Building

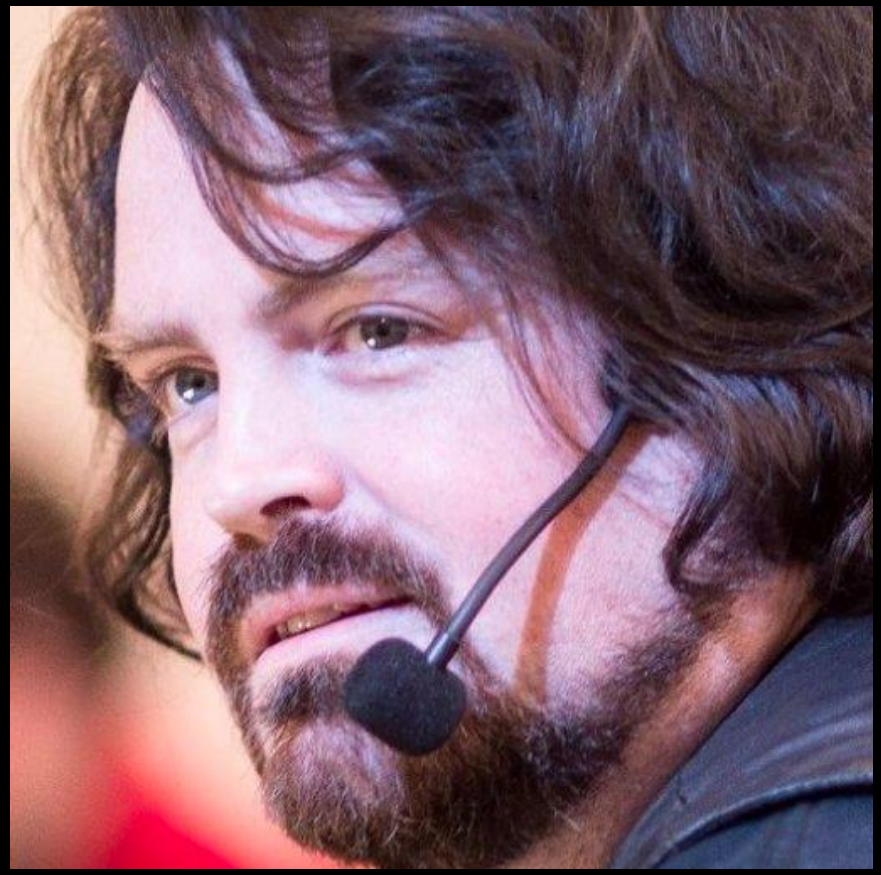
RESTful APIs

with

ASP.NET

@dylanbeattie

SELA|DEVELOPER|PRACTICE



Dylan Beattie

@dylanbeattie

- Building websites since 1992
- Systems architect at www.spotlight.com
- Web site: www.dylanbeattie.net

Slides & code:

www.github.com/dylanbeattie/sela2017/

Workshop Programme

09:00 Introduction: Defining REST.

