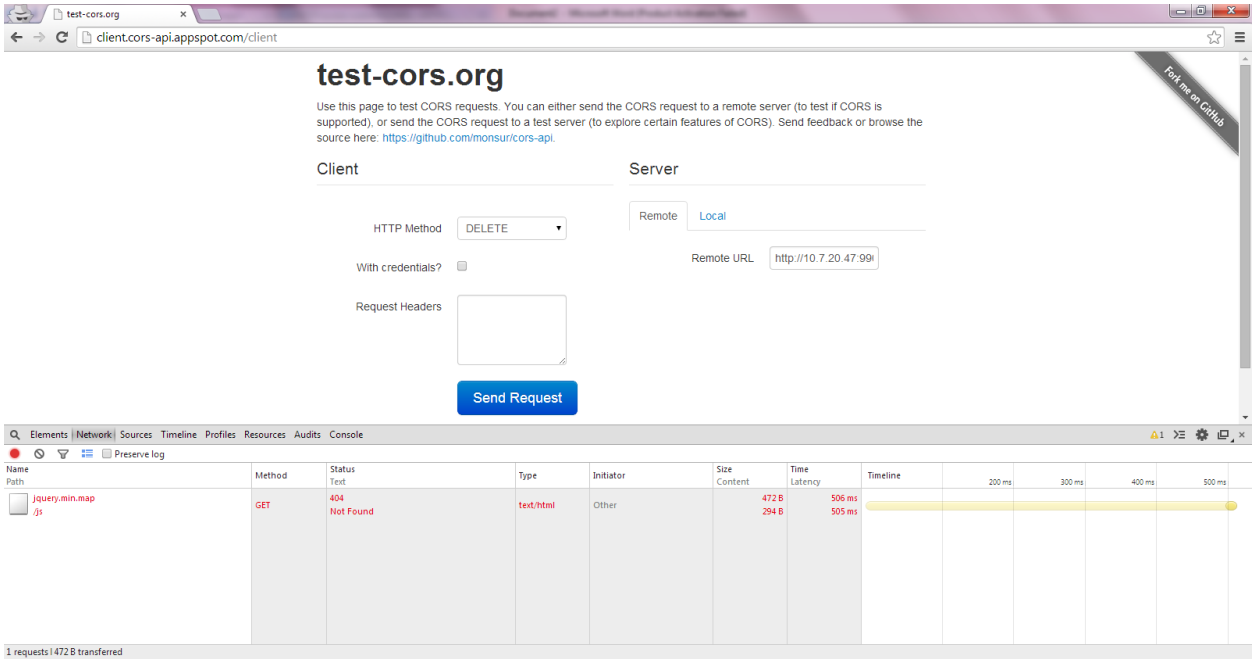
Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

8 Select Delete Method.

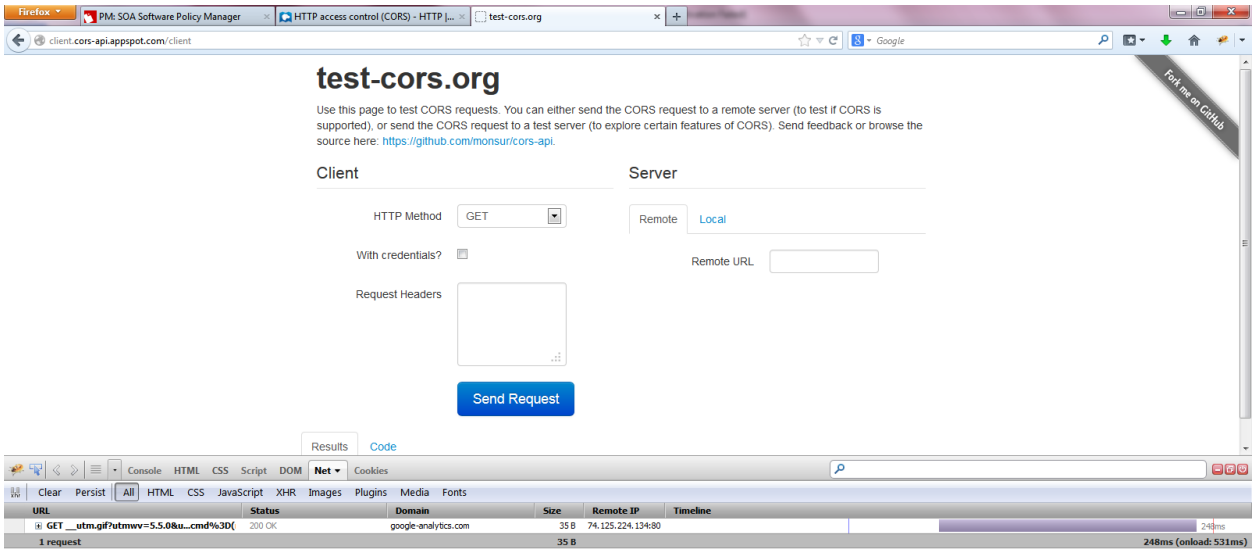
9 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

