DOMException

The DOMException interface represents an abnormal event (called an exception) that occurs as a result of calling a method or accessing a property of a web API. This is how error conditions are described in web APIs.

Each exception has a name, which is a short "PascalCase"-style string identifying the error or abnormal condition.

DOMException is a Serializable object, so it can be cloned with structuredClone() or copied between Workers using postMessage().

**Constructor**

DOMException()

Returns a DOMException object with a specified message and name.

The DOMException() constructor returns a DOMException object with a specified message and name.

Syntax

JS

Copy to Clipboard

new DOMException()

new DOMException(message)

new DOMException(message, name)

**Parameters**

1. message Optional

A description of the exception. If not present, the empty string '' is used.

1. name Optional

A string. If the specified name is a standard error name, then getting the code property of the DOMException object will return the code number corresponding to the specified name.

1. Return value

A newly created DOMException object.

**Examples**

In this example, pressing the button causes a custom DOMException to be thrown, which is then caught and the custom error message shown in an alert.

HTML

Play

Copy to Clipboard

<button>Trigger DOM Exception</button>

<p id="output"></p>

JS

const button = document.querySelector("button");

button.onclick = () => {

try {

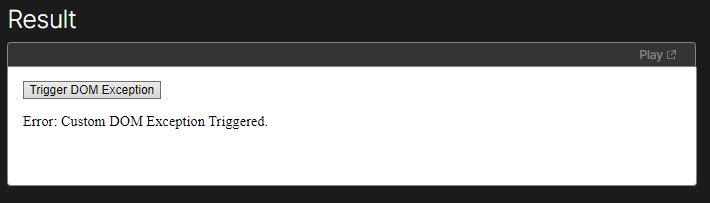
throw new DOMException("Custom DOM Exception Triggered.");

} catch (error) {

document.querySelector("#output").textContent = `Error: ${error.message}`;

}

};



**Instance properties**

1. DOMException.message Read only

Returns a string representing a message or description associated with the given error name.

1. DOMException.name Read only

Returns a string that contains one of the strings associated with an error name.

[**Error names**](https://developer.mozilla.org/en-US/docs/Web/API/DOMException#error_names)

Common error names are listed here. Some APIs define their own sets of names, so this is not necessarily a complete list.

Note that the following deprecated historical errors don't have an error name but instead have only a legacy constant code value and a legacy constant name:

* Legacy code value: 2, legacy constant name: DOMSTRING\_SIZE\_ERR
* Legacy code value: 6, legacy constant name: NO\_DATA\_ALLOWED\_ERR
* Legacy code value: 16, legacy constant name: VALIDATION\_ERR

**Note:** Because historically the errors were identified by a numeric value that corresponded with a named variable defined to have that value, some of the entries below indicate the legacy code value and constant name that were used in the past.

* **IndexSizeError**

The index is not in the allowed range. For example, this can be thrown by the [Range](https://developer.mozilla.org/en-US/docs/Web/API/Range) object. (Legacy code value: 1 and legacy constant name: INDEX\_SIZE\_ERR)

* **HierarchyRequestError**

The node tree hierarchy is not correct. (Legacy code value: 3 and legacy constant name: HIERARCHY\_REQUEST\_ERR)

* **WrongDocumentError**

The object is in the wrong [Document](https://developer.mozilla.org/en-US/docs/Web/API/Document). (Legacy code value: 4 and legacy constant name: WRONG\_DOCUMENT\_ERR)

* **InvalidCharacterError**

The string contains invalid characters. (Legacy code value: 5 and legacy constant name: INVALID\_CHARACTER\_ERR)

* **NoModificationAllowedError**

The object cannot be modified. (Legacy code value: 7 and legacy constant name: NO\_MODIFICATION\_ALLOWED\_ERR)

* **NotFoundError**

The object cannot be found here. (Legacy code value: 8 and legacy constant name: NOT\_FOUND\_ERR)

* **NotSupportedError**

The operation is not supported. (Legacy code value: 9 and legacy constant name: NOT\_SUPPORTED\_ERR)

* **InvalidStateError**

The object is in an invalid state. (Legacy code value: 11 and legacy constant name: INVALID\_STATE\_ERR)

* **InUseAttributeError**

The attribute is in use. (Legacy code value: 10 and legacy constant name: INUSE\_ATTRIBUTE\_ERR)

* **SyntaxError**

The string did not match the expected pattern. (Legacy code value: 12 and legacy constant name: SYNTAX\_ERR)

* **InvalidModificationError**

The object cannot be modified in this way. (Legacy code value: 13 and legacy constant name: INVALID\_MODIFICATION\_ERR)

* **NamespaceError**

The operation is not allowed by Namespaces in XML. (Legacy code value: 14 and legacy constant name: NAMESPACE\_ERR)

* **InvalidAccessError**

The object does not support the operation or argument. (Legacy code value: 15 and legacy constant name: INVALID\_ACCESS\_ERR)

* **SecurityError**

The operation is insecure. (Legacy code value: 18 and legacy constant name: SECURITY\_ERR)

* **NetworkError Experimental**

A network error occurred. (Legacy code value: 19 and legacy constant name: NETWORK\_ERR)

* **AbortError Experimental**

The operation was aborted. (Legacy code value: 20 and legacy constant name: ABORT\_ERR)

* **URLMismatchError** **Experimental**

The given URL does not match another URL. (Legacy code value: 21 and legacy constant name: URL\_MISMATCH\_ERR)

* **QuotaExceededError Experimental**

The quota has been exceeded. (Legacy code value: 22 and legacy constant name: QUOTA\_EXCEEDED\_ERR)

* **TimeoutError**

The operation timed out. (Legacy code value: 23 and legacy constant name: TIMEOUT\_ERR)

* **InvalidNodeTypeError Experimental**

The node is incorrect or has an incorrect ancestor for this operation. (Legacy code value: 24 and legacy constant name: INVALID\_NODE\_TYPE\_ERR)

* **DataCloneError Experimental**

The object can not be cloned. (Legacy code value: 25 and legacy constant name: DATA\_CLONE\_ERR)

* **EncodingError Experimental**

The encoding or decoding operation failed (No legacy code value and constant name).

* **NotReadableError Experimental**

The input/output read operation failed (No legacy code value and constant name).

* **UnknownError Experimental**

The operation failed for an unknown transient reason (e.g. out of memory) (No legacy code value and constant name).

* **ConstraintError Experimental**

A mutation operation in a transaction failed because a constraint was not satisfied (No legacy code value and constant name).

* **DataError Experimental**

Provided data is inadequate (No legacy code value and constant name).

* **TransactionInactiveError Experimental**

A request was placed against a transaction that is currently not active or is finished (No legacy code value and constant name).

* **ReadOnlyError Experimental**

The mutating operation was attempted in a "readonly" transaction (No legacy code value and constant name).

* **VersionError Experimental**

An attempt was made to open a database using a lower version than the existing version (No legacy code value and constant name).

* **OperationError Experimental**

The operation failed for an operation-specific reason (No legacy code value and constant name).

* **NotAllowedError**

The request is not allowed by the user agent or the platform in the current context, possibly because the user denied permission (No legacy code value and constant name).