09:20 Hypermedia as the Engine of Application State

09:40 Resource decomposition

10:00 Resource Expansion

10:30-10:45 BREAK

10:45 Representing Actions with Hypermedia

11:15 Implementing HTTP PATCH

11:55 Demo: Tools, Tips and Tricks

12:15 - 13:15 LUNCH

13:15 Content Negotiation

13:35 Caching and Layering

13:45 Security and Authentication

14:15 API Versioning

14:45 – 15:00 BREAK

15:00 Testing and Monitoring APIs

15:30 Discussion: API Strategy

16:00 FINISH

Introduction

Defining REST

Architectural Constraints

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.**

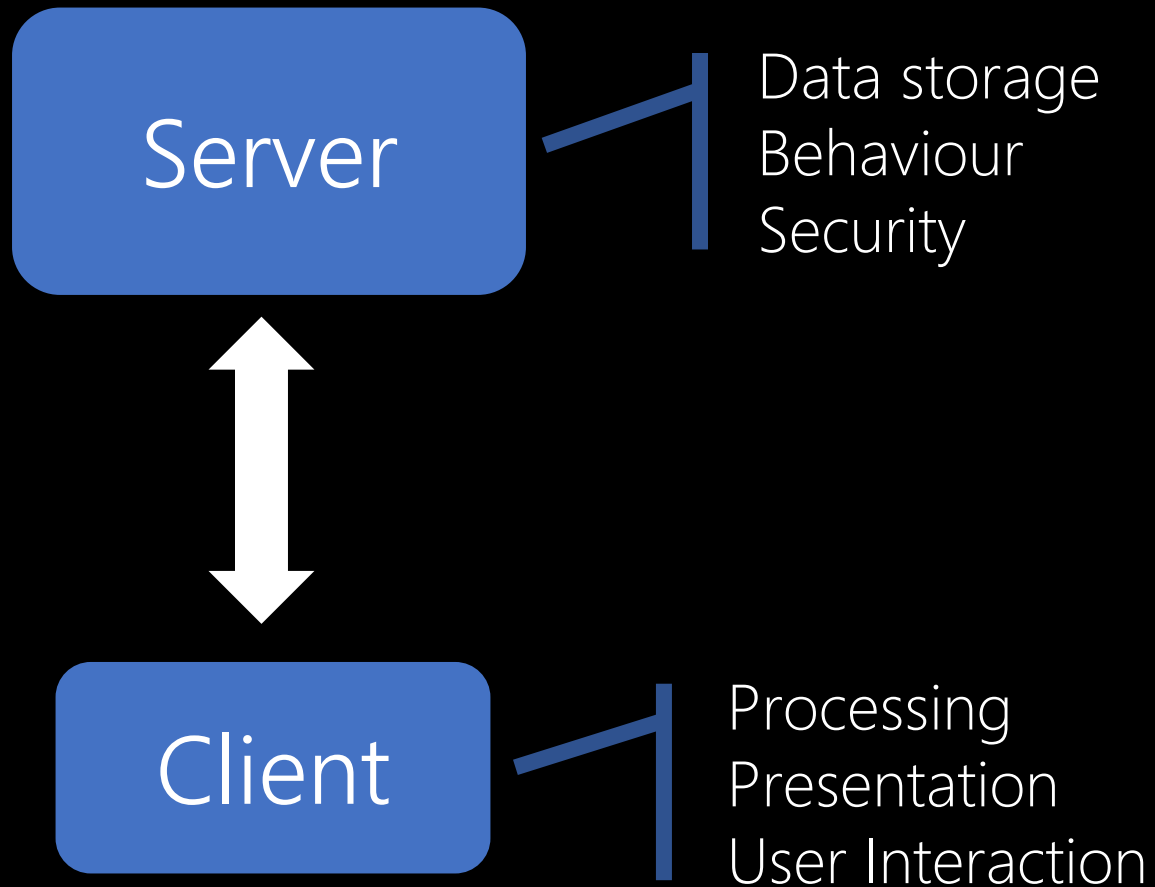
Many of the constraints are **directly opposed to short-term efficiency.**"

Representational State Transfer (REST)

