

[thepolyglotdeveloper.com](https://www.thepolyglotdeveloper.com)

# Bypass CORS Errors When Testing APIs Locally

*Nic Raboy*

4 Minuten

---

Our website is made possible by displaying online advertisements to our visitors. Please consider supporting us by disabling your ad blocker.

Anyone who has worked with a RESTful API using JavaScript knows that testing can be a complete pain if the API owner hasn't enabled CORS on their server. So what is CORS?

According to [Wikipedia](#), it is the following:

Cross-origin resource sharing (CORS) is a mechanism that allows many resources (e.g., fonts, JavaScript, etc.) on a web page to be requested from another domain outside the domain the resource originated from.

Often API owners will leave CORS disabled even though their API is open to the public. In my opinion it doesn't feel public if the API owner is not allowing requests from all angles.

Here are a few tricks I've picked up in regards to bypassing the awful CORS errors you receive in your browser when testing.

**Advertise to  
The Polyglot Developer  
audience!**

[LEARN MORE](#)

Note that there could still be scenarios where these tricks fail to work. These tricks should help regardless.

## Altering the security of your web browser

I'd start by fiddling with your web browser to resolve this issue. It is the easiest and quickest in my opinion.

### Google Chrome & Chromium

Chrome has a wonderful command line argument to disable web security. To start Chrome with web security disabled, run the following command:

```
/Applications/Google\ Chrome.app/Contents/MacOS  
/Google\ Chrome --disable-web-security
```

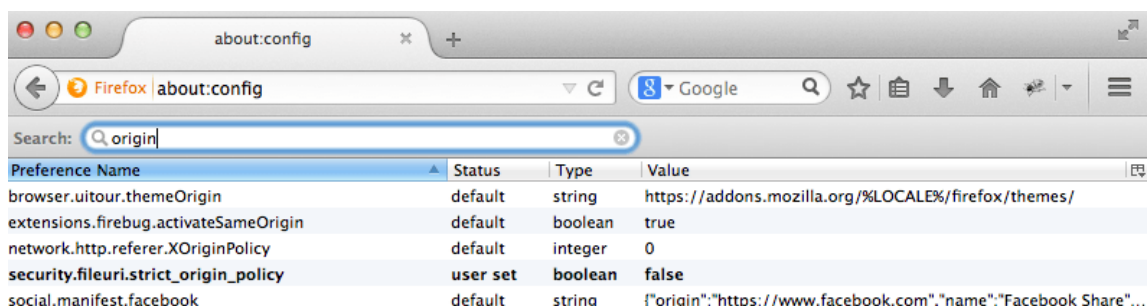
If you're using Windows, navigate to the path of your installation via a command prompt and run the following:

```
Chrome.exe --disable-web-security
```

### Mozilla Firefox

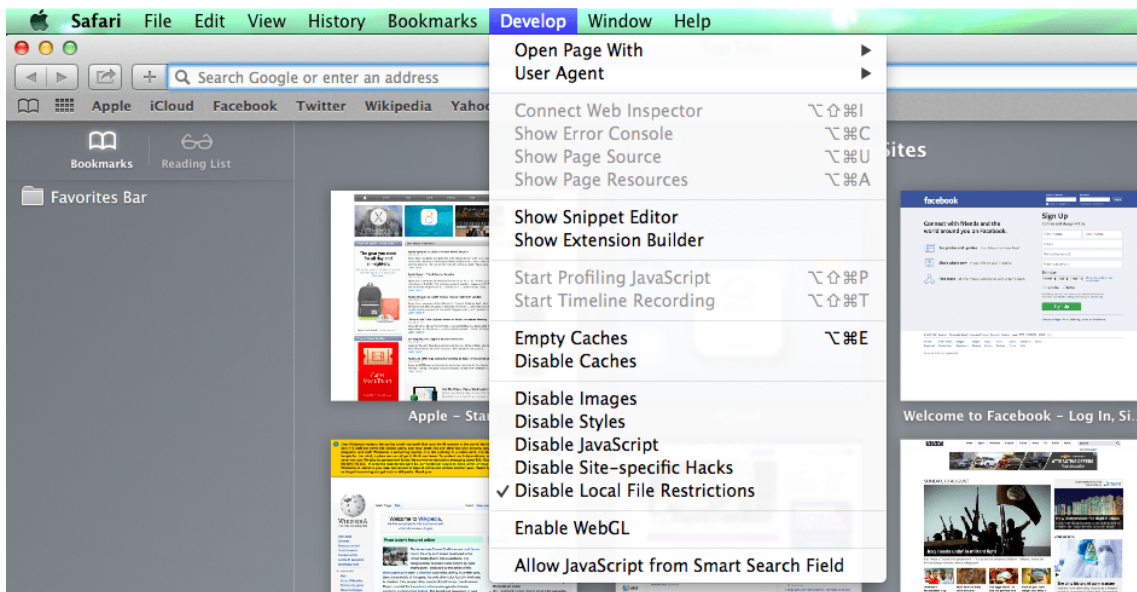
It is often debatable if this actually works, but it worked for me. Open your Firefox web browser and type `about:config` into the URL bar. Agree to the statement about risk and do a search for: `security.fileuri.strict_origin_policy`

This setting should be changed to false.



## Safari

I personally use Safari for my API testing. Not sure why, but I'm not complaining. Start by enabling the **Develop** menu from **Preferences -> Advanced**. When this is done you may need to restart Safari. In the **Develop** menu make sure that **Disable Local File Restrictions** is checked. That is all there is too it.



## Start up a small server

There could be a scenario where your requests are still giving you a hard time. Maybe the browsers don't like making requests from **file:///** or maybe there is another reason.

If you've got Python installed (most Mac and Linux users do), you can start a quick local web server for testing. Using the command prompt, navigate to the directory that has your HTML files and run the following command:

```
python -m SimpleHTTPServer
```

Your files should now be accessible from **http://localhost:8000/** and may have a good chance of working when **file:///** does not.

## Ask the server owner politely to add CORS

## support

It doesn't take much effort to enable cross origin resource sharing on a server. As mentioned on [enable-cors.org](https://enable-cors.org), the owner only needs to add `Access-Control-Allow-Origin: *` to the response header. There are even instructions on how to do this in various programming languages, all of which are not too difficult and make a world of difference to developers experiencing these errors.

**Advertise to**  
**The Polyglot Developer**  
**audience!**

LEARN MORE



## **Nic Raboy**

Nic Raboy is an advocate of modern web and mobile development technologies. He has experience in Java, JavaScript, Golang and a variety of frameworks such as Angular, NativeScript, and Apache Cordova. Nic writes about his development experiences related to making web and mobile development easier to understand.