



UNIVERSITY OF CALIFORNIA, IRVINE

Architectural Styles and the Design of Network-based Software Architectures

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Information and Computer Science

by

Roy Thomas Fielding

2000

"When given a name, a coordinated set of architectural constraints

becomes an

architectural Style"



"REST is software design on

the scale of decades:

every detail is intended to promote software longevity and independent evolution.

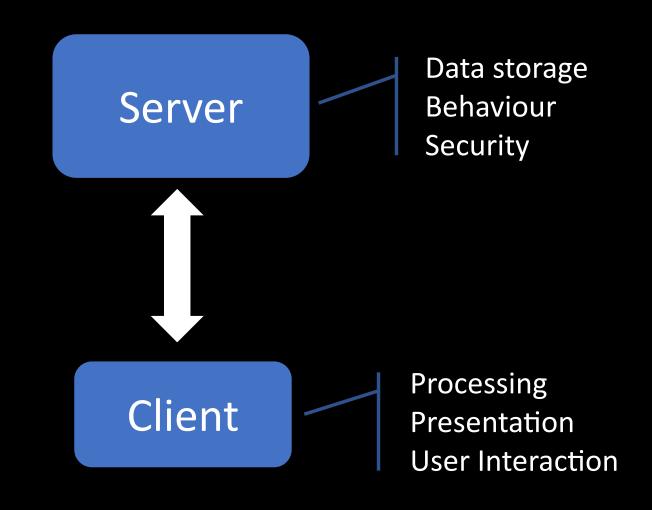


Representational State Transfer (REST)

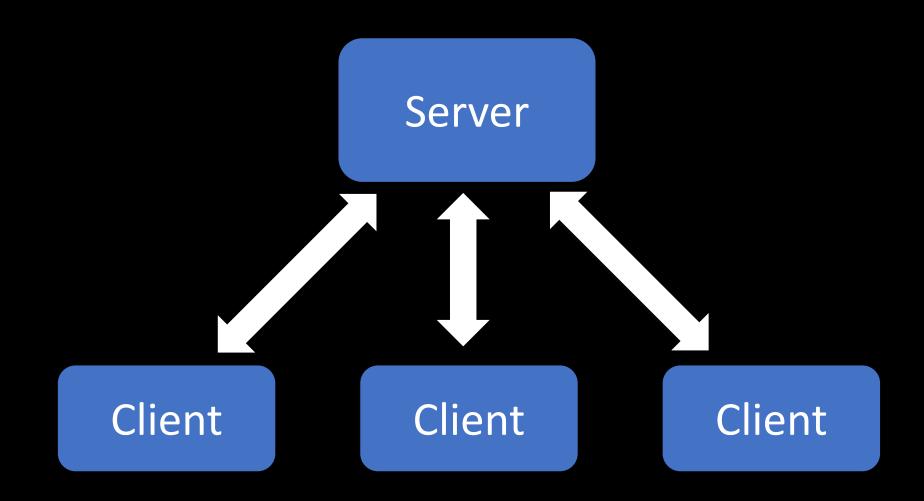
A coordinated set of architectural constraints...