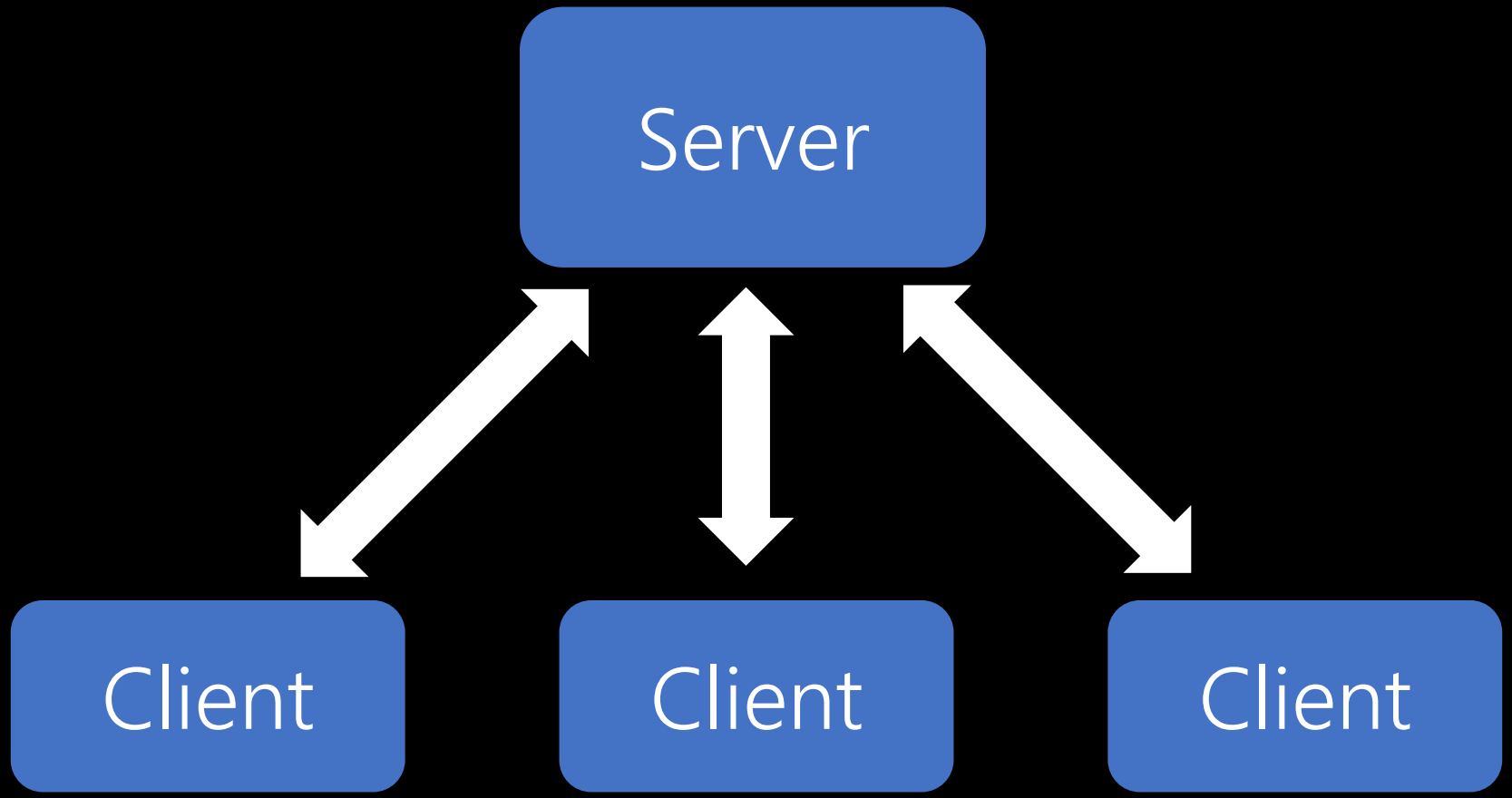
A coordinated set of architectural constraints...

