Blob

The **`Blob`** object represents a blob, which is a file-like object of immutable, raw data; they can be read as text or binary data, or converted into a `ReadableStream` so its methods can be used for processing the data.

Blobs can represent data that isn't necessarily in a JavaScript-native format. The `File` interface is based on `Blob`, inheriting blob functionality and expanding it to support files on the user's system.

Using blobs

To construct a `Blob` from other non-blob objects and data, use the `Blob()` constructor. To create a blob that contains a subset of another blob's data, use the `slice()` method. To obtain a `Blob` object for a file on the user's file system, see the `File` documentation.

The APIs accepting `Blob` objects are also listed in the `File` documentation.

Constructor

`Blob()`

Returns a newly created `Blob` object which contains a concatenation of all of the data in the array passed into the constructor.

Instance properties

`Blob.prototype.size` | **Read only**

The size, in bytes, of the data contained in the `Blob` object.

`Blob.prototype.type` | **Read only**

A string indicating the MIME type of the data contained in the `Blob`. If the type is unknown, this string is empty.

Instance methods

`Blob.prototype.arrayBuffer()`

Returns a promise that resolves with an `ArrayBuffer` containing the entire contents of the `Blob` as binary data.

`Blob.prototype.slice()`

Returns a new `Blob` object containing the data in the specified range of bytes of the blob on which it's called.

`Blob.prototype.stream()`

Returns a `ReadableStream` that can be used to read the contents of the `Blob`.

`Blob.prototype.text()`

Returns a promise that resolves with a `USVString` containing the entire contents of the `Blob` interpreted as UTF-8 text.

Examples

Creating a blob

Creating a blob

The `Blob()` constructor can create blobs from other objects. For example, to construct a blob from a JSON string:

```
const obj = {hello: 'world'};
const blob = new Blob([JSON.stringify(obj, null, 2)], {type : 'application/json'});
```

Creating a URL representing the contents of a typed array

The following code creates a JavaScript typed array and creates a new `Blob` containing the typed array's data. It then calls `URL.createObjectURL()` to convert the blob into a URL.

HTML

```
<p>This example creates a typed array containing the ASCII codes
   for the space character through the letter Z, then converts it
   to an object URL. A link to open that object URL is created.
   Click the link to see the decoded object URL.</p>
```

JavaScript

The main piece of this code for example purposes is the `typedArrayToURL()` function, which creates a `Blob` from the given typed array and returns an object URL for it. Having converted the data into an object URL, it can be used in a number of ways, including as the value of the `<img>` element's `src` attribute (assuming the data contains an image, of course).

```
function typedArrayToURL(typedArray, mimeType) {
  return URL.createObjectURL(new Blob([typedArray.buffer], {type: mimeType}))
}
```

```
const bytes = new Uint8Array(59);

for(let i = 0; i < 59; i++) {
  bytes[i] = 32 + i;
}


const url = typedArrayToURL(bytes, 'text/plain');

const link = document.createElement('a');
link.href = url;
link.innerText = 'Open the array URL';

document.body.appendChild(link);
```

Result

Click the link in the example to see the browser decode the object URL.

This example creates a typed array containing the ASCII codes for the space character through the letter Z, then converts it to an object URL. A link to open that object URL is created. Click the link to see the decoded object URL.

Open the array URL

Extracting data from a blob

One way to read content from a `Blob` is to use a `FileReader`. The following code reads the content of a

`Blob` as a typed array:

```
const reader = new FileReader();
reader.addEventListener('loadend', () => {
   // reader.result contains the contents of blob as a typed array
});
reader.readAsArrayBuffer(blob);
```

Another way to read content from a `Blob` is to use a `Response`. The following code reads the content of a `Blob` as text:

```
const text = await (new Response(blob)).text();
```

Or by using `Blob.prototype.text()`:

```
const text = await blob.text();
```

By using other methods of `FileReader`, it is possible to read the contents of a Blob as a string or a data URL.

Specifications

Specification	Status	Comment
File API The definition of 'The `Blob` interface' in that specification.	Working Draft	Initial definition.

Browser compatibility

Browser compatibility

Report problems with this compatibility data on GitHub

`Blob`	
Chrome	5
Edge	12
Firefox	4
Internet Explorer	10
Opera	11
Safari	5.1
WebView Android	≤ 37
Chrome Android	18
Firefox Android	14
Opera Android	11
Safari on iOS	6
Samsung Internet	1.0
`Blob()` constructor	
Chrome	20
Edge	12
Firefox	13
Internet Explorer	10

Internet Explorer	10
Opera	12
Safari	8
WebView Android	37
Chrome Android	25
Firefox Android	14
Opera Android	12
Safari on iOS	8
Samsung Internet	1.5
`arrayBuffer()`	
Chrome	76
Edge	79
Firefox	69
Internet Explorer	**No**
Opera	**No**
Safari	14
WebView Android	76
Chrome Android	76
Firefox Android	**No**
Opera Android	54

Opera Android	54
Safari on iOS	14
Samsung Internet	12.0

`size`	
Chrome	5
Edge	12
Firefox	4
Internet Explorer	10
Opera	11
Safari	5.1
WebView Android	≤ 37
Chrome Android	18
Firefox Android	**No**
Opera Android	**No**
Safari on iOS	**No**
Samsung Internet	1.0

`slice()`	
Chrome	21
Edge	12
Firefox	13

Internet Explorer	10
Opera	12
Safari	5.1 - x-
WebView Android	≤ 37
Chrome Android	25
Firefox Android	14
Opera Android	Yes
Safari on iOS	Yes
Samsung Internet	1.5
`stream()`	
Chrome	76
Edge	79
Firefox	69
Internet Explorer	**No**
Opera	**No**
Safari	**No**
WebView Android	76
Chrome Android	76

Firefox Android	**No**
Opera Android	54
Safari on iOS	**No**
Samsung Internet	12.0
`text()`	
Chrome	76
Edge	79
Firefox	69
Internet Explorer	**No**
Opera	**No**
Safari	**No**
WebView Android	76
Chrome Android	76
Firefox Android	**No**
Opera Android	54
Safari on iOS	**No**
Samsung Internet	12.0
`type`	
Chrome	5
Edge	12

Firefox	4
Internet Explorer	10
Opera	11
Safari	5.1
WebView Android	≤ 37
Chrome Android	18
Firefox Android	**No**
Opera Android	**No**
Safari on iOS	**No**
Samsung Internet	1.0

Full support

No support

See implementation notes.

-x- Requires a vendor prefix or different name for use.

See also

- `BlobBuilder`
- `FileReader`
- `File`
- `URL.createObjectURL`
- Using files from web applications

Last modified: Feb 19, 2021, by MDN contributors