...intended to promote software longevity and independent evolution

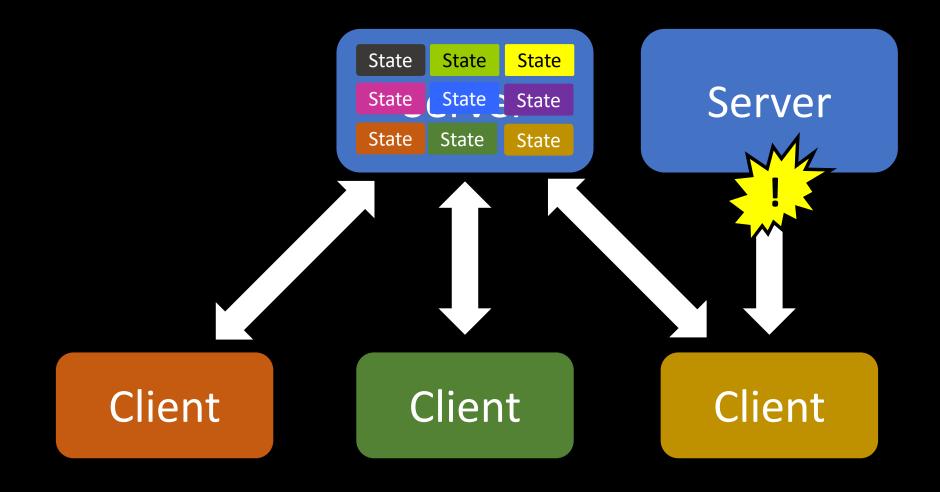
1. Client-Server



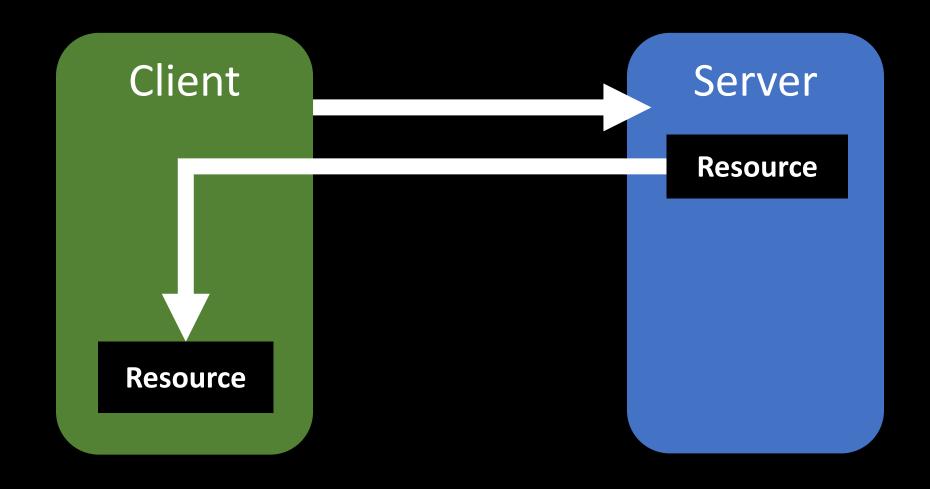
1. Client-Server



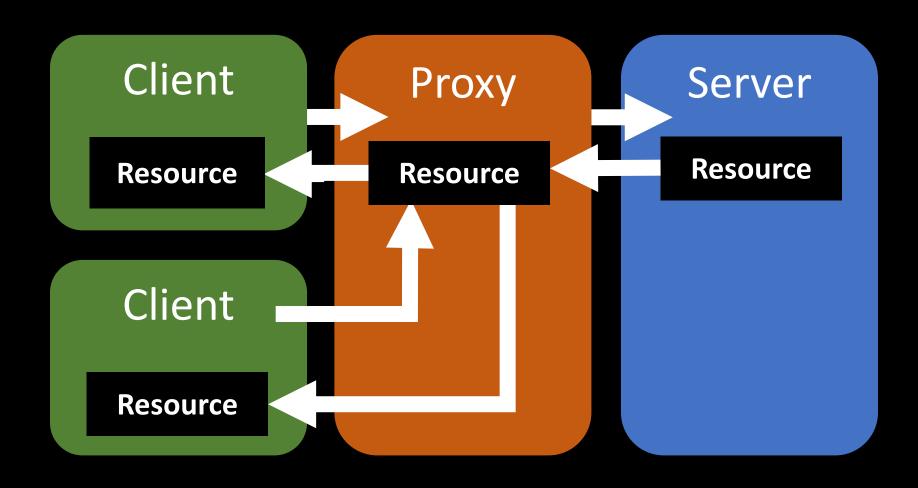
2. Stateless



3. Cacheable



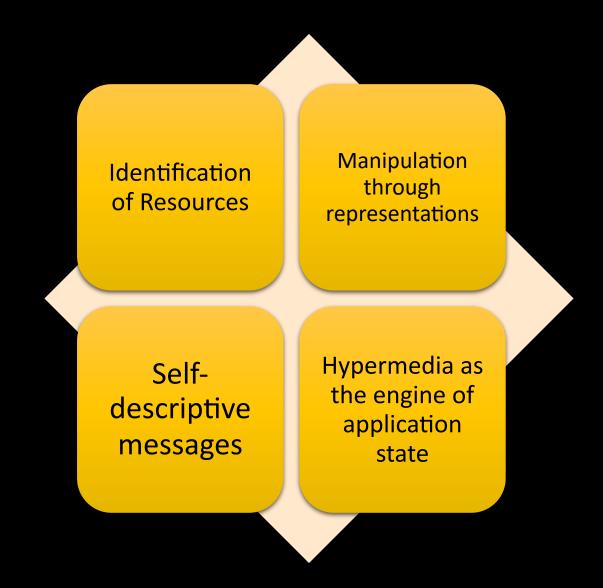
4. Layered System



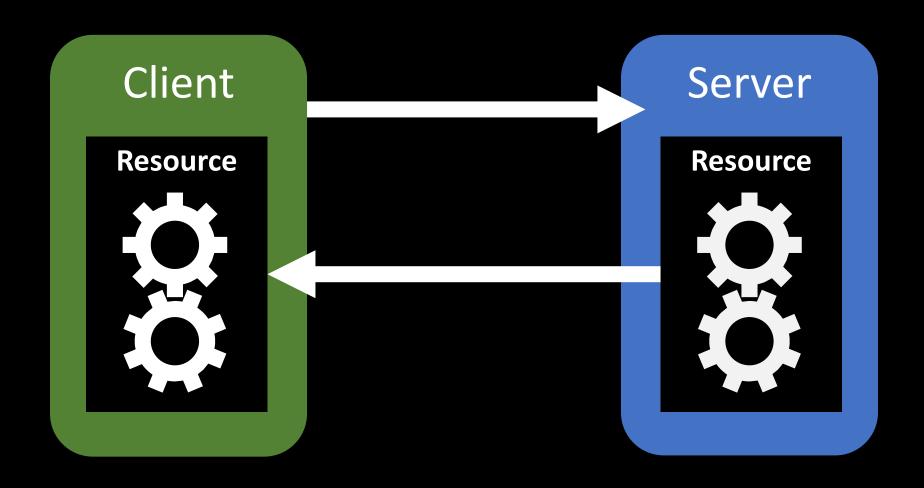
4. Layered System (continued)

Client Server Security Appliance Caching Proxy Client

5. Uniform Interface



6. Code on Demand



A Set of Architectural Constraints

- 1. Client-server
- 2. Stateless
- 3. Cacheable
- 4. Layered System
- 5. Uniformed Interface

6. Code-on-demand

OPTIONAL



GET / HTTP/1.1

```
200 OK
Content-Type: application/json
{
   "message": "Welcome to Herobook!"
}
```

GET /profiles HTTP/1.1

```
200 OK
Content-Type: application/json
  "username": "ironman",
  "name": "Tony Stark",
```

```
POST /profiles HTTP/1.1
 "username": "blackwidow"
"name": "Наташа Романова",
201 Created
Location: http://api.herobook.local/profiles/blackwidow
 "username": "blackwidow",
 "name": "Наташа Романова"
```



https://readthedocs.org/herobook/

HEROBOOK API DOCUMENTATION

GET /profiles

POST /profiles

```
GET /profiles HTTP/1.1
200 OK
"_links" : { ... },
 "items" : {
  { "username":"ironman", "name":"Tony Stark" }
  { "username":"blackwidow", "name":"Наташа Романова" },
  { "username": "spidey", "name": "Peter Parker" },
  { /* +2 million users! Wow! */ },
  { "username":"ducky", "name":"Howard the Duck" }
```

"Your API is using all our bandwidth!"

"Your stupid API is overloading our database!"

GET /profiles?page=1 HTTP/1.1

```
200 OK
 "_links" : { ... },
 "items" : [
  { "username": "ironman", "name": "Tony Stark" }
  { "username": "spidey", "name": "Peter Parker" },
  { "username": "blackwidow", "name": "Наташа Романова" },
  { "username": "cap", "name": "Steve Rogers" },
  { "username": "storm", "name": "Ororo Munroe" }
```

GET /profiles?page=2 HTTP/1.1
200 OK
GET /profiles?page=3 HTTP/1.1
200 OK

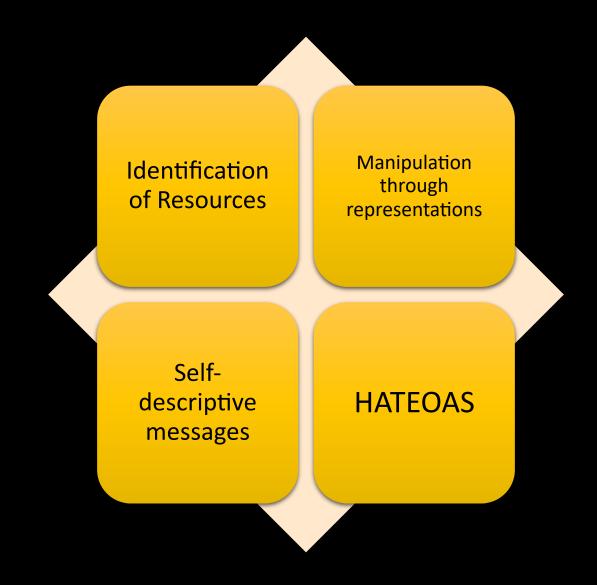
GET /profiles?page=1 H 200 OK

GET /profiles?page= HTTP/1.1

204 No Content



5. Uniform Interface



CHOOSE YOUR OWN ADVENTURE * 6

YOU'RE THE HERO OF THE STORY! CHOOSE FROM 40 POSSIBLE ENDINGS.

YOUR CODE NAME IS JONAH



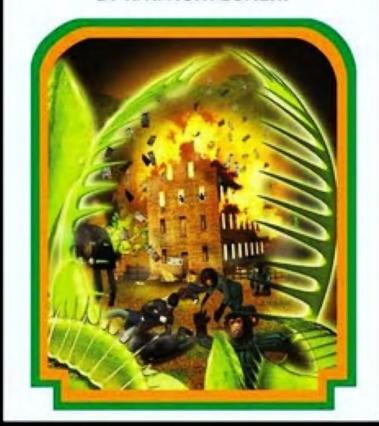
ILLUSTRATED BY PAUL GRANGER

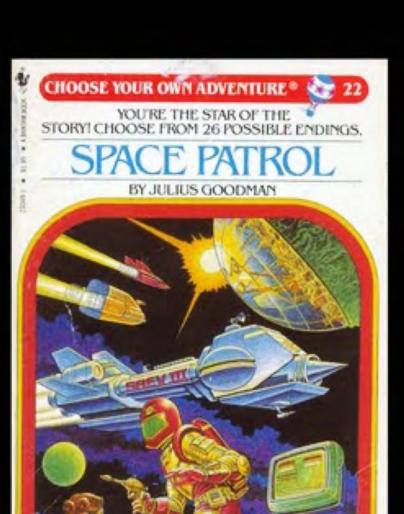
CHOOSE YOUR OWN ADVENTURE® 6

THE CLASSIC SERIES IS BACK! CHOOSE FROM 20 POSSIBLE ENDINGS.

