

Skicka JSON

- **Metod**
 - POST, PUT, DELETE, etc.
- **URL**
 - `http://localhost:8000/`, `http://localhost:8000/dogs`, etc.
- **Innehållstyp**
 - `JSON` = Content-Type: `application/json`
- **Datan (dvs. *request body*)**
 - `"Some piece of text"`
 - `50241`
 - `[1, 4, 6, 44, 1]`
 - `{ "title": "Hackers", "year": 1995 }` *Obs! nycklar måste ha citationstecken*

Inspector Console Debugger Network Style Editor

Filter URLs

All HTML CSS JS XHR Fonts Images Media WS Other

New Request Search Blocking

Metod POST http://localhost:8000/dogs URL

URL Parameters

☒ name value

Headers

Innehållstyp ☒ Content-Type application/json

☒ name value

Body

Request body { "name": "Arya", "age": 4, "breed": "Husky" }

Clear Send

Ta emot JSON

Exempeldata: { "title": "Hackers", "year": 1995 }

```
async function handler(request) {  
  if (request.method == "POST") {  
    const movie = await request.json();  
    console.log(movie.title, movie.year);  
  }  
  return new Response("Hello!");  
}  
  
Deno.serve(handler);
```

Bonus: servern kan svara med JSON

```
// ...
```

```
if (request.method == "POST") {  
  const movie = await request.json();  
  addMovieToDatabase(movie);  
  const answer = { message: `The movie ${movie.title} was added` };  
  const data = JSON.stringify(answer);  
  return new Response(data, {  
    headers: { "Content-Type": "application/json" }  
  });  
}
```

```
// ...
```

Bonus: servern kan svara med JSON

```
async function handler(request) {  
  if (request.method == "POST") {  
    // ...  
  }  
  
  const dogs = [ ... t.ex. från en fil ... ];  
  const data = JSON.stringify(dogs);  
  return new Response(data, {  
    headers: { "Content-Type": "application/json" }  
  });  
}  
  
Deno.serve(handler);
```