

There and back again

HTTP APIs

Follow your nose

GET /entry

Follow your nose

200 OK

{

“furnishings”: “/furnishings”,

“appliances”: “/b16f18d2-7636-4a4a-9820-ad9179f68c52”,

“tableware”: “/tableware”,

“menswear”: “/menswear”,

...

Follow your nose

GET /tableware

Follow your nose

200 OK

{

“plates”: “/crockery/keep-the-kids-away”,

“cutlery”: “/silverware”,

“glassware”: “/crockery/no-we-mean-it-this-time”,

...

Follow your nose

GET /silverware

Follow your nose

200 OK

```
{  
  "sets": "/silverware/boxes",  
  "teaspoons": "/silverware/teaspoons",  
  "serving": "/silverware/serving",  
  "knives": "/kitchenware/knives",  
  ...  
}
```

Follow your nose

GET /me/a/knife/because/I'm/about/to/stab/someone

Search

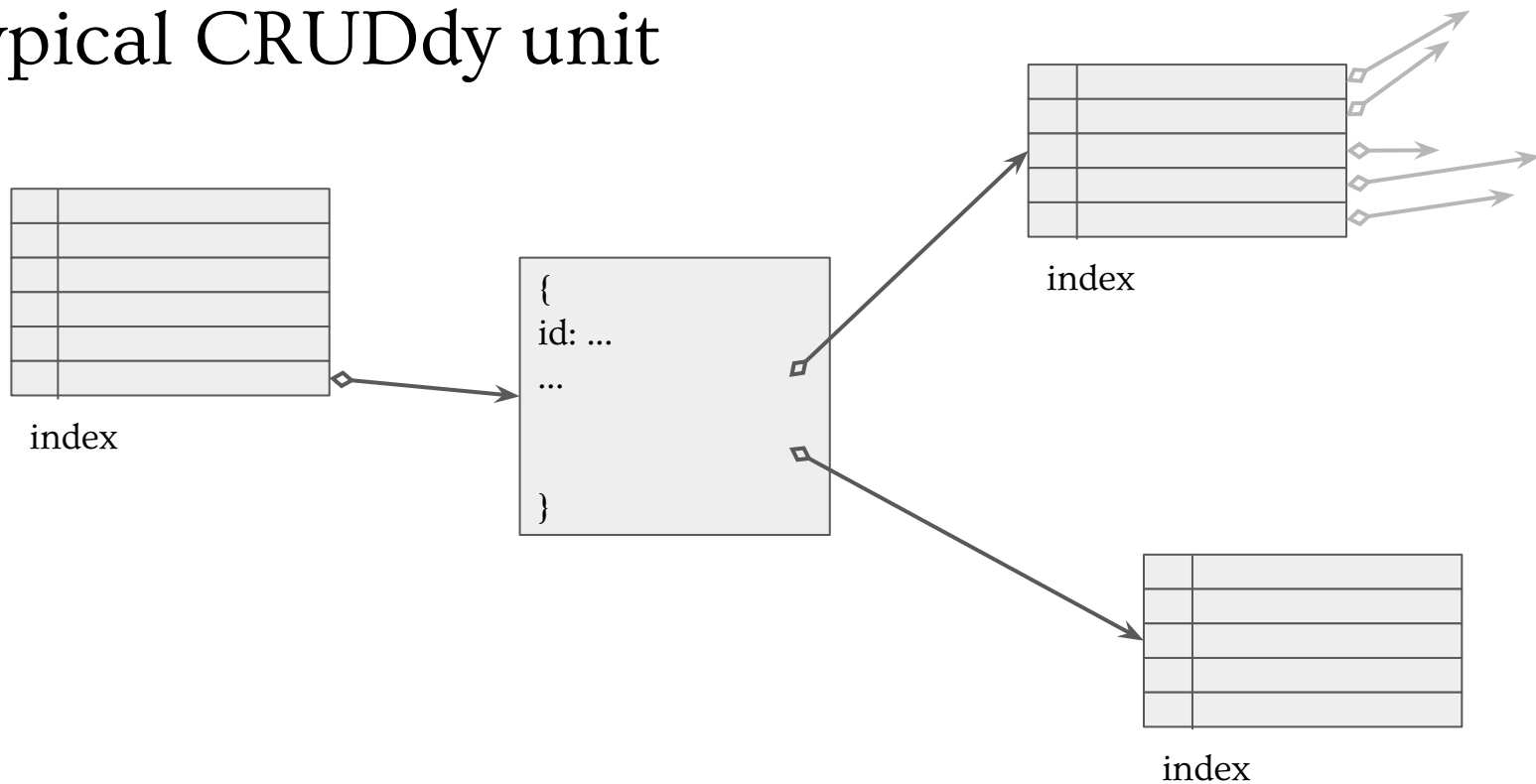
Obviously, this is a contrived example

A search function is probably better for this

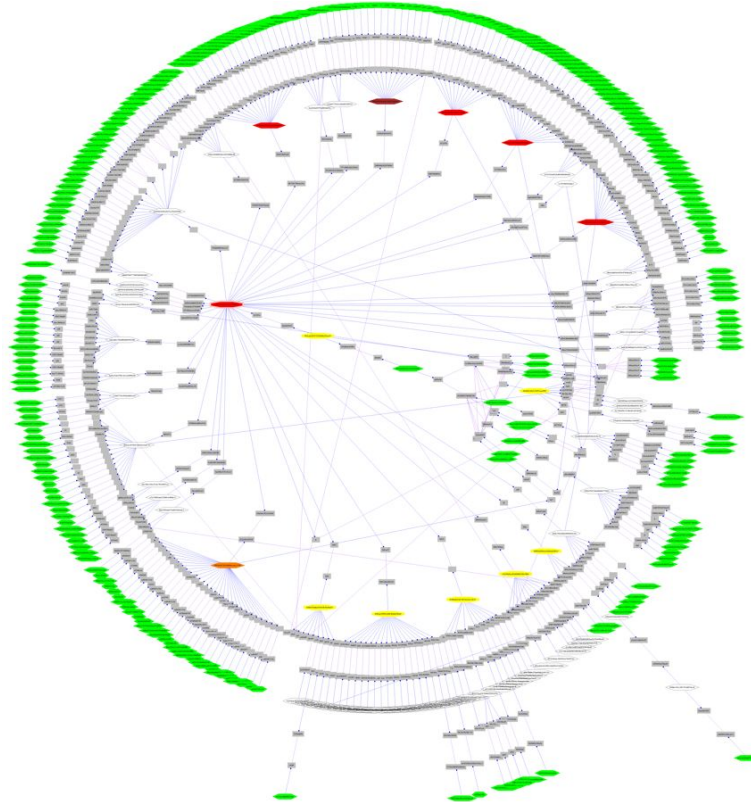
But a lot of HTTP APIs are built this way

And they aren't very fast

Typical CRUDdy unit



Typical (tiny) CRUDdy application



What if

... you could talk about a response

What if

... you could talk about a response

... at the same time you make the request?

Parallel dependent requests

GET /object
[[id=foo]]

200 OK

{ "other": "/some/url/...", ... }

GET <foo[other]>

200 OK

...what you really wanted...

hx URI

Identifies an exchange:

hx://b5dd5901aef3f33de572/7

A request or response:

hx://b5dd5901aef3f33de572/7/q

Or pieces of the same:

hx://b5dd5901aef3f33de572/7/a/h/location

hxr URI

As with hx, but is dereferenced twice

hxr:///O/a/h/location?201

Causes the URI from the Location header field to be used if the status was 201

Messages or representations?

Build mechanically around HTTP messages

Might be better to talk about representations

- Enable use of redirects

- ... including 304

- ... can suppress intermediate results (HEAD?)

Drawbacks

These expressions are complicated

Drawbacks

These expressions are complicated

Even simple use cases are hard to express

Might not be expressive enough for real use cases

Reverse JavaScript

Clients send servers a bit of script instead

That script could make requests to gather information

... and use that information to make more requests

Like a server-side service worker