HOUSE OF DANGER

BY R. A. MONTGOMERY







YOU'RE THE STAR OF THE STORY! CHOOSE FROM 21 POSSIBLE ENDINGS!

HANDS-ON HYPERMEDIA

DevTernity Riga 2018



You're at DevTernity and it's AWESOME! You've seen some great talks, met some great people. Now you're watching an amazing talk about REST. After the coffee break, you have a choice — Sven Peters is talking about successful teams, Jakub Nabrdalik is talking about how to make your tests rock, or Marco Heimeshoff is talking about the pillars of domain driven design... so what are you going to do?

To watch Sven's talk about teams, turn to page 23

To go to Jakub's testing talk, turn to page 41

To go to Marco's talk about DDD, turn to page 52

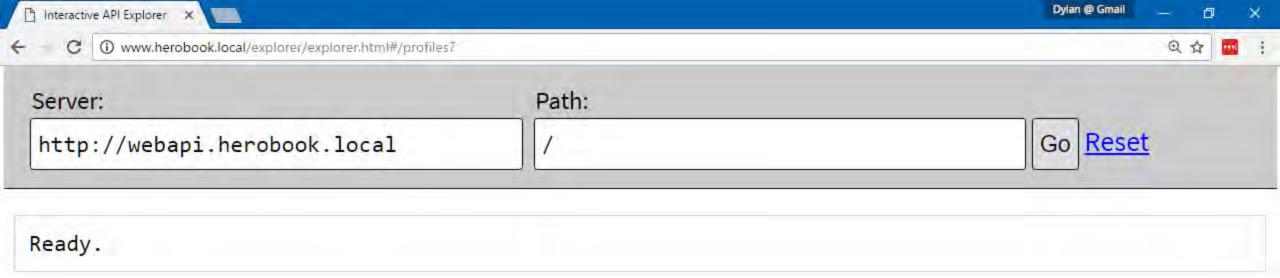
```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
 "_links" : {
 "items:" [
  { "username":"ironman", "name":"Tony Stark" }
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  { "username": "pepper", "name": "Pepper Potts" },
  { "username": "storm", "name": "Ororo Munroe" }
```

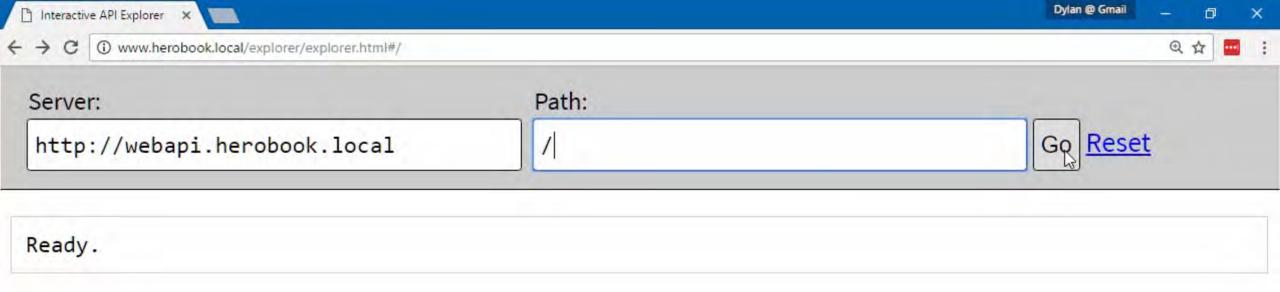
```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
 " links" : {
  "self" : { "href" : "http://herobook/profiles?index=0" },
  "next": { "href": "http://herobook/profiles?index=5" },
  "last" : { "href" : "http://herobook/profiles?index=220" }
 "items:" [
  { "username":"ironman", "name":"Tony Stark" }
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  { "username": "pepper", "name": "Pepper Potts" },
  { "username": "storm", "name": "Ororo Munroe" }
```

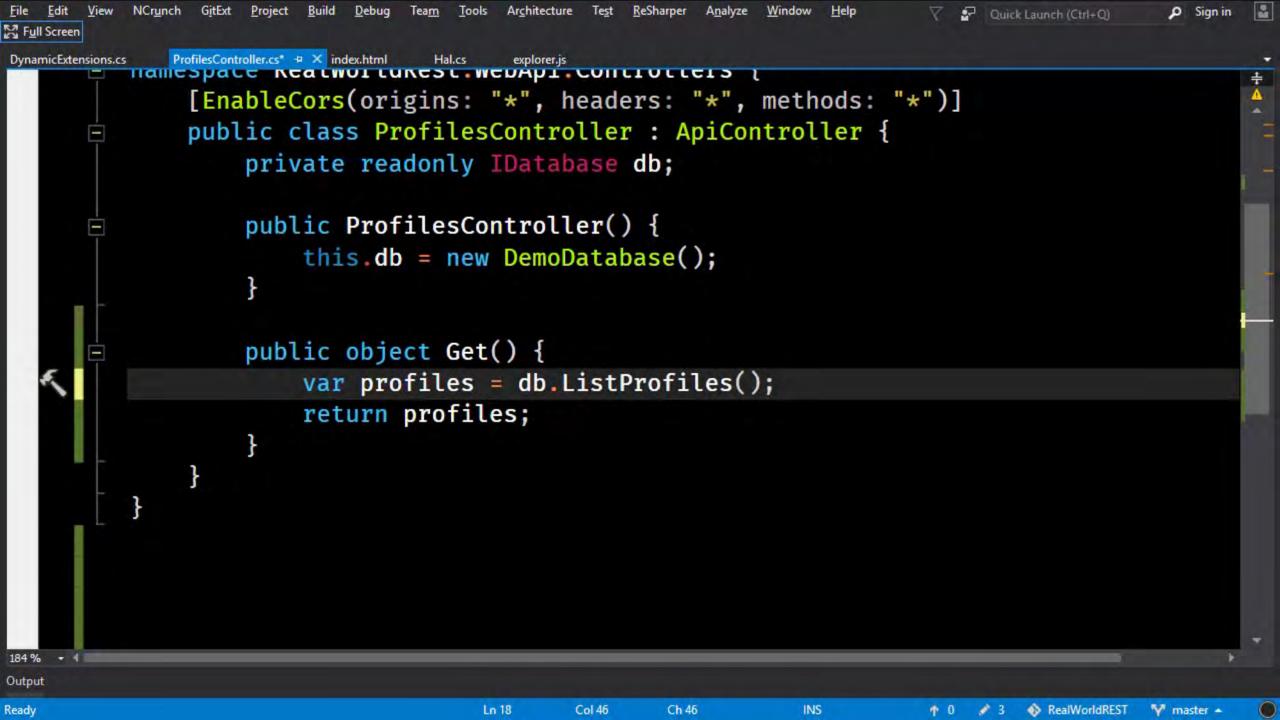
```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
 " links" : {
  "self" : { "href" : "http://herobook/profiles?index=0" },
  "next": { "href": "http://herobook/profiles?index=5" },
  "last" : { "href" : "http://herobook/profiles?index=220" }
 "count": 5,
 "index" : 0,
 "total": 223,
 "items:" [
  { "username":"ironman", "name":"Tony Stark" }
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  { "username": "pepper", "name": "Pepper Potts" },
  { "username": "storm", "name": "Ororo Munroe" }
```