...intended to promote software longevity and independent evolution

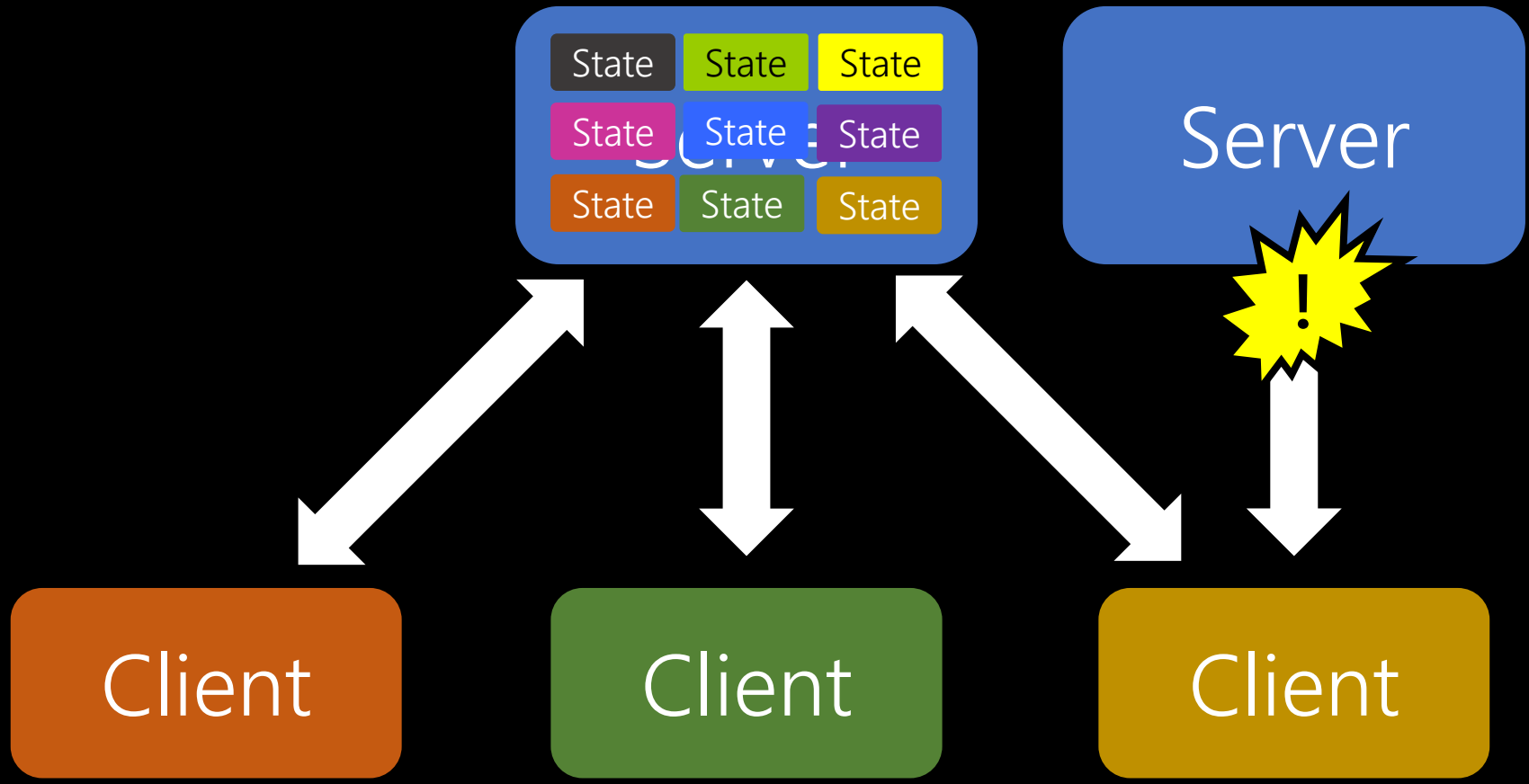
1. Client–Server



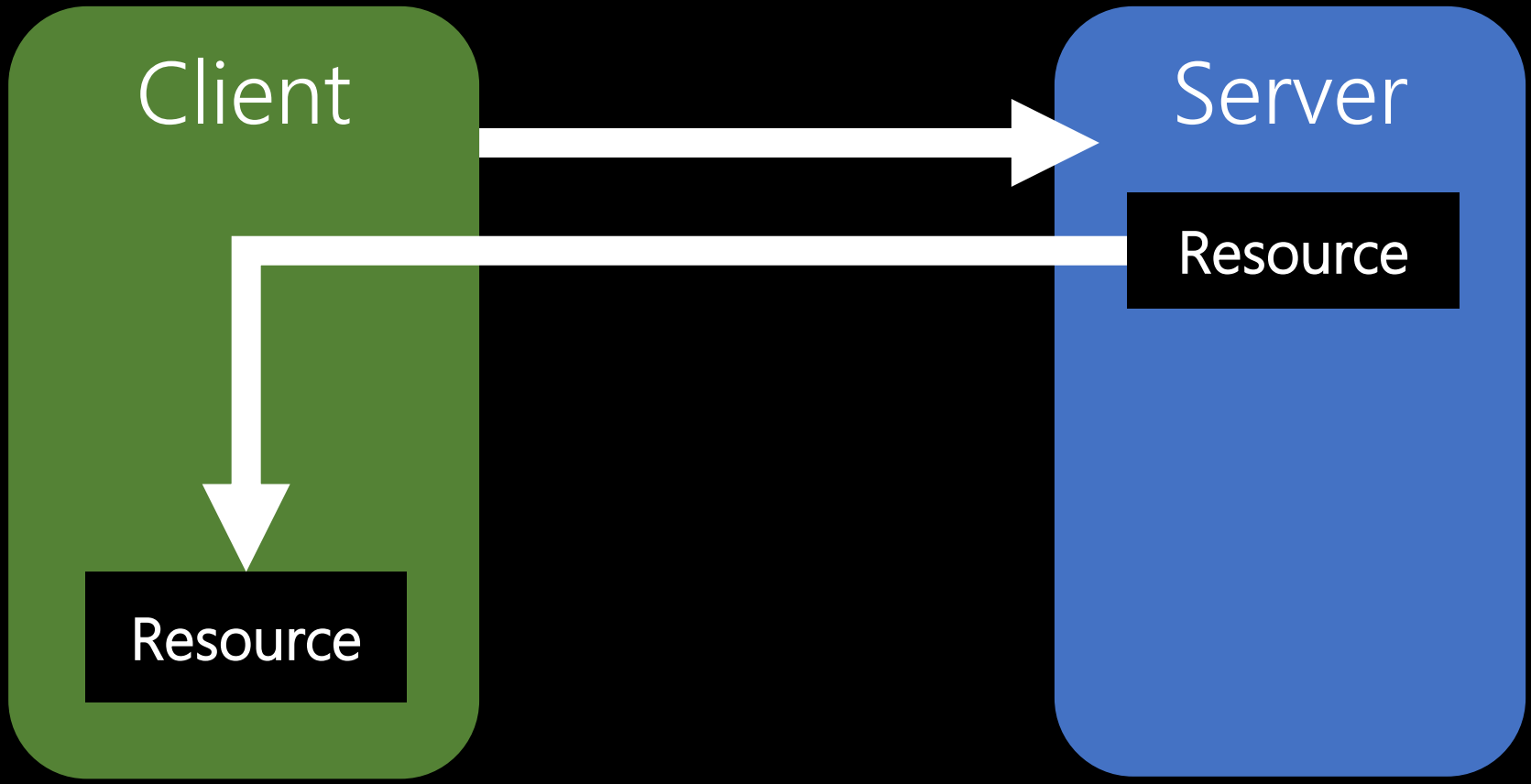