10 Navigate to Network tab.

11 The client over Chrome:



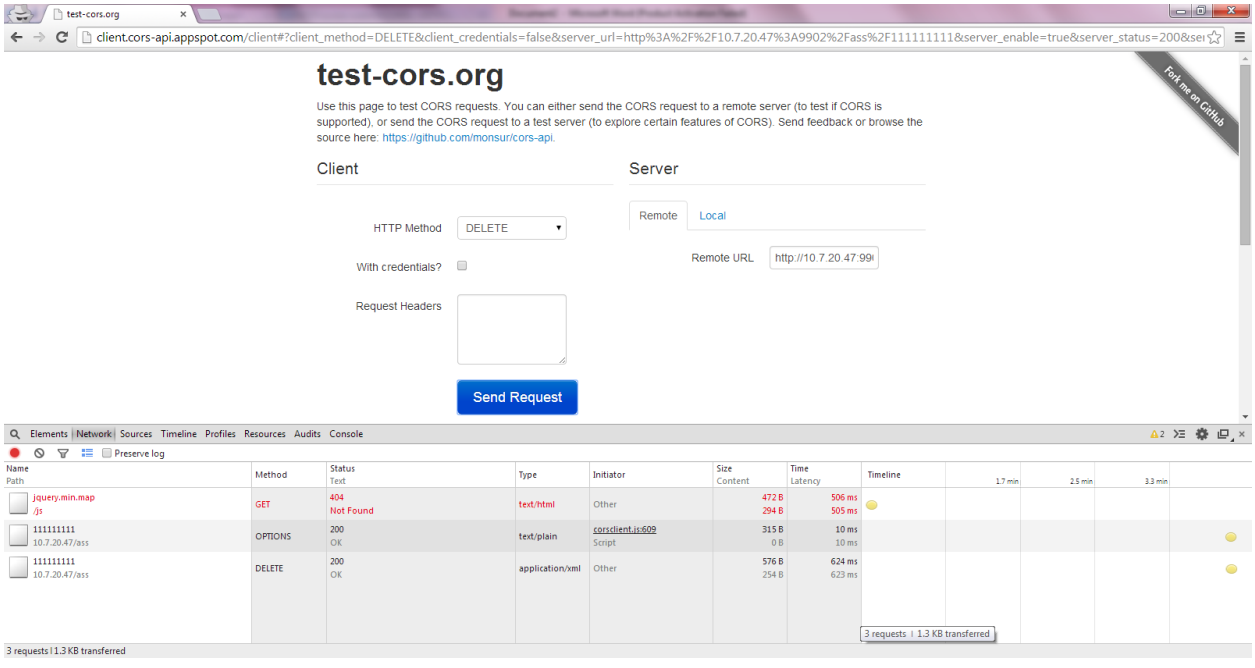
12 The client over Firefox:



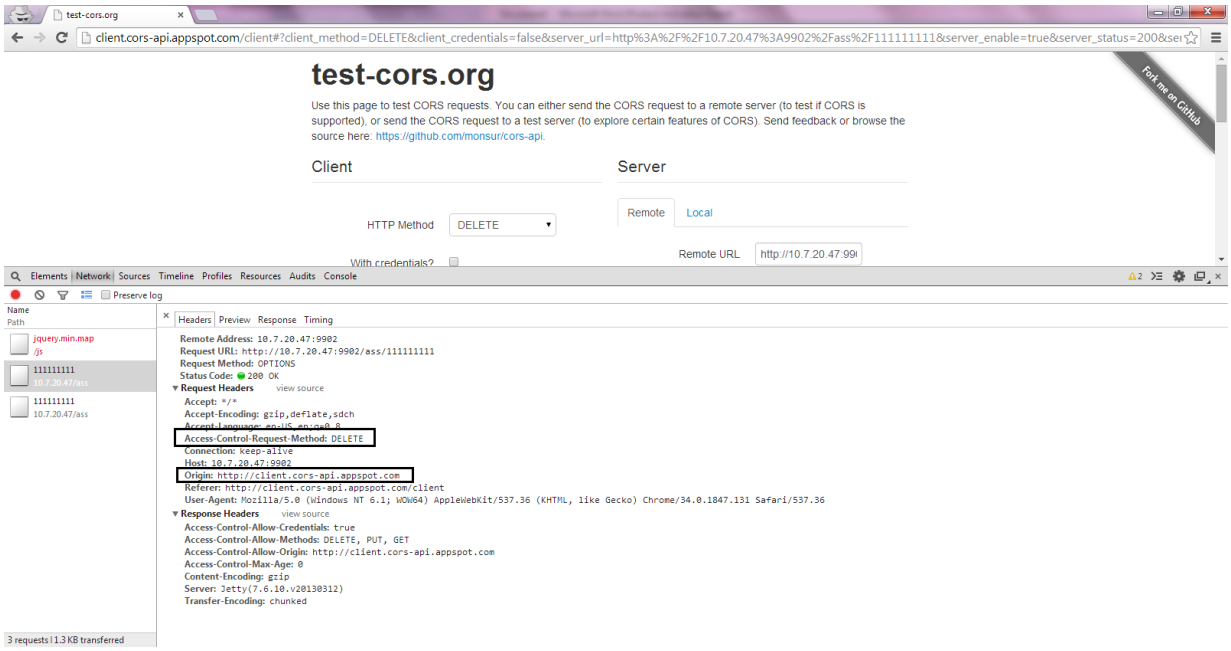
13 Send request to DELETE method.

14 Initially a preflight message is sent (OPTION method is used to send preflight request).

15 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

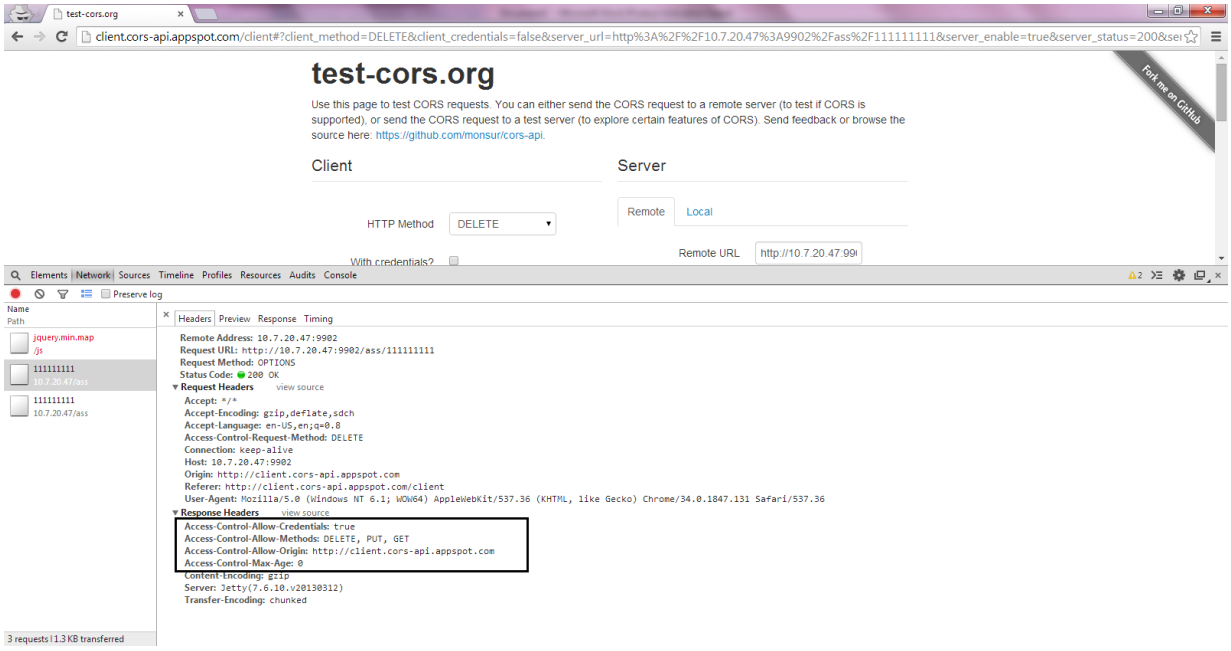


16 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.