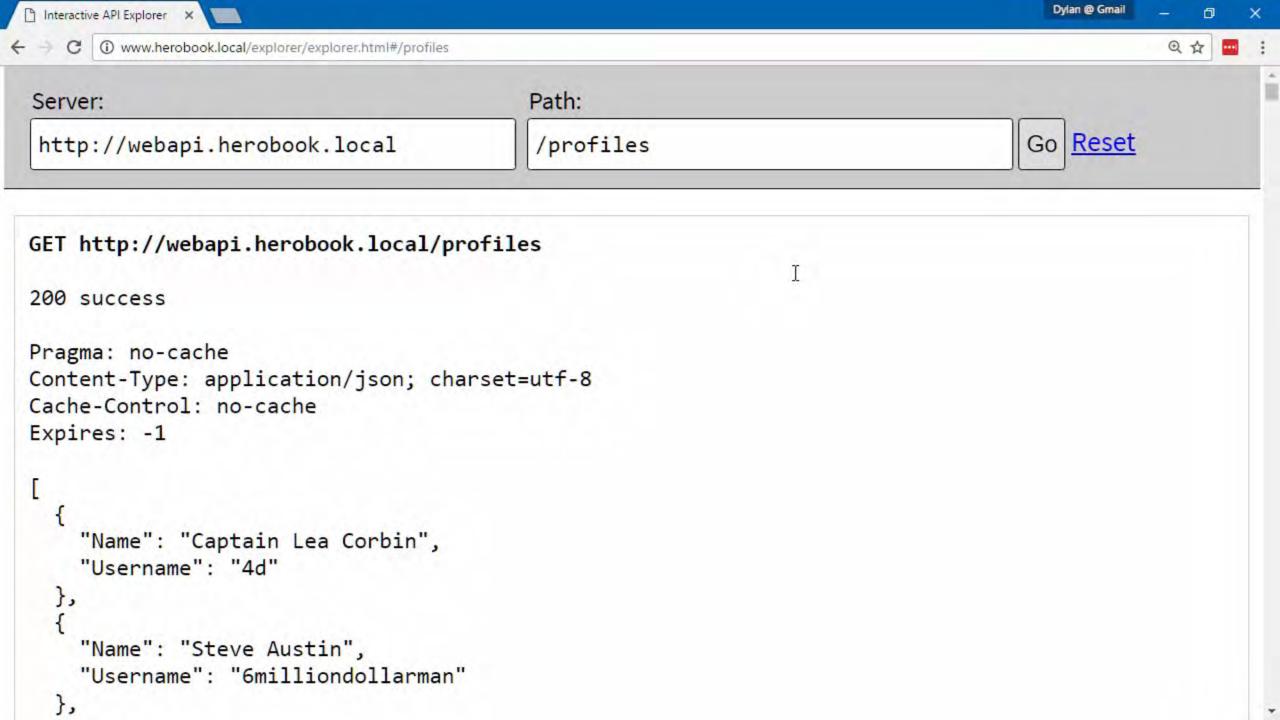
Enough talk already... SHOW ME THE

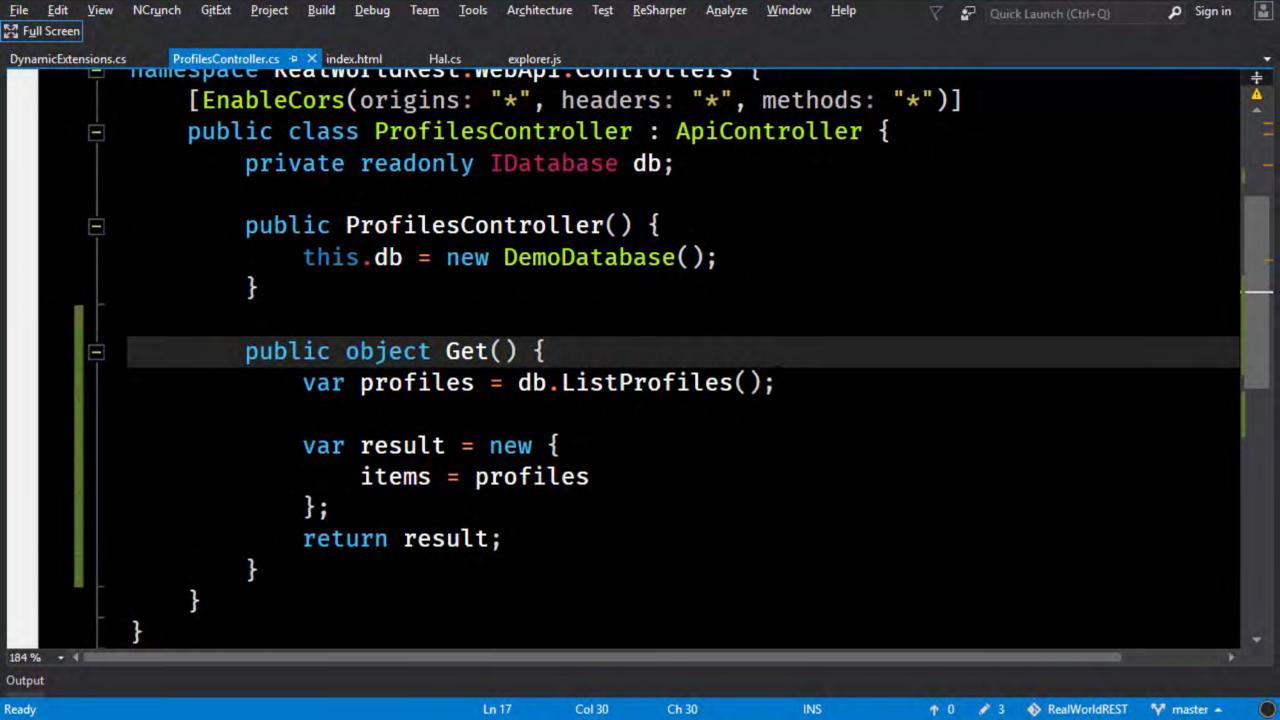
https://github.com/tonsky/FiraCode

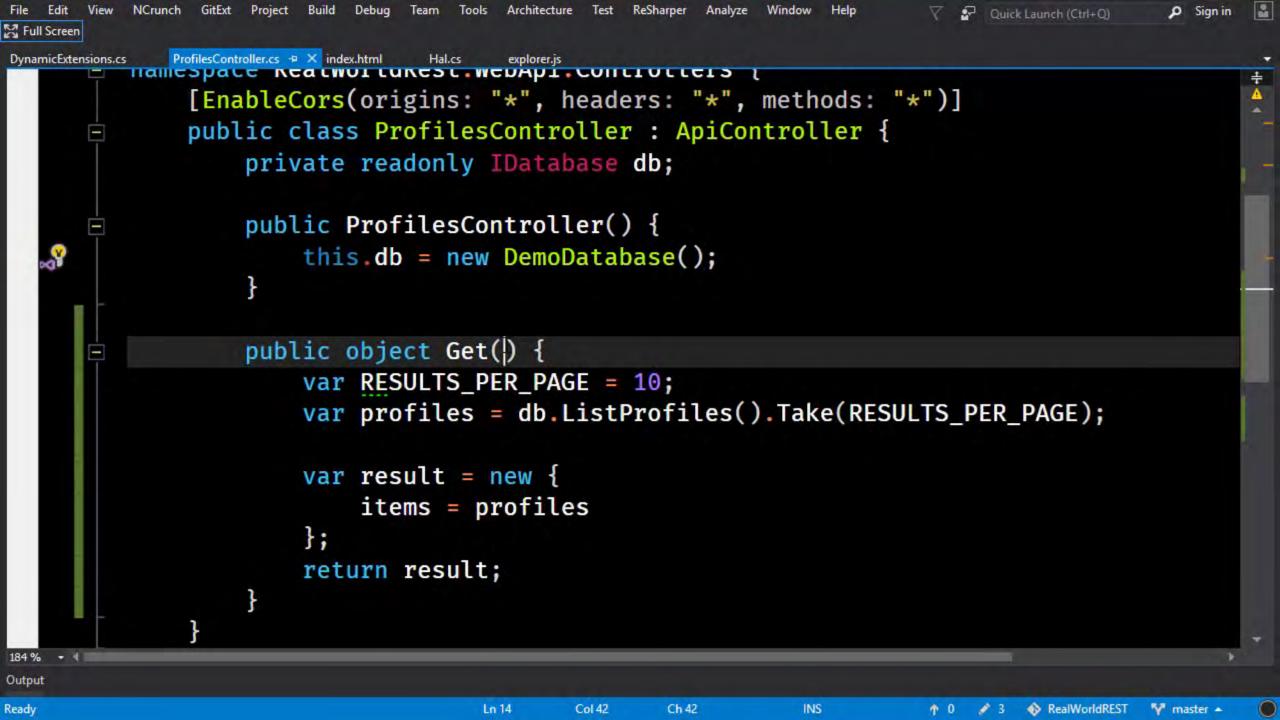


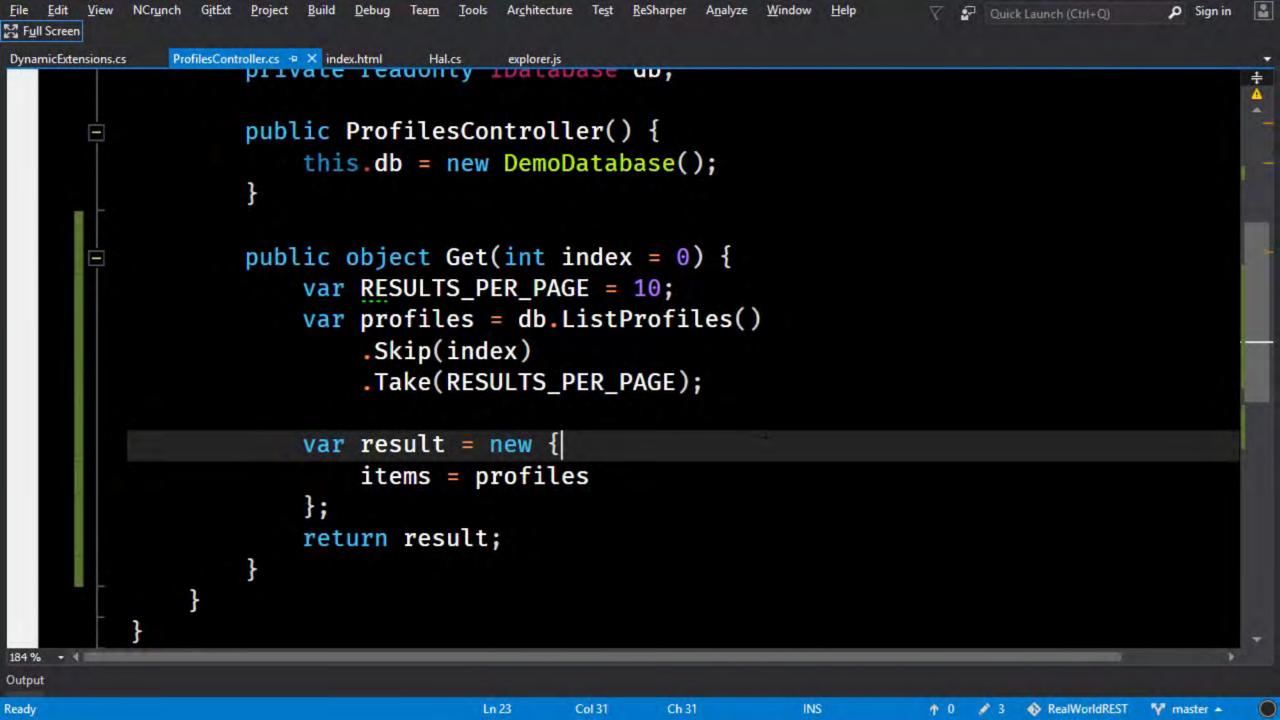


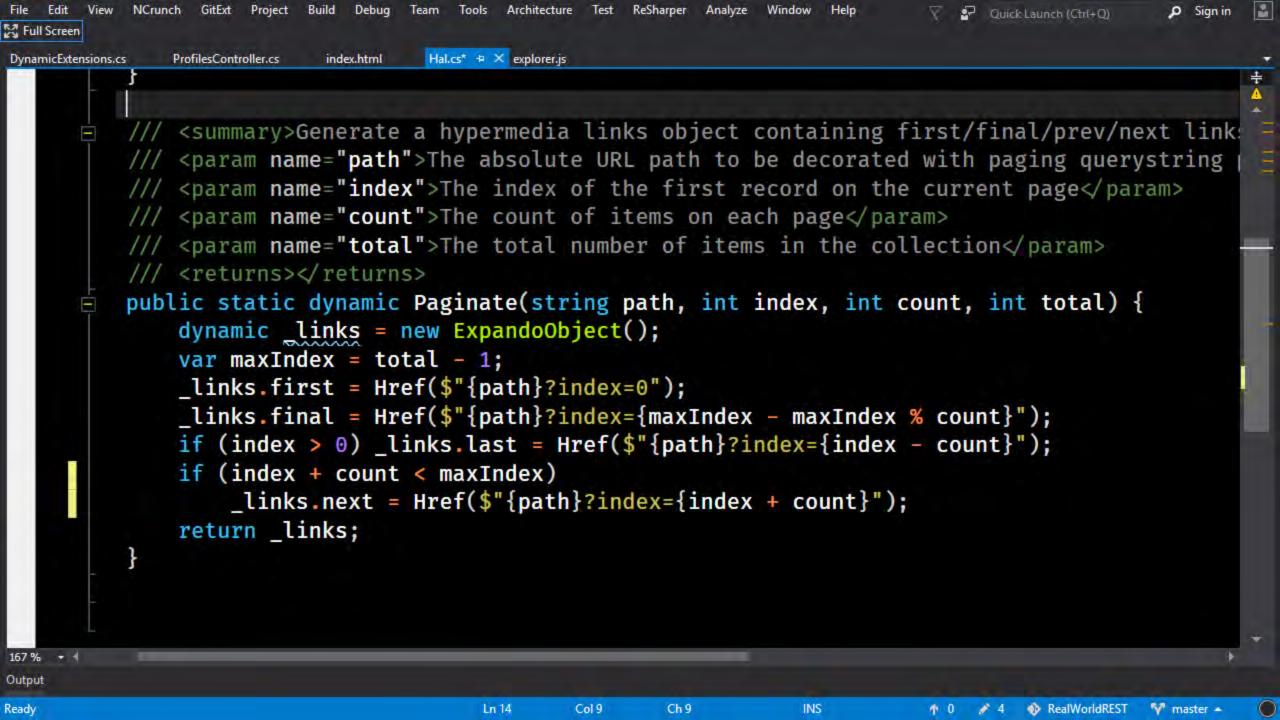


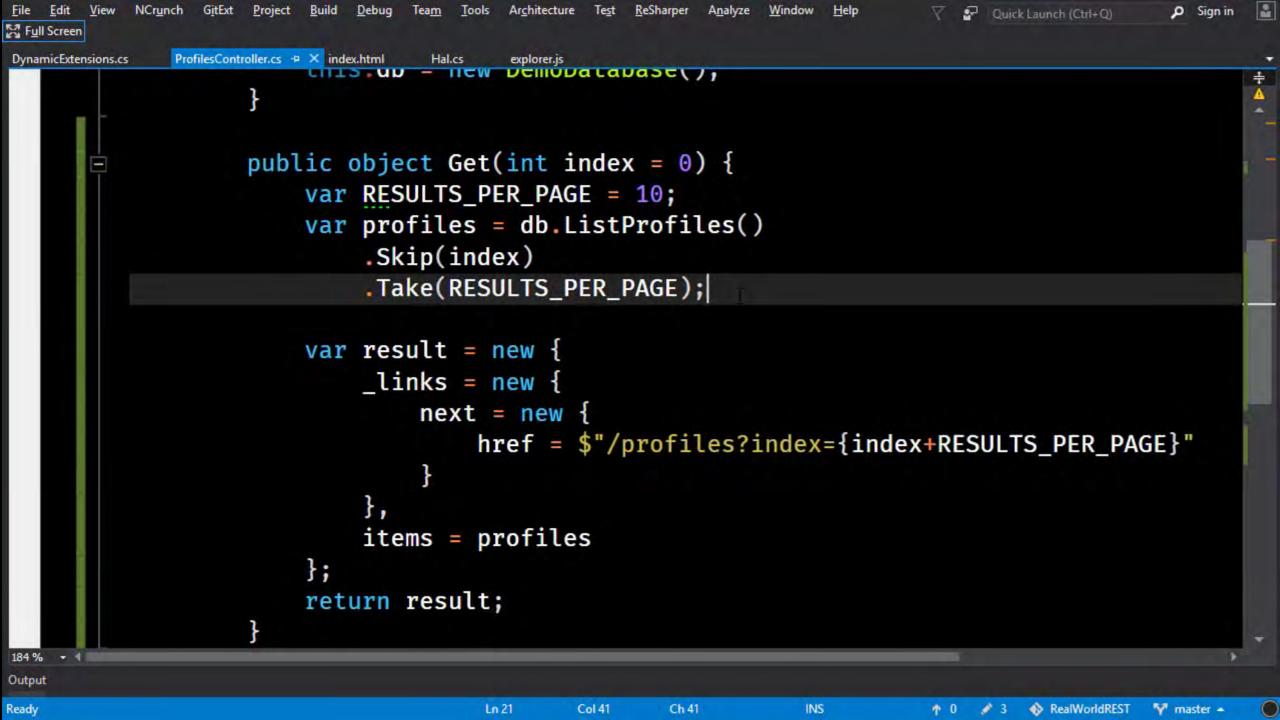












GET /profiles/ironman HTTP/1.1

```
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
  { "username":"ironman", "name":"Tony Stark" },
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  /* another 500 friends here... */
```

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
  { "username": "hulk", "name": "Bruce Banner" },
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  /* another 500 friends here... */
 "updates" : [
```

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
  { "username":"hulk", "name":"Bruce Banner" },
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  /* another 500 friends here... */
 "updates" : [
   "update": "New suit's gonna have a selfie stick! Oh yeah!",
    "posted": "2016-04-16T18:25:43.511Z"
```

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
  { "username":"hulk", "name":"Bruce Banner" },
  { "username": "spidey", "name": "Peter Parker" },
  { "username":"blackwidow", "name":"Ната́ша Романов" },
  /* another 500 friends here... */
 "updates" : [
   "update": "New suit's gonna have a selfie stick! Oh yeah!",
   "posted": "2016-04-16T18:25:43.511Z"
   "update": "Selfie stick broke. Oh well. Back to the drawing board",
   "posted": "2016-04-17T08:26:13.511Z"
```

GET /profiles/ironman HTTP/1.1

```
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
   "username":"hulk", "name":"Bruce Banner" },
   "username":"spidey", "name":"Peter Parker" },
   "username":"blackwidow", "name":"Ната́ша Романов" },

`another 500 friends here... */

