Find the fix for:



1. Uncaught Type Error:

Make sure that which ever variables throws undefined error, is assigned a value to it.

1. TypeError: 'undefined' is not an object

Can check in the following way (extra = and making length a truthy evaluation)

if (typeof(sub.from) !== 'undefined' && sub.from.length)

3.TypeError null is not an Object:

Javascript Error Null is not an Object.

Put the code so it executes after the elements are defined, either with a DOM ready callback or place the source under the elements in the HTML.

4.Script Error: Script error” is what browsers send to the onerror callback when an error originates from a JavaScript file served from a different origin (different domain, port, or protocol). It’s painful because, even though there’s an error occurring, you don’t know what the error is, nor from which code it’s originating. And that’s the whole purpose of window.onerror — getting insight into uncaught errors in your application.

The Fix: CORS Attributes and Headers

In order to get visibility into a JavaScript exception thrown by scripts originating from different origins, you must do two things.

1. Add a crossorigin=”anonymous” script attribute

<script src="http://another-domain.com/app.js" crossorigin="anonymous"></script>

1. Add a cross-origin HTTP header: Access-Control-Allow-Origin: \*
2. An Alternative Solution: try/catch: Sometimes we’re not in a position to adjust the HTTP headers of scripts our web application is consuming. In those situations, there’s an alternative approach: using try/catch.

5. Typeerror object doesn't support property

Instead of this line,

myObjects = Object.values(deneme);

write,

myObjects = Object.keys(deneme).map(itm => deneme[itm]);

Because, Object.values is an experimental feature and it is not being supported in IE.

6. uncaught range error maximum call stack size exceeded

It means that somewhere in your code, you are calling a function which in turn calls another function and so forth, until you hit the call stack limit.

This is almost always because of a recursive function with a base case that isn't being met.

Viewing the stack

Consider this code...

(function a() {

a();

})();

7. Uncaught TypeError: Cannot read property 'length':

You are not passing the variable correctly. One fast olution is to make a global variable.

8. referenceerror event is not defined

You're declaring (some of) your event handlers incorrectly.

You need "event" to be a parameter to the handlers. WebKit follows IE's old behavior of using a global symbol for "event", but Firefox doesn't. When you're using jQuery, that library normalizes the behavior and ensures that your event handlers are passed the event parameter.