17 The response headers are also observed viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Methods: DELETE, PUT, GET
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Max-Age: 0



18 The same can be seen in usage data for the service by enabling auditing.

19 Since the preflight was successful, the actual DELETE request is sent following that by the browser.

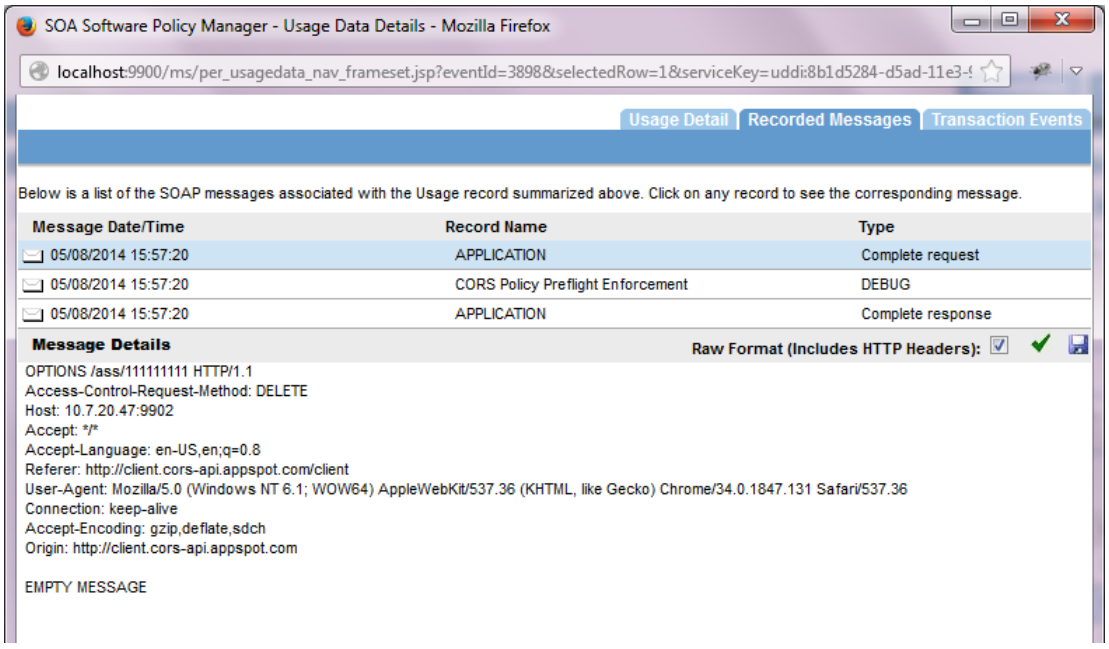
20 The request and response headers viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Expose-Headers: x-response-for-cors-pass2

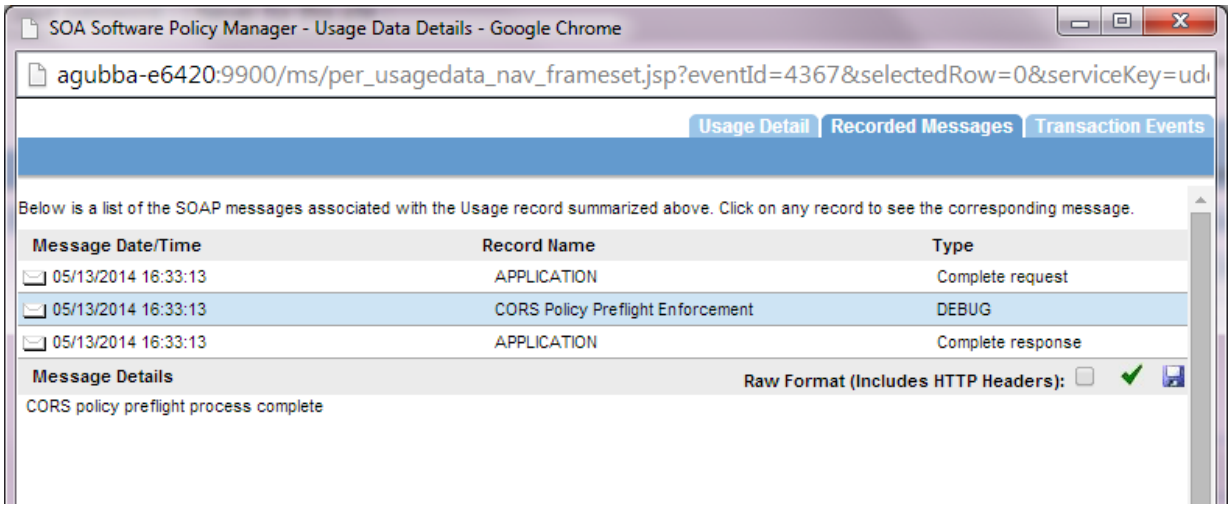
21 All the headers provided in Expose headers in the policy should be displayed.