 "updates" : [
   { "id" : 1234,
    "update" : "Working a new Iron Man suit!",
    "posted" : "2016-02-23T18:25:43.511Z"
   { "id" : 1543,
    "update" : "New suit's gonna have a selfie stick! Oh yeah!",
    "posted" : "2016-04-16T18:25:43.511Z"
   { "id" : 1782,
    "update" : "Selfie stick broke. Oh well. Back to the drawing board",
    "posted": "2016-04-17T08:26:13.511Z"
```

GET /profiles/ironman HTTP/1.1

```
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
  { "username":"hulk", "name":"Bruce Banner",
                                                        "updates" : [ { . . . }, { . . . }, { . . . } ]
   { "username": "spidey", "name": "Peter Parker",
                                                        "updates" : [ { . . . }, { . . . }, { . . . } ]
   { "username":"blackwidow", "name":"Ната́ша Романов", "<mark>updates" : [ { . . . }, { . . . }</mark> ]
   /* another 500 friends here... */
 "updates":[
   { "id" : 1234,
    "update": "Working a new Iron Man suit!",
    "posted": "2016-02-23T18:25:43.511Z"
   { "id" : 1543,
    "update": "New suit's gonna have a selfie stick! Oh yeah!",
    "posted": "2016-04-16T18:25:43.511Z"
   { "id" : 1782,
    "update": "Selfie stick broke. Oh well. Back to the drawing board",
    "posted": "2016-04-17T08:26:13.511Z"
```

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
 "username": "ironman",
 "name": "Tony Stark",
 "friends" : [
   "friends" : [ { . . . }, { . . . } ],
    "username": "spidey", "name": "Peter Parker", "updates" : [ { . . . }, { . . . } ]
    "friends" : [ { . . . }, { . . . } ],
    "friends" : [ { . . . }, { . . . } ],
   /* another 500 friends here... */
 "updates" : [
   { "id" : 1234,
     "update" : "Working a new Iron Man suit!",
     "posted": "2016-02-23T18:25:43.511Z"
     "id" : 1543,
     "update" : "New suit's gonna have a selfie stick! Oh yeah!",
