

Serverupplägg

CORS vs Same Origin

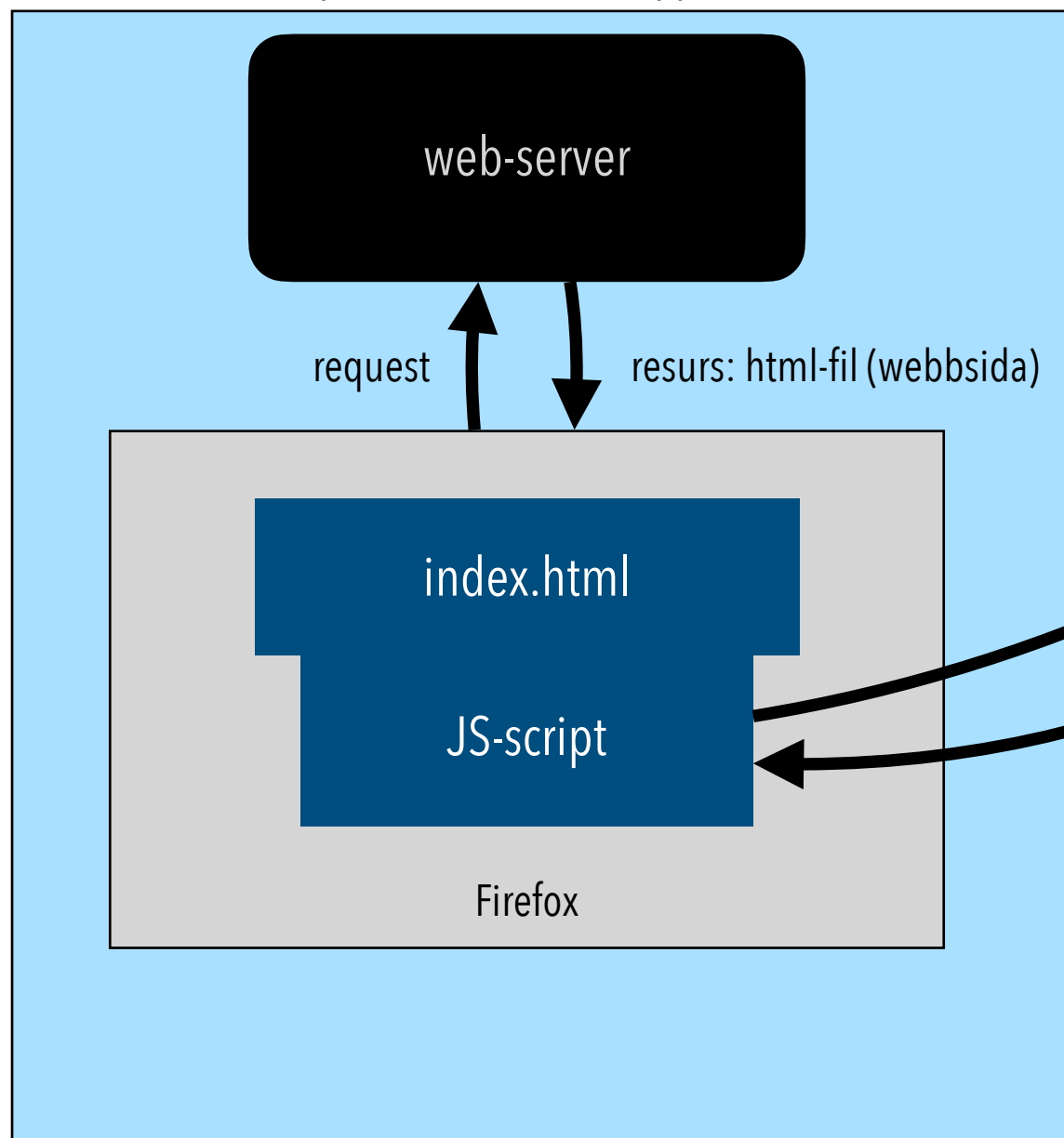
Cross-Origin: Olika servrar

Denna server levererar index.html.