22 The same can be seen in usage data for the service by enabling auditing.

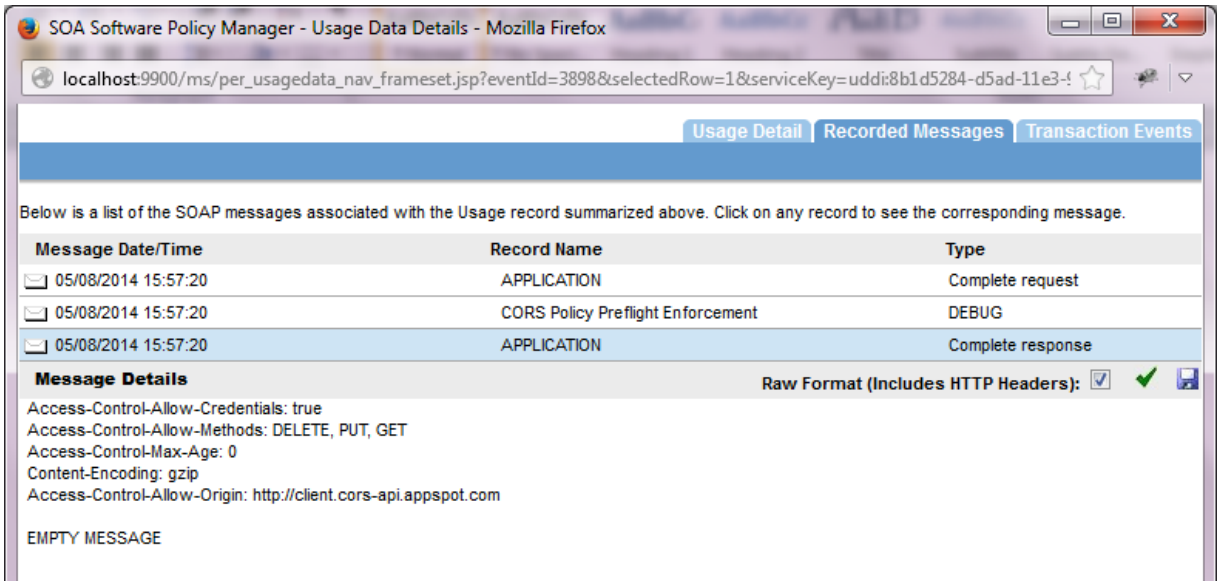
Preflight request message:



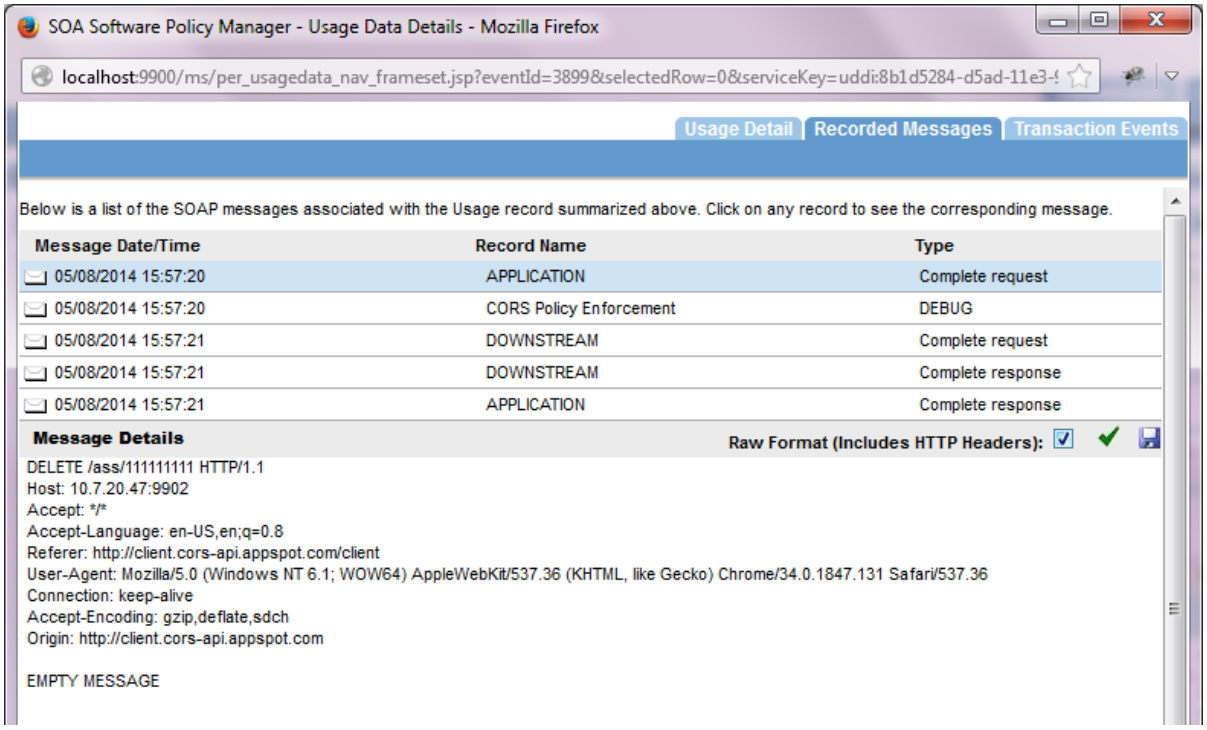
Preflight policy enforcement:



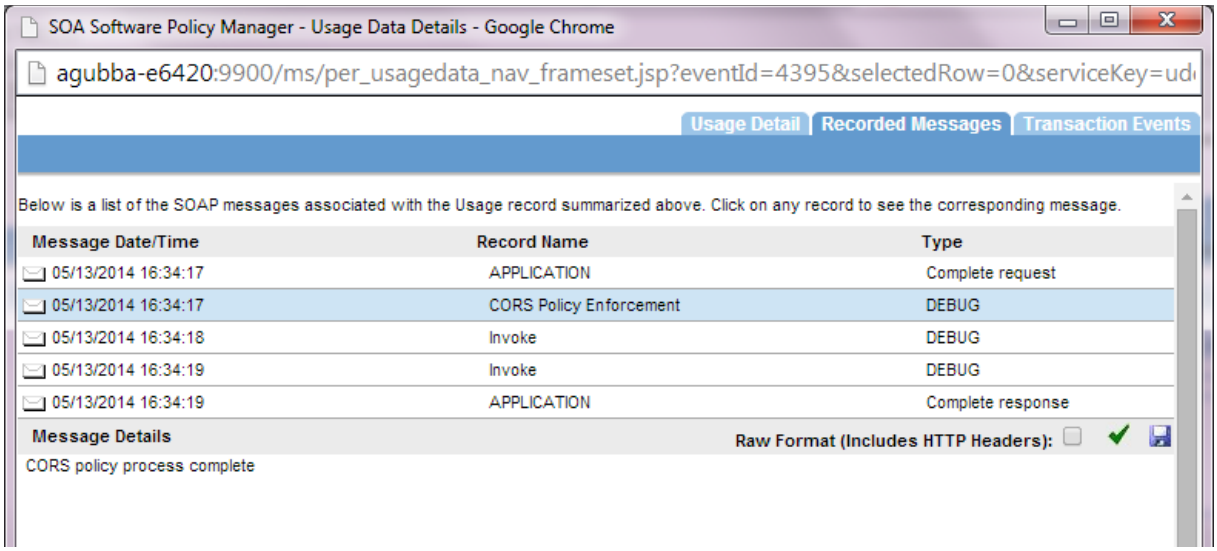
Preflight response message:



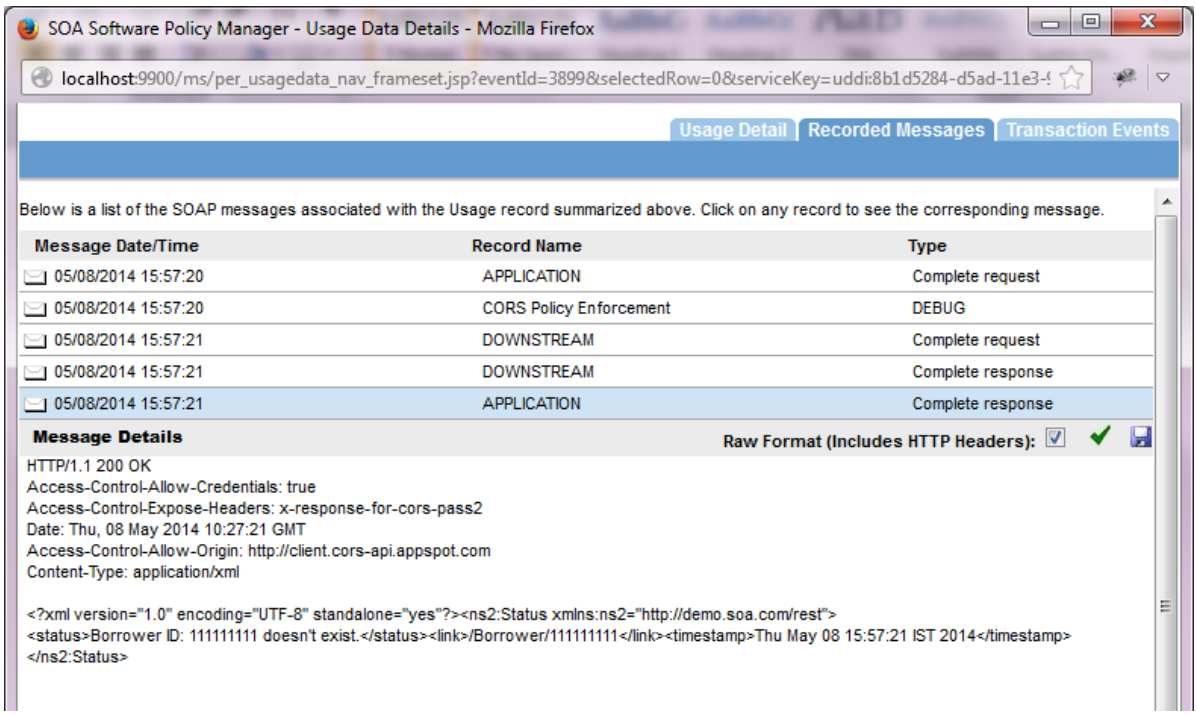
DELETE request:



CORS policy enforcement during actual request:



Delete response message:



Scenario 2: CORS with Credentials (Basic Authentication Policy)

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS** – REST Borrower)

3 Virtualize the PS on Network Director (Example service name: **VS1**)

4 Create CORS policy with the following configuration:

5 Allow Credentials component must be set to **true**:

- Max Age: 0
- Allow Credentials: true
- Allow Origins: http://client.cors-api.appspot.com
- Allow Headers:
- Expose Headers: x-response-for-cors-pass2

6 Attach the policy to **VS1**.

7 Attach Basic authentication policy to **VS1**.

8 Navigate to client: http://client.cors-api.appspot.com/client

9 Provide URL of the **VS1** resource for DELETE method.

Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

10 Select Delete Method.

11 Select with Credentials check box in the client.

12 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

13 View network monitoring for the browser.

14 Send request to DELETE method.

15 The client prompts for credentials.

16 Provide valid credentials for basic authentication.

17 Initially a preflight message is sent (OPTION method is used to send preflight request).

18 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

19 If the Allow Credentials flag is **true** in the policy, the DELETE request is sent to the service.

20 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.

21 The response headers are also observed viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Methods: DELETE, PUT, GET
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Max-Age: 0

22 The same can be seen in usage data for the service by enabling auditing.

23 Since the preflight was successful, the actual DELETE request is sent following that by the browser.

24 The request and response headers viz.,

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Expose-Headers: x-response-for-cors-pass2

25 The same can be seen in usage data for the service by enabling auditing.

Scenario 3: Allow All Origins

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS -** REST Borrower)

3 Virtualize the **PS** on Network Director (Example service name: **VS1**).

4 Create CORS policy the following configuration:

- Max Age: 0
- Allow Credentials: true
- Allow Origins: *
- Allow Headers:
- Expose Headers: x-response-for-cors-pass2

5 Attach the policy to **VS1**.

6 Navigate to client: http://client.cors-api.appspot.com/client.

7 Provide URL of the **VS1** resource for DELETE method.

Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

8 Select Delete Method.

9 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

10 Navigate to Network tab.

11 Send request to DELETE method.

12 Initially a preflight message is sent (OPTION method is used to send preflight request).

13 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

14 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.