     "posted": "2016-04-16T18:25:43.511Z"
     "id" : 1782,
     "update" : "Selfie stick broke. Oh well. Back to the drawing board",
     "posted": "2016-04-17T08:26:13.511Z"
```

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
  "username": "ironman",
  "name": "Tony Stark",
  "friends" : [
    { "username": "hulk", "name": "Bruce Banner",
      "updates" : [ \{ \ldots \}, \{ \ldots \}, \{ \ldots \} ],
      "friends" : [
           { "updates" : { . . . } }
     "username": "spidey", "name": "Peter Parker",
      "updates" : [ { . . . }, { . . . } ]
        "friends" : [
           { "updates" : { . . . } },
           { "updates" : { . . . } },
           { "updates" : { . . . } }
     "username":"blackwidow", "name":"Ната́ша Романов",
      "updates" : [ \{ ... \}, \{ ... \}, \{ ... \} ]
         "friends" : [
{ "updates" : { . . . } },
           { "updates" : { . . . } }, { "updates" : { . . . } }
    /* another 500 friends here... */
  "updates" : [
     { "id" : 1234,
       "update" : "Working a new Iron Man suit!",
       "posted": "2016-02-23T18:25:43.511Z"
       "id" : 1543,
       "update" : "New suit's gonna have a selfie stick! Oh yeah!",
       "posted": "2016-04-16T18:25:43.511Z"
       "id" : 1782,
       "update" : "Selfie stick broke. Oh well. Back to the drawing board",
        "posted" : "2016-04-17T08:26:13.511Z"
```

BIG

"Your stupid API is using all our bandwidth - again!"

"Your API has overloaded the database again...

I told you we should have built it using PHP."

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/hal+json
" links": {
  "self": { "href" "/profiles/ironman" },
  "friends": { "href": "/profiles/ironman/friends" },
  "photos": { "href": "/profiles/ironman/photos" },
  "updates": { "href": "/profiles/ironman/updates" }
 "username": "ironman",
"name" : "Tony Stark"
```

