Navneet

Security headers

**Table of Contents**

[X-XSS-Protection (read here) 2](#_Toc2336311)

[Payload 2](#_Toc2336312)

[Code (/WEB-INF/result/X-XSS-Protection.jsp) 2](#_Toc2336313)

[Experiment 2](#_Toc2336314)

[Observation #1 2](#_Toc2336315)

[Access-Control-Allow-Origin (read here) 3](#_Toc2336316)

[Code: /WEB-INF/result/cors.jsp 3](#_Toc2336317)

[Experiment 3](#_Toc2336318)

[Observation #1 3](#_Toc2336319)

[Cache-Control (read here) 4](#_Toc2336320)

[Code (/WEB-INF/test/Cache-Control.jsp) 4](#_Toc2336321)

[Experiment 4](#_Toc2336322)

[Observation #1 4](#_Toc2336323)

[Observation #2 5](#_Toc2336324)

[Referrer-Policy (Read Here) 6](#_Toc2336325)

[Code (/WEB-INF/test/referrer-policy.jsp) 6](#_Toc2336326)

[Experiment 6](#_Toc2336327)

[Observation #1 6](#_Toc2336328)

[Observation #2 8](#_Toc2336329)

[X-Frame-Options (read here) 9](#_Toc2336330)

[Code(/WEB-INF/test/Bank.jsp) 9](#_Toc2336331)

[Experiment 9](#_Toc2336332)

[Observation #1 9](#_Toc2336333)

[Observation #2 9](#_Toc2336334)

[Content Security Policy (read here) 11](#_Toc2336335)

[Code(/WEB-INF/test/Content-Security-Policy.jsp) 11](#_Toc2336336)

[Experiment 11](#_Toc2336337)

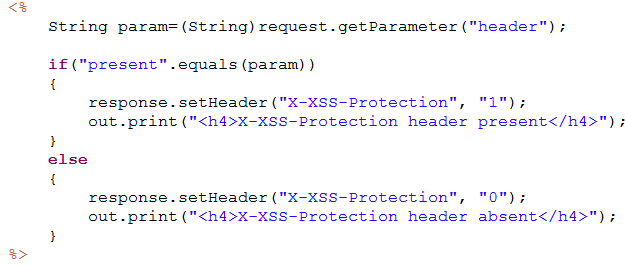
[Observation #1 11](#_Toc2336338)

# X-XSS-Protection ([read here](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection))

## Payload

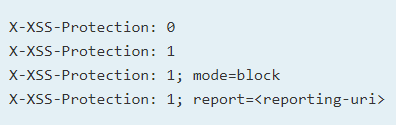
* <script>alert(‘XSS’)</alert>

## Code (/WEB-INF/result/X-XSS-Protection.jsp)



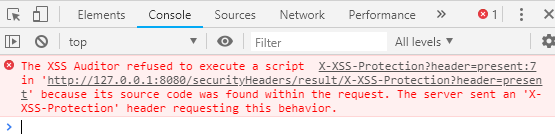
## Experiment

Play around with the following values to observe browser response.



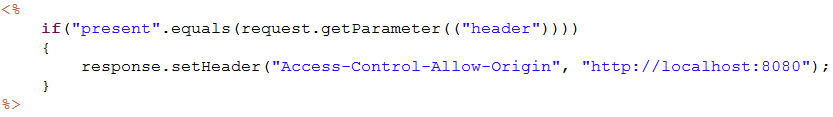
## Observation #1

* X-XSS-Protection: 1



# Access-Control-Allow-Origin ([read here](https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS))

## Code: /WEB-INF/result/cors.jsp



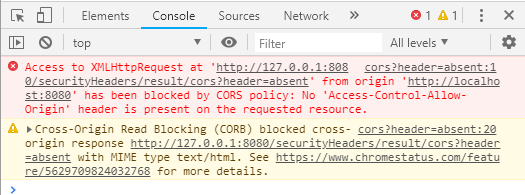
## Experiment

There are various other headers, which you can experiment with -

* Access-Control-Allow-Methods
* Access-Control-Allow-Credentials
* Access-Control-Max-Age
* Access-Control-Expose-Headers

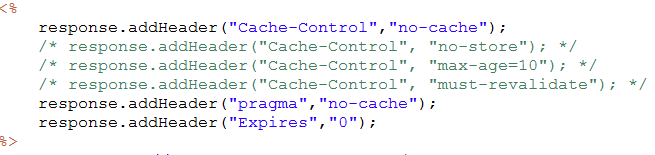
## Observation #1

* Access-Control-Allow-Origin: http://localhost:8080



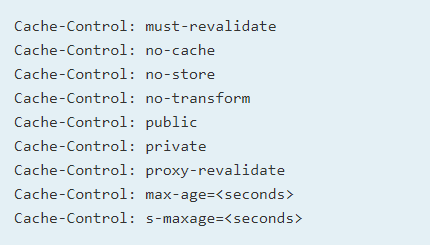
# Cache-Control ([read here](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control))

## Code (/WEB-INF/test/Cache-Control.jsp)



## Experiment

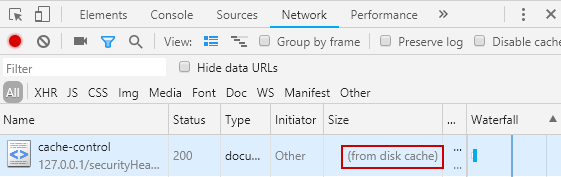
Play around with different cache-control directive to see how it affects browser’s caching behavior.