Den levererar även filen index.js, som innehåller en JS-script (kod) som laddas upp i index.html

Denna server tar emot requests på sökvägen /random.

Resursen den levererar är en slumpmässig siffra mellan 1 och 100



request

resurs: slumpmässig siffra

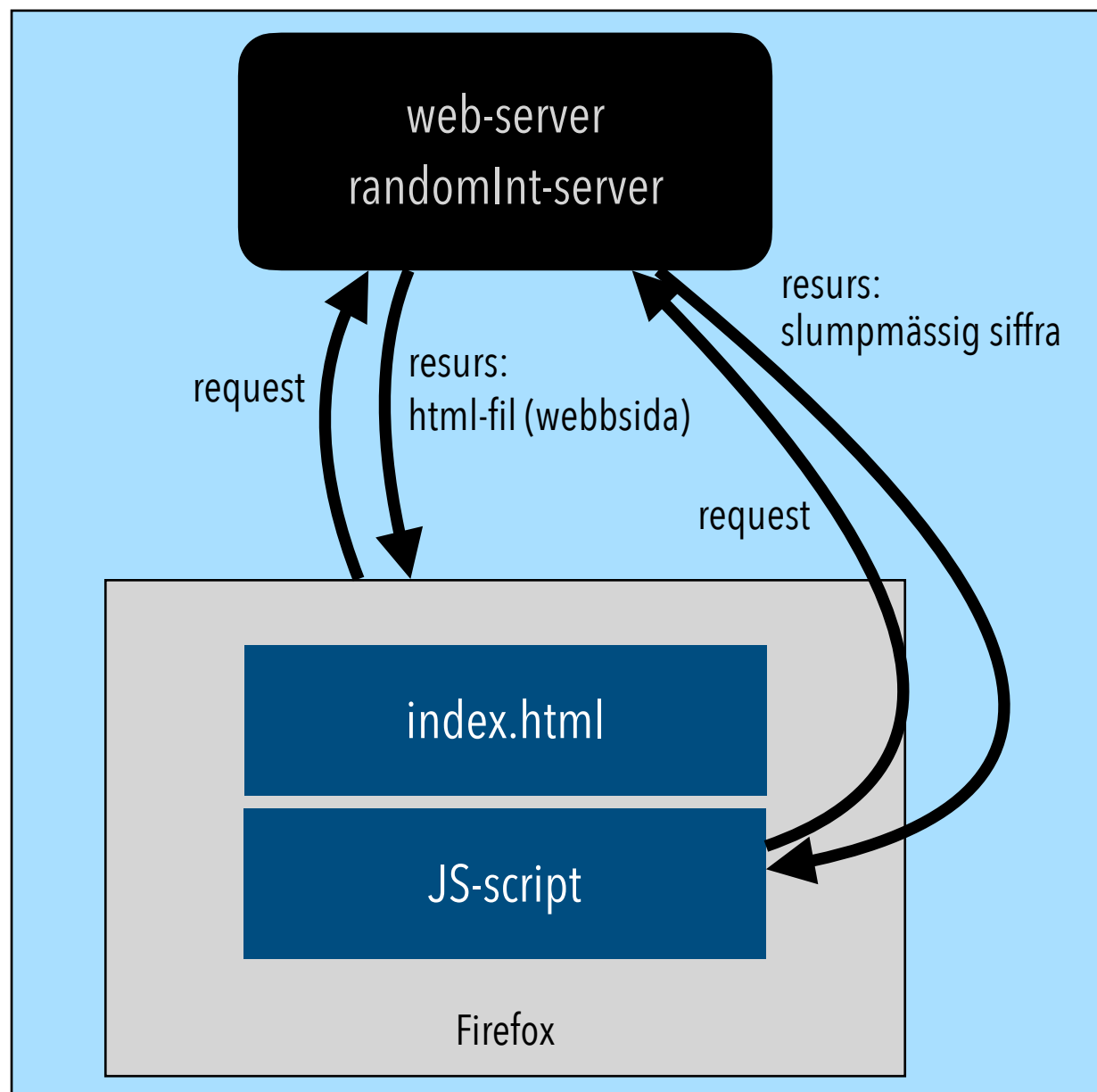
Cross-Origin Resource Sharing: klienten som skickar förfrågan (som vill ha resursen) finns på en annan server.