GET /profiles/ironman HTTP/1.1 200 OK GET /profiles/ironman/friends HTTP/1.1 200 OK GET /profiles/ironman/updates HTTP/1.1 200 OK GET /profiles/ironman/photos HTTP/1.1 200 OK GET /profiles/ironman/photos/1234 HTTP/1.1 200 OK GET /profiles/ironman/photos/1234/comments HTTP/1.1 200 OK GET /profiles/ironman/photos/1345 HTTP/1.1 200 OK GET /profiles/ironman/photos/1345/comments HTTP/1.1 200 OK GET /profiles/ironman/photos/1456 HTTP/1.1 200 OK GET /profiles/ironman/photos/1456/comments HTTP/1.1 200 OK

"I need to make 50 API calls just to draw one web page!"

"The web server just crashed because the IIS logs filled up the C: drive.

We never had this problem with PHP..."

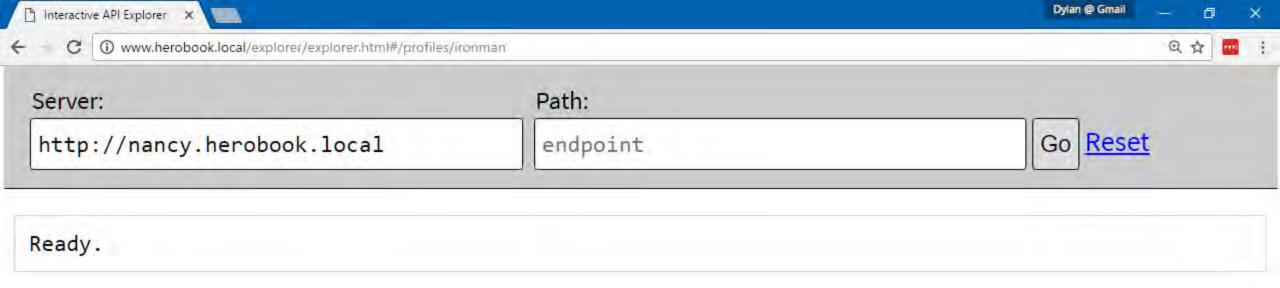
```
GET /profiles/ironman
200 OK
Content-Type: application/json
{
    "_links": {
        "self" : { "href": "/profiles/ironman" },
        "friends" : { "href": "/profiles/ironman/friends" },
        "photos" : { "href": "/profiles/ironman/photos" },
        "updates" : { "href": "/profiles/ironman/updates" }
    },
    "username" : "ironman",
    "name" : "Tony Stark"
```

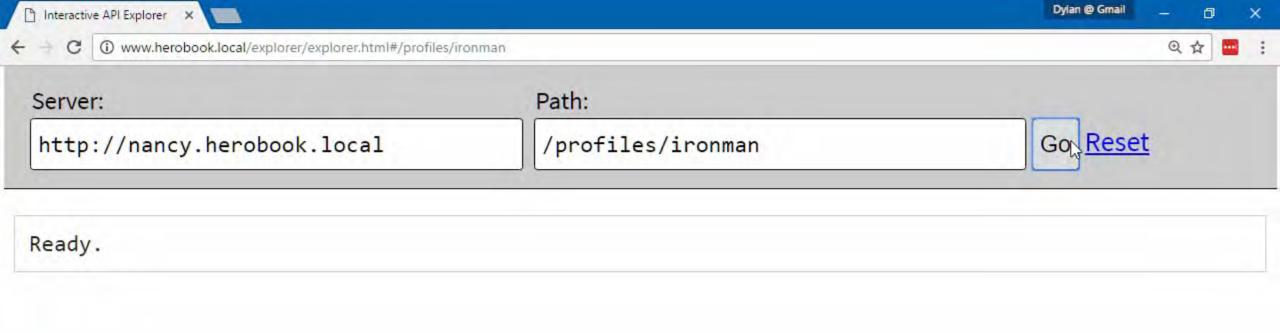
```
GET /profiles/ironman?expand=updates HTTP/1.1
200 OK
Content-Type: application/json
{
    "_links": {
        "self" : { "href": "/profiles/ironman" },
        "friends" : { "href": "/profiles/ironman/friends" },
        "photos" : { "href": "/profiles/ironman/photos" },
        "updates" : { "href": "/profiles/ironman/updates" }
    },
    "username" : "ironman",
    "name" : "Tony Stark"
    "_embedded" : {
```

```
GET /profiles/ironman?expand=updates HTTP/1.1
200 OK
Content-Type: application/json
 " links": {
  "self" : { "href": "/profiles/ironman" },
  "friends": { "href": "/profiles/ironman/friends" },
  "photos" : { "href": "/profiles/ironman/photos" },
  "updates" : { "href": "/profiles/ironman/updates" }
 "username": "ironman",
 "name" : "Tony Stark"
 " embedded":{
  "updates" : [
    "update": "Working a new Iron Man suit – with a built-in selfie stick!",
    "posted": "2016-02-23T18:25:43.511Z"
    "update": "Selfie stick broke. Oh well. Back to the drawing board",
    "posted": "2016-04-17T08:26:13.511Z"
```

DEMO #2

Resource Expansion





```
File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window

    Quick Launch (Ctrl+Q)

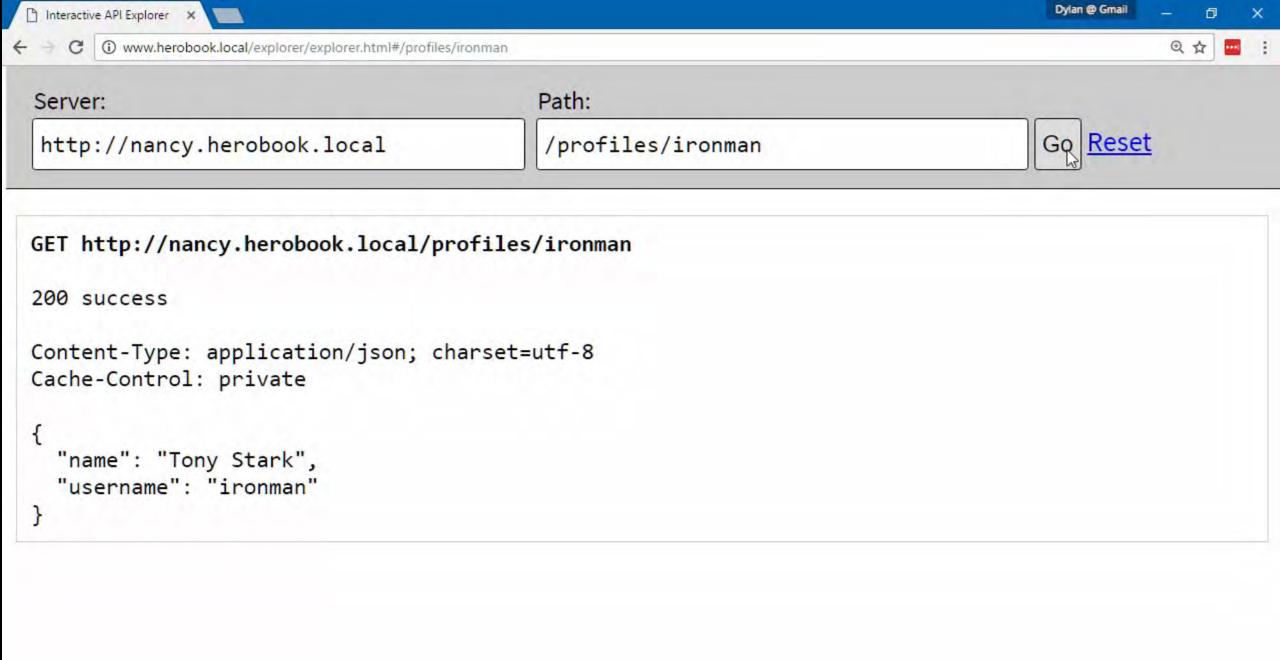
Full Screen
             ProfilesModule.cs ≠ ×
DynamicExtensions.cs
          namespace RealWorldRest.NancyFX.Modules {
            public class ProfilesModule : NancyModule {
               private readonly IDatabase db;
               public ProfilesModule(IDatabase db) {
                 this.db = db;
                 Get["/profiles"] = _ ⇒ GetProfiles();
                 Get["/profiles/{username}"] = args ⇒ GetProfile(args.username);
                 Get["/profiles/{name}/friends"] = args ⇒ GetFriends(args.name);
                 Post["/profiles"] = args ⇒ PostProfile(args);
                 Post["/profiles/{name}/friends"] = args ⇒ PostFriendship(args);
               private dynamic GetProfile(string username) {
                 var profile = db.LoadProfile(username);
                 return profile;
184 %
Output
                                               Col 3
                                                                                     RealWorldREST V master -
Item(s) Saved
                                                       Ch 3
                                                                  INS
```