1. Client–Server



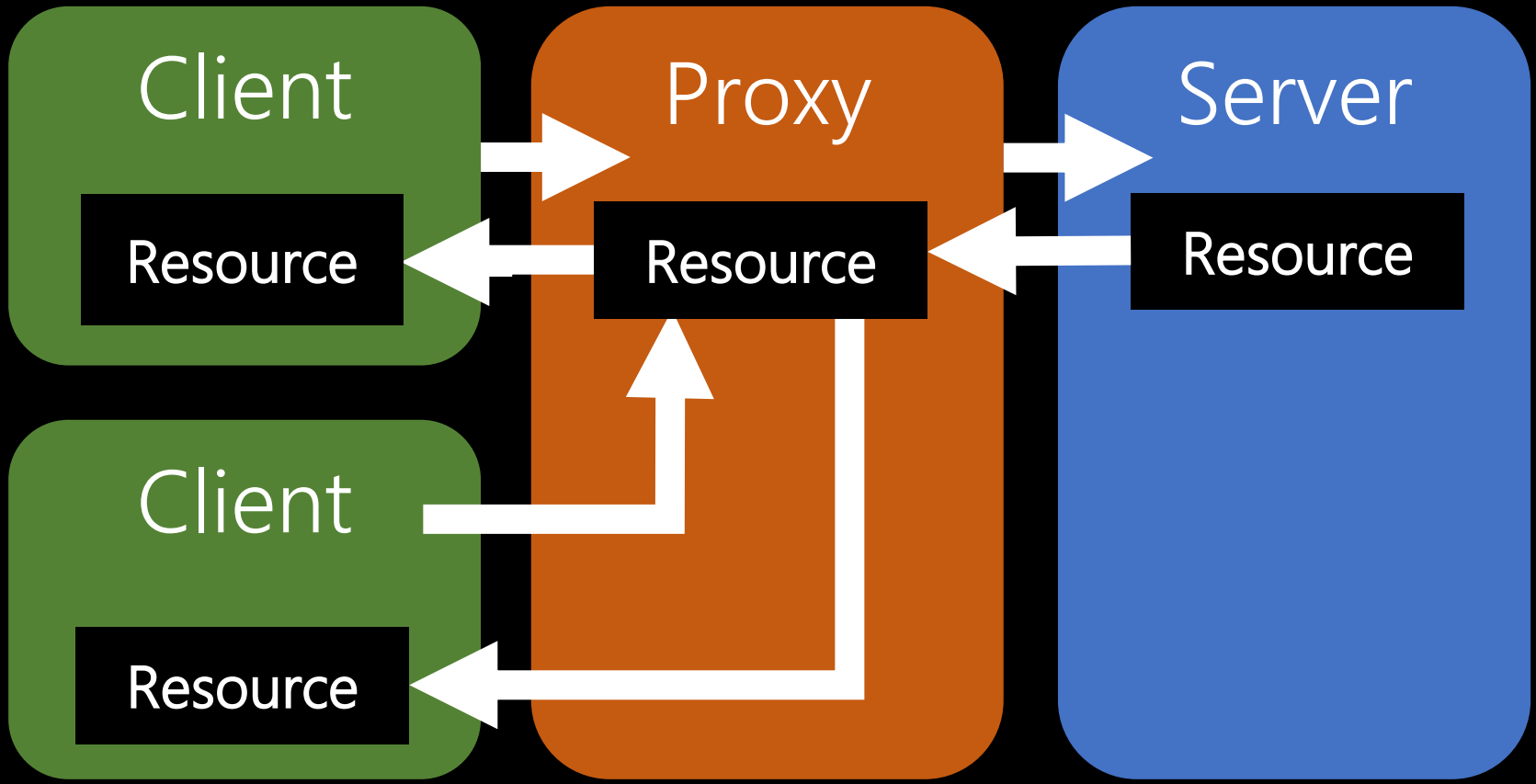
2. Stateless


State
State
State
State
State
State
State
State
State
Server
!
Client
Client
Client

