

Routing

- *Pretty URLs*
- `http://kennel.se/hundar`
 - Visa alla hundar
- `http://kennel.se/hundar/42`
 - Visa en hund (ID = 42)
- `http://dogstagram.com/users/19032/posts`
 - Visa en användares inlägg (ID = 19032)
- `http://dogstagram.com/users/5928/posts?year=2024`
 - Kombineras ofta med parametrar

Routing

- Radera en hund baserat på ett ID i URL:en (istället för att skicka med det)
 - DELETE /dogs/42
- Redigera en hund, t.ex. åldern
 - PATCH /dogs/57
{ age: 5 }

URLPattern

- URLPattern-objektet ger oss möjlighet att:
 - kunna ange en sökväg inklusive t.ex. "id"
 - kunna "testa" ifall en URL matchar det vi angivit
 - kunna ta ut värdet från URL:en, t.ex. id:et
- `const route = new URLPattern({ pathname: "/users/:id" });`
 - Observera att vi skriver `":id"` för att ange någonting
 - Det kan finnas flera i samma sökväg (med olika namn förstås)
- `const match = route.exec("http://site.com/users/15");`
 - `match` är `null` ifall det **inte** matchade
 - Därför kan vi t.ex. skriva `if (match) { ... }`
- `console.log(match.pathname.groups.id); // "15"`

```
const dogsRoute = new URLPattern({ pathname: "/dogs/:id" });

function handler(request) {
  const match = dogsRoute.exec(request.url);

  if (match) {
    const id = match.pathname.groups.id;
    return new Response(`You requested dog ${id}`);
  }

  return new Response("Dog Not Found", { status: 404 });
}

Deno.serve(handler);
```