/// mdn web docs __

IDBDatabase: createObjectStore() method

Baseline Widely available

> **Note:** This feature is available in <u>Web Workers</u>.

The `createObjectStore()` method of the <u>IDBDatabase</u> interface creates and returns a new <u>IDBObjectStore</u>.

The method takes the name of the store as well as a parameter object that lets you define important optional properties. You can use the property to uniquely identify individual objects in the store. As the property is an identifier, it should be unique to every object, and every object should have that property.

This method can be called *only* within a <u>versionchange</u> transaction.

Syntax

```js
createObjectStore(name)
createObjectStore(name, options)
```

Parameters

`name`

The name of the new object store to be created. Note that it is possible to create an object store with an empty name.

`options` (Optional)

An options object whose attributes are optional parameters to the method. It includes the following properties:

> `keyPath` (Optional)
>
> The <u>key path</u> to be used by the new object store. If empty or not specified, the object store is created without a key path and uses <u>out-of-line keys</u>. You can also pass in an array

as a `keyPath`.

> **autoIncrement** (Optional)
>
>   If `true`, the object store has a [key generator](). Defaults to `false`.

Return value

A new [IDBObjectStore]().

Exceptions

This method may raise a [DOMException]() with a `name` of one of the following types:

`ConstraintError` [DOMException]()

   Thrown if an object store with the given name (based on a case-sensitive comparison) already exists in the connected database.

`InvalidAccessError` [DOMException]()

   Thrown if `autoIncrement` is set to true and `keyPath` is either an empty string or an array.

`InvalidStateError` [DOMException]()

   Thrown if the method was not called from a `versionchange` transaction callback.

`SyntaxError`

   Thrown if the `keyPath` option contains an invalid key path.

`TransactionInactiveError` [DOMException]()

   Thrown if a request is made on a source database that does not exist (for example, when the database has been deleted or removed), or if the associated upgrade transaction has completed or is processing a request.

Examples

JS

```js
// Let us open our database
const request = window.indexedDB.open("toDoList", 4);

// This handler is called when a new version of the database
// is created, either when one has not been created before
// or when a new version number is submitted by calling
// window.indexedDB.open().
// This handler is only supported in recent browsers.
request.onupgradeneeded = (event) => {
```

```js
  const db = event.target.result;

  db.onerror = (event) => {
    note.appendChild(document.createElement("li")).textContent =
      "Error loading database.";
  };

  // Create an objectStore for this database

  const objectStore = db.createObjectStore("toDoList", {
    keyPath: "taskTitle",
  });

  // define what data items the objectStore will contain

  objectStore.createIndex("hours", "hours", { unique: false });
  objectStore.createIndex("minutes", "minutes", { unique: false });
  objectStore.createIndex("day", "day", { unique: false });
  objectStore.createIndex("month", "month", { unique: false });
  objectStore.createIndex("year", "year", { unique: false });

  objectStore.createIndex("notified", "notified", { unique: false });

  note.appendChild(document.createElement("li")).textContent =
    "Object store created.";
};
```

Specifications

Specification
[Indexed Database API 3.0](#)   [# ref-for-dom-idbdatabase-createobjectstore](#)①

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

		Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android	WebView on iOS
createObjectStore		23	12	10	15	8	25	22	14	8	1.5	4.4	8

Tip: you can click/tap on a cell for more information.

Full support

See also

- [Using IndexedDB](#)

- Starting transactions: `IDBDatabase`

- Using transactions: `IDBTransaction`

- Setting a range of keys: `IDBKeyRange`

- Retrieving and making changes to your data: `IDBObjectStore`

- Using cursors: `IDBCursor`

- Reference example: [To-do Notifications]   ([View the example live]   ).

Help improve MDN

Was this page helpful to you?

[Yes]   [No]

[Learn how to contribute](#).

This page was last modified on Dec 16, 2024 by [MDN contributors](#).