

Ta emot data

- Data skickas vanligen med HTTP-metoden POST
 - Däremot finns det flera andra metoder som används (vi kommer dit framöver)
- Request-objektet har metoden `Request.text()`
 - Detta "konverterar" förfrågans *body* till text (dvs. en sträng)

```
async function handler(request) {  
  if (request.method == "POST") {  
    const data = await request.text();  
    console.log(`Received: ${data}`);  
  }  
  
  return new Response("Hello!");  
}  
  
Deno.serve(handler);
```

async / await är två nyckelord ni får leva med, vi kommer att gå igenom dem framöver

Att skicka data (Firefox)

Detta kommer vi göra via JavaScript i framtiden

The screenshot shows the Firefox Developer Tools interface with the Network tab selected. The browser window displays "Hello!". The Network tab shows a list of requests, with the first one being a POST request to "http://localhost:8000/". The left sidebar of the Network tab is expanded, showing the "New Request" form. The form has sections for "URL Parameters", "Headers", and "Body". The "Body" section contains the text "Här kan vi skriva in den data vi vill ange". At the bottom of the form are "Clear" and "Send" buttons. A red arrow points to the "Send" button. On the left side of the image, three red arrows point to the "POST" method dropdown, the "URL Parameters" section, and the "Body" section of the "New Request" form.

localhost:8000/

localhost:8000

150%

Hello!

Inspector Console Debugger Network Style Editor Performance Memory Storage

Filter URLs

Disable Cache No Throttling

Status	Method	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:8000	/	document	plain	144 B	6 B
200	GET	localhost:8000	favicon.ico	FaviconLoader.sys...	plain	144 B	6 B

New Request Search Blocking

POST http://localhost:8000/

URL Parameters

name value

Headers

name value

Body

Här kan vi skriva in den data vi vill ange

Clear Send

2 requests 12 B / 288 B transferred Finish: 18 ms DOMContentLoaded: 5 ms load: 7 ms