Ln 11

```
File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
                                                                              Quick Launch (Ctrl+Q)
Full Screen
             ProfilesModule.cs ≠ X
DynamicExtensions.cs
                 Get["/profiles"] = _ ⇒ GetProfiles();
                 Get["/profiles/{username}"] = args ⇒ GetProfile(args.username);
                 Get["/profiles/{name}/friends"] = args ⇒ GetFriends(args.name);
                 Post["/profiles"] = args ⇒ PostProfile(args);
                 Post["/profiles/{name}/friends"] = args ⇒ PostFriendship(args);
               private dynamic GetProfile(string username) {
                 var profile = db.LoadProfile(username);
                 return profile;
               private dynamic GetProfiles() {
                 var index = (int?)Request.Query["index"] ?? 0;
                 var count = 10;
                 var profiles = db.ListProfiles().Skip(index).Take(count);
                 var total = db.CountProfiles();
                 var links - Hal Daginate ("/nrofiles" index count total).
184 %
Output
                                       Ln 25
                                              Col 46
                                                      Ch 46
                                                                 INS
                                                                               🧪 2 🚯 RealWorldREST 🐬 master 🔺
```

```
File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper
                                                          Analyze Window
                                                                                 Quick Launch (Ctrl+Q)
Full Screen
             ProfilesModule.cs* ≠ X
DynamicExtensions.cs
                 Get["/profiles"] = _ ⇒ GetProfiles();
                 Get["/profiles/{username}"] = args ⇒ GetProfile(args.username);
                 Get["/profiles/{name}/friends"] = args ⇒ GetFriends(args.name);
                 Post["/profiles"] = args ⇒ PostProfile(args);
                 Post["/profiles/{name}/friends"] = args ⇒ PostFriendship(args);
               private dynamic GetProfile(string username) {
                 var profile = db.LoadProfile(username);
                 profile._links = new { };
                 return profile;
               private dynamic GetProfiles() {
                 var index = (int?)Request.Query["index"] ?? 0;
                 var count = 10;
                 var profiles - dh listDrofiles() Skin(index) Take(count).
184 %
                                                                                   [3] (local variable) int count
Output
                                                                                  🧪 2 🚯 RealWorldREST 🐬 master 🔺
```

Ln 25 Col 45 Ch 45 INS



```
File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help

■ Quick Launch (Ctrl+Q)

Full Screen
DynamicExtensions.cs
             ProfilesModule.cs* ≠ X
              private dynamic GetProfile(string username) {
                 var profile = db.LoadProfile(username).ToDynamic();
                 profile._links = new {
                   self = new { href = $"/profiles/{username}"},
                   friends = new { href = $"/profiles/{username}/friends"}
                 };
                 return profile;
              private dynamic GetProfiles() {
                var index = (int?)Request.Query["index"] ?? 0;
                var count = 10;
                var profiles = db.ListProfiles().Skip(index).Take(count);
                 var total = db.CountProfiles();
                var links - Hal Daginate ("/nrofiles" index count total).
Output
```

Ln 31

Col 1

↑ 0

RealWorldREST

₩ master •

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/json
 " links": {
  "self" : { "href": "/profiles/ironman" },
  "friends": { "href": "/profiles/ironman/friends" },
  "photos" : { "href": "/profiles/ironman/photos" },
  "updates" : { "href": "/profiles/ironman/updates" }
 "name": "Tony Stark",
 "username": "ironman",
 "height": 192,
 "weight": 85,
 "location": { "latitude": 59.93, "longitude": 30.33 },
 "status": "Out saving the world. Again.",
 "hometown": "Malibu, USA"
 "email": "tony@stark.com",
 "website": "www.ironman.com",
 "last modified" : "2015-08-12T19:45:43.511Z"
```

PUT /profiles/ironman HTTP/1.1

```
Content-Type: application/json
 "name": "Tony Stark",
"username": "ironman",
 "height": 192,
"weight": 85,
 "location": { "lat": 34.02, "lon": -118.77 },
"status": "Just got back from saving the world. Again.",
 "hometown": "Malibu, USA"
"email": "tony@stark.com",
"website": "www.ironman.com",
 "birthdate": "1972-01-24"
```

409 Conflict

"Why do I need to
PUT the entire profile
just to update some
coordinates?"

"All my updates keep failing with a 409 Conflict – HELP!"

PUT /profiles/ironman HTTP/1.1 Content-Type: application/json "status": "Thinking about a disco Iron Man suit." /* note JSON doesn't include ANY other fields */

204 No Content

PUT /profiles/ironman/status HTTP/1.1

"Finished saving the world. I need a drink."

204 No Content

The PATCH method requests that

a set of changes

described in the request entity be applied to the resource identified by the Request URI.

The set of changes is represented in a format called a "patch document" identified by a media type.

PATCH /file.txt HTTP/1.1

Host: www.example.com

Content-Type: application/example

If-Match: "e0023aa4e"

Content-Length: 100

[description of changes]

PATCH /file.txt HTTP/1.1

Host: www.api.herobook.local

Content-Type: application/example

If-Match: "e0023aa4e"

Content-Length: 100

[description of changes]

```
PATCH /profiles/ironman HTTP/1.1
Content-Type: application/x-unix-diff
11c11
"status": "Just got home from saving the world.",
"status": "World saved. I need a drink.",
```

200 OK

```
PATCH /profiles/ironman HTTP/1.1
Content-Type: application/json-patch+json
If-Match: "abc123"
  "op": "replace",
  "path": "/status",
  "value": "Finished saving the world. I need a drink."
 "op": "move",
  "from": "/friends/hulk",
  "path": "/friends[0]"
```

200 OK

"Oh, integration will be easy...
we have a REST API."

"Our software doesn't do that but you can use our REST API to build your own plugins!"



Paldies!

Slides & code:

https://github.com/dylanbeattie/RealWorldREST

@dylanbeattie

dylan@dylanbeattie.net