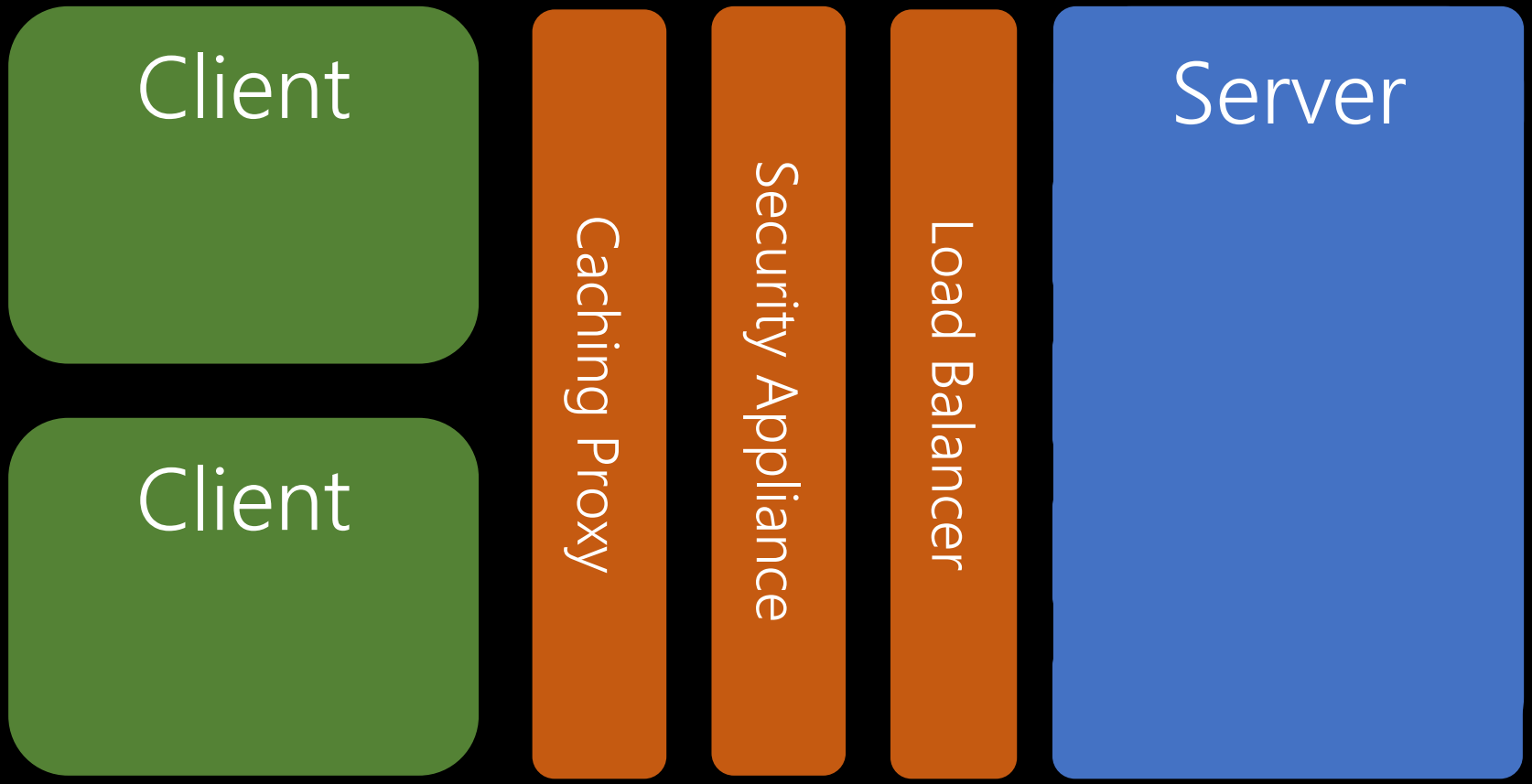
3. Cacheable



