

How to Fix ReferenceError: \$ is Not Defined in JavaScript

Apr 24, 2023



Table of Contents

Summarize and analyze this article with 🎯

🤖 Google AI Mode or 💬 ChatGPT or 🔍 Perplexity or 🤖 Claude or 🦇 Grok (X).

The JavaScript `ReferenceError: $ is not defined` occurs when the jQuery library is used but is not properly loaded or is not available in the current scope of the code.

In JavaScript, the `$` symbol is often used as a shorthand alias for the jQuery library. This error indicates that the code is trying to use jQuery functionality by referencing the `$` symbol, but the library is not available or has not been loaded correctly.

What Causes Javascript ReferenceError: \$ is Not Defined

Ask AI

This error can occur for several reasons, such as:

- The jQuery library is not included in the script that references it.
- The library is included with a typo or error in the script
- The script is executed before the jQuery library is loaded, which can be caused by incorrect placement of the `script` tag in the HTML file.
- The code is running in a different scope or environment where the `$` symbol is not defined or has a different meaning.

ReferenceError: \$ is Not Defined Example

Here's an example of a Javascript `ReferenceError: $ is not defined` thrown when jQuery is not properly loaded in a script that uses it:

```
<html>
<head>
    <title>Example</title>
</head>
<body>
    <button id="myButton">Click me</button>
    <script>
        $(document).ready(function() {
            $('#myButton').click(function() {
                alert('Button clicked!');
            });
        });
    </script>
</body>
</html>
```

In this example, the code is trying to use the jQuery library. Since jQuery is not included in the above script, running it throws the error:

Uncaught ReferenceError: \$ is not defined

How to Fix ReferenceError: \$ is Not Defined

To fix the `ReferenceError: $ is not defined` error, jQuery should be properly loaded before any code that references it is executed. This can be done by including the library in the HTML file using a `script` tag, either by downloading the library and hosting it locally or by using a content delivery network (CDN) link.

Any syntax errors or typos in the script should also be checked, as they may prevent jQuery from loading correctly.

The earlier example can be updated to fix the error by using the above approach:

```
<html>
<head>
    <title>Example</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</head>
<body>
    <button id="myButton">Click me</button>
    <script>
        $(document).ready(function() {
            $('#myButton').click(function() {
                alert('Button clicked!');
            });
        });
    </script>
</body>
</html>
```

Here, the jQuery library is included using a `script` tag with a CDN link. This makes it available to be used in the script, which avoids the `ReferenceError: $ is not defined` error.

Track, Analyze and Manage Errors With Rollbar

Managing errors and exceptions in your code is challenging. It can make deploying production code an unnerving experience. Being able to track, analyze, and manage errors in real-time can help you to proceed with more confidence. Rollbar automates error monitoring and triaging, making fixing JavaScript errors easier than ever. [Try it today!](#)

Related Resources

5 ways to reduce noise when logging your JavaScript exceptions

How to Catch (and Fix) JavaScript TypeErrors

How to Fix EvalError in JavaScript



Build with confidence. Release with clarity.

Rollbar helps you track what breaks, understand why, and improve what comes next.

- ✓ 5K free events per month, forever
- ✓ 14-day full feature trial
- ✓ Easy and quick installation

[Get started in minutes](#)

Plans starting at \$0. Take off with our 14-day full feature Free Trial.

PRODUCT

	DOCUMENTATION
Product	Docs Overview
Pricing	Setting up Rollbar
Customers	Notifications
Platforms	Deploy Tracking
Integrations	Telemetry
Changelog	Security & Compliance
Roadmap	API

LANGUAGES/FRAMEWORKS

JavaScript	Go
iOS	Java
Next.js	.NET
React	Node.js
React Native	PHP
Rails	Python
Ruby	Django
More...	

COMPANY

About Us
Careers
Contact Us

