Request: textStream() method

- Syntax
  - Parameters
  - Return value
  - Exceptions
- Examples
  - Reading request body content as a text stream
- Specifications
- Browser compatibility
- See also

{{APIRef("Fetch API")}}{{AvailableInWorkers}}{{SeeCompatTable}}

The **`textStream()`** method of the {{domxref("Request")}} interface returns a {{domxref("ReadableStream")}} that can be used to read the contents of the request body in chunks of UTF-8.

This provides an easier mechanism for streaming the request body than piping the {{domxref("Request.body")}} byte stream through a {{domxref("TextDecoderStream")}}.

> [!NOTE] If invoked on a `Request` with a null body, for example a {{httpmethod("GET")}} request, `textStream()` will return a valid empty stream.

Syntax

```
textStream()
```

Parameters

None.

Return value

A {{domxref("ReadableStream")}}.

Exceptions

- {{jsxref("TypeError")}}
  - : Thrown if the request body is disturbed or locked.

Examples

Reading request body content as a text stream

This example shows how to read a request body as a text stream.

We create a sample `Request`, obtain a `ReadableStream` of its body using `textStream()`, then read the text via a reader created using {{domxref("ReadableStream.getReader()")}}.

```
const pElem = document.querySelector("p");

const req = new Request("https://example.com", {
  method: "POST",
  body: '{"hello": "world"}',
});

async function streamRequestText(request) {
  const textStream = request.textStream();
  // instead of
  // const textStream = request.body.pipeThrough(new TextDecoderStream());
```

```
  const reader = textStream.getReader();

  while (true) {
    const { value, done } = await reader.read();
    if (done) break;
    pElem.textContent += value;
  }
}

streamRequestText(req);
```

Specifications

{{Specifications}}

Browser compatibility

{{Compat}}

See also

- {{domxref("ReadableStream")}}
- {{domxref("ReadableStream.getReader()")}}
- {{domxref("Response.body")}}
- {{domxref("Response.textStream()")}}

□□□ #2
□ □□□ □□□ 2026-10-31 03:27:53 UTC
□ □□□ □□□ 2026-10-31 03:27:53 UTC