## Observation #1

* Cache-Control: no-cache

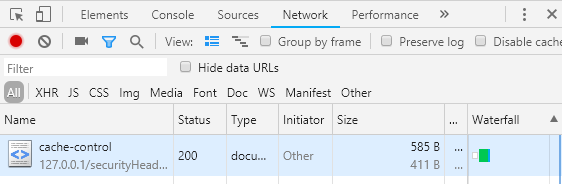
Browser actually caches the web page.



## Observation #2

* Cache-Control: no-store

Browser does not cache web page. Every time it fetches page from server.



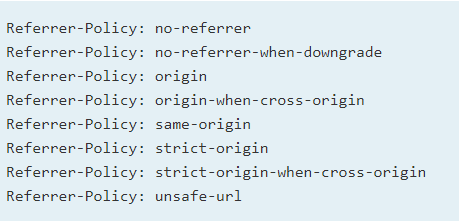
# Referrer-Policy ([Read Here](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control))

## Code (/WEB-INF/test/referrer-policy.jsp)



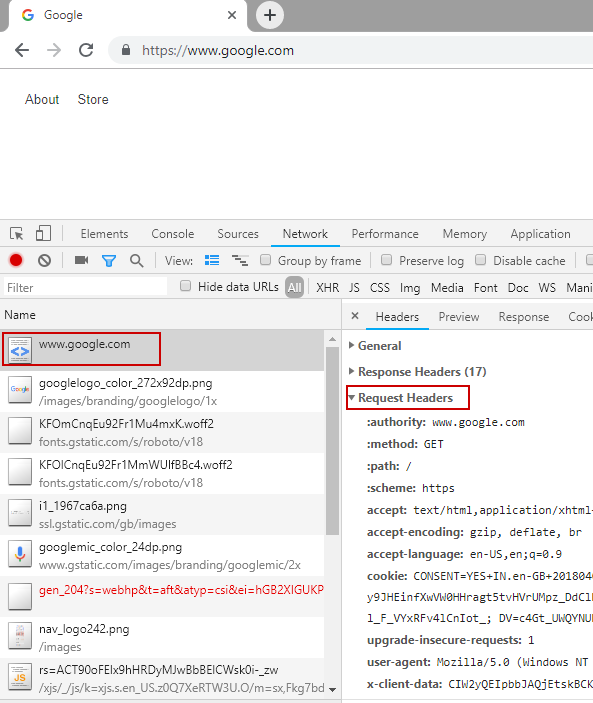
## Experiment

Play around with various directives to see how the browsers behaves.



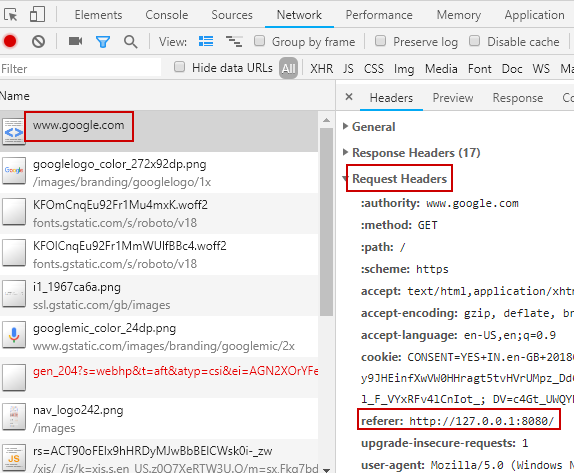
## Observation #1

* referrer-policy: no-referrer



## Observation #2

* referrer-policy:origin-when-cross-origin

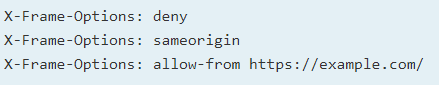


# X-Frame-Options ([read here](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options))

## Code(/WEB-INF/test/Bank.jsp)

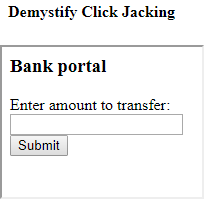


## Experiment



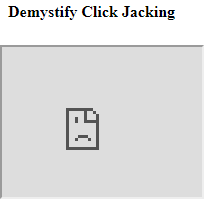
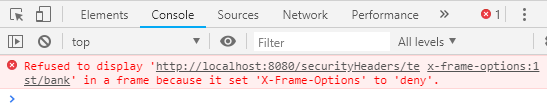
## Observation #1

* X-Frame-Options: sameorigin



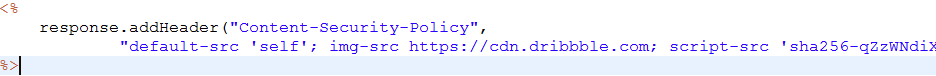
## Observation #2

* X-Frame-Options: deny



# Content Security Policy ([read here](https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP))

## Code(/WEB-INF/test/Content-Security-Policy.jsp)



## Experiment

There are various options available to experiment with

* report-uri
* unsafe-inline

## Observation #1