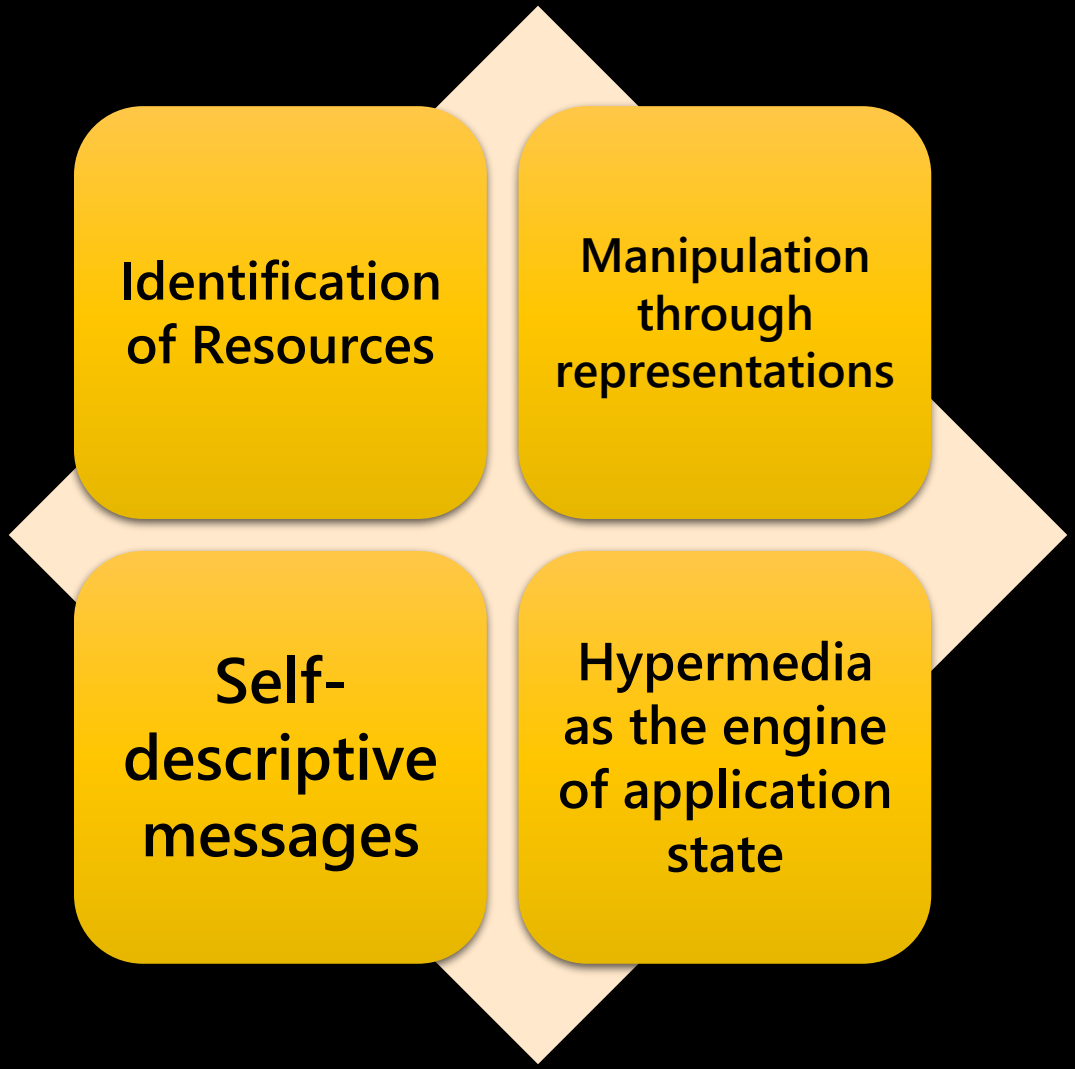
4. Layered System