15 The response headers are also observed viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Methods: DELETE, PUT, GET
- Access-Control-Allow-Origin: *
- Access-Control-Max-Age: 0

16 All the Origins should be allowed. This can be verified by sending the request from a different client.

Scenario 4: Allow Custom Headers (Cookie Authentication)

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS** –REST Borrower)

3 Virtualize the **PS** on Network Director (Example service name: **VS1**).

4 Create CORS policy with the following configuration. Allow Headers component must be set to **Cookie** so as to allow cookie to be passed to the policy:

- Max Age: 0
- Allow Credentials: true
- Allow Origins: http://client.cors-api.appspot.com
- Allow Headers: **Cookie**
- Expose Headers: x-response-for-cors-pass2

5 Attach the policy to **VS1**.

6 Add Cookie Identity system.

7 Attach Cookie authentication policy to **VS1**.

8 Navigate to client: http://client.cors-api.appspot.com/client .

9 Provide URL of the VS1 resource for DELETE method.

Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

10 Select Delete Method.

11 Select with Credentials check box in the client.

12 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

13 View network monitoring for the browser.

14 Send request to DELETE method.

15 The client prompts for credentials.

16 Provide valid credentials for basic authentication.

17 Initially a preflight message is sent (OPTION method is used to send preflight request).

18 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

19 If the Allow Credentials flag is **true** in the policy, the DELETE request is sent to the service.

20 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.

21 The response headers are also observed viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Methods: DELETE, PUT, GET
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Max-Age: 0

22 The same can be seen in usage data for the service by enabling auditing.

23 Since the preflight was successful, the actual DELETE request is sent following that by the browser.

24 In the response for DELETE request, Authentication Cookie is generated.

25 Send Cookie in Custom Headers in the Client and Disable credentials option.

26 Send request, and the request should pass the Cookie. Cookie authentication should be successful when sending a valid cookie.

Scenario 5: CORS Preflight Limitation within MaxAge Defined (in seconds)

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS** – REST Borrower)

3 Virtualize the **PS** on Network Director (Example service name: **VS1**)

4 Create CORS policy with the following configuration:

- Max Age: 10
- Allow Credentials: true
- Allow Origins: http://client.cors-api.appspot.com
- Allow Headers:
- Expose Headers: x-response-for-cors-pass2

5 Attach the policy to **VS1**.

6 Navigate to client: http://client.cors-api.appspot.com/client

7 Provide URL of the **VS1** resource for DELETE method.

Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

8 Select Delete Method.

9 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

10 View network monitoring for the browser.

11 Send request to DELETE method.

12 Initially a preflight message is sent (OPTION method is used to send preflight request).

13 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

14 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.

15 The response headers are also observed viz.:

- Access-Control-Allow-Credentials: true
- Access-Control-Allow-Methods: DELETE, PUT, GET
- Access-Control-Allow-Origin: http://client.cors-api.appspot.com
- Access-Control-Max-Age: 0

16 The same can be seen in usage data for the service by enabling auditing.

17 Since the preflight was successful, the actual DELETE request is sent following that by the browser.

18 Send a request again from client with in the MaxAge limit (i.e., 10 seconds).

19 Preflight request will not be sent as the previous preflight request is cached already.

20 After the MaxAge limit is exceeded, send a request again.

21 Now the request should again be preflighted.

Scenario 6: Origins Mismatch during Preflight

1 Create Policy Manager instance.

2 Create a physical REST service (Example service name: **PS** – REST Borrower).

3 Virtualize the **PS** on Network Directory (Example service name: **VS1**).

4 Create CORS policy with the following configuration:

- Max Age: 10
- Allow Credentials: true
- Allow Origins: http://abc.com
- Allow Headers:
- Expose Headers: x-response-for-cors-pass2

5 Attach the policy to **VS1**.

6 Navigate to client: http://client.cors-api.appspot.com/client .

7 Provide URL of the **VS1** resource for DELETE method.

Example: http://10.7.20.47:9902/ass/111111111 - Used IP address to hit my host.

8 Select Delete Method.

9 Enable Developer tool for Browser (F12 for Chrome or Firebug in case of Firefox).

10 View network monitoring for the browser.

11 Send request to DELETE method.

12 Initially a preflight message is sent (OPTION method is used to send preflight request).

13 After the success of preflight request, depending on the headers sent in response of preflight, the browser sends the actual DELETE request to the API.

14 The headers sent by the client for preflight viz., Access-Control-Request-Method and Origin can be seen by clicking on the options request.

15 Since the Origin is not a match, preflight is not sent and the request is processed as usual.

16 The response headers related to CORS are also not seen.