Notera att när du öppnar en html-fil direkt från webbläsaren (genom att släppa den i webbläsaren, tex), så agerar webbläsaren som en sorts web-server. Då hamnar du alltid i detta upplägg.

Same-Origin: Samma server

Denna server levererar index.html och index.js.

Den tar även emot requests på sökvägen /random, och levererar en slumpmässig siffra mellan 1 och 100



Same Origin: klienten som skickar förfrågan finns på samma server.

Cross-Origin: randomInt-server

Svarar endast på GET-requests till sökvägen /random

Resursen den levererar ("serves") är en random siffra mellan 1 och 100

```
function handler(request) {  
  const url = new URL(request.url);
```

```
  // =====  
  // Hantering av CORS  
  const headersCORS = new Headers();  
  headersCORS.set("Access-Control-Allow-Origin", "*"); // Allow all origins  
  if (request.method === "OPTIONS") {  
    return new Response(null, { headers: headersCORS });  
  }  
  // =====
```

Dessa instruktioner skapar headers som gör att responsen från denna server blir tillgänglig för "externa" scripts.

```
  if (url.pathname === "/random") {  
    if (request.method === "GET") {  
      const randomInt = 1 + Math.floor(100 * Math.random()); // mellan 1 och 100  
      return new Response (randomInt, { headers: headersCORS });  
    }  
  }  
  
  return new Response("Invalid Request", {headers: headersCORS, status: 400});  
}
```

```
Deno.serve({ port: 4242 }, handler); // Endast en server per port i vår dator!
```

Cross-Origin: web-server

Server (levererar) index.html och filerna som den länkar till (som index.js)

```
import { serveFile, serveDir } from "jsr:@std/http";
function handler(request) {
  const url = new URL(request.url);

  if (url.pathname == "/") {
    return serveFile(request, "index.html");
  }
```

```
  if (url.pathname == "/favicon.ico") {
    return serveFile(request, "public/favicon.png");
  }
```

Bara jag som är så trött på meddelandet om att favicon.ico inte har hittats?

```
  const opts = { fsRoot: "public", urlRoot: "static" };
  return serveDir(request, opts);
}
```

```
Deno.serve(handler); // 8000 är default port för en Deno server
```

Same Origin: Endast en server

Den levererar både filerna och randomInt

```
import { serveFile, serveDir } from "jsr:@std/http";
```

```
function handler(request) {  
  const url = new URL(request.url);
```

```
  if (url.pathname === "/random") {  
    if (request.method === "GET") {  
      const randomInt = 1 + Math.floor(100 * Math.random());  
      return new Response (randomInt);  
    }  
    return new Response("Bad Method", {status: 400});  
  }  
}
```

Denna del tar hand om /random

```
  if (url.pathname === "/") { return serveFile(request, "index.html"); }  
  if (url.pathname === "/favicon.ico") { return serveFile(request, "public/favicon.png"); }
```

```
  const opts = { fsRoot: "public", urlRoot: "static" };  
  return serveDir(request, opts);
```

Denna del tar hand om filerna

```
}
```

```
Deno.serve(handler);
```

Vi kommer att använda en extern server (CORS)

Det är mycket vanligt att man har det upplägget: olika servrar som levererar olika saker.

Externa APIs (som andra tillhandahåller) är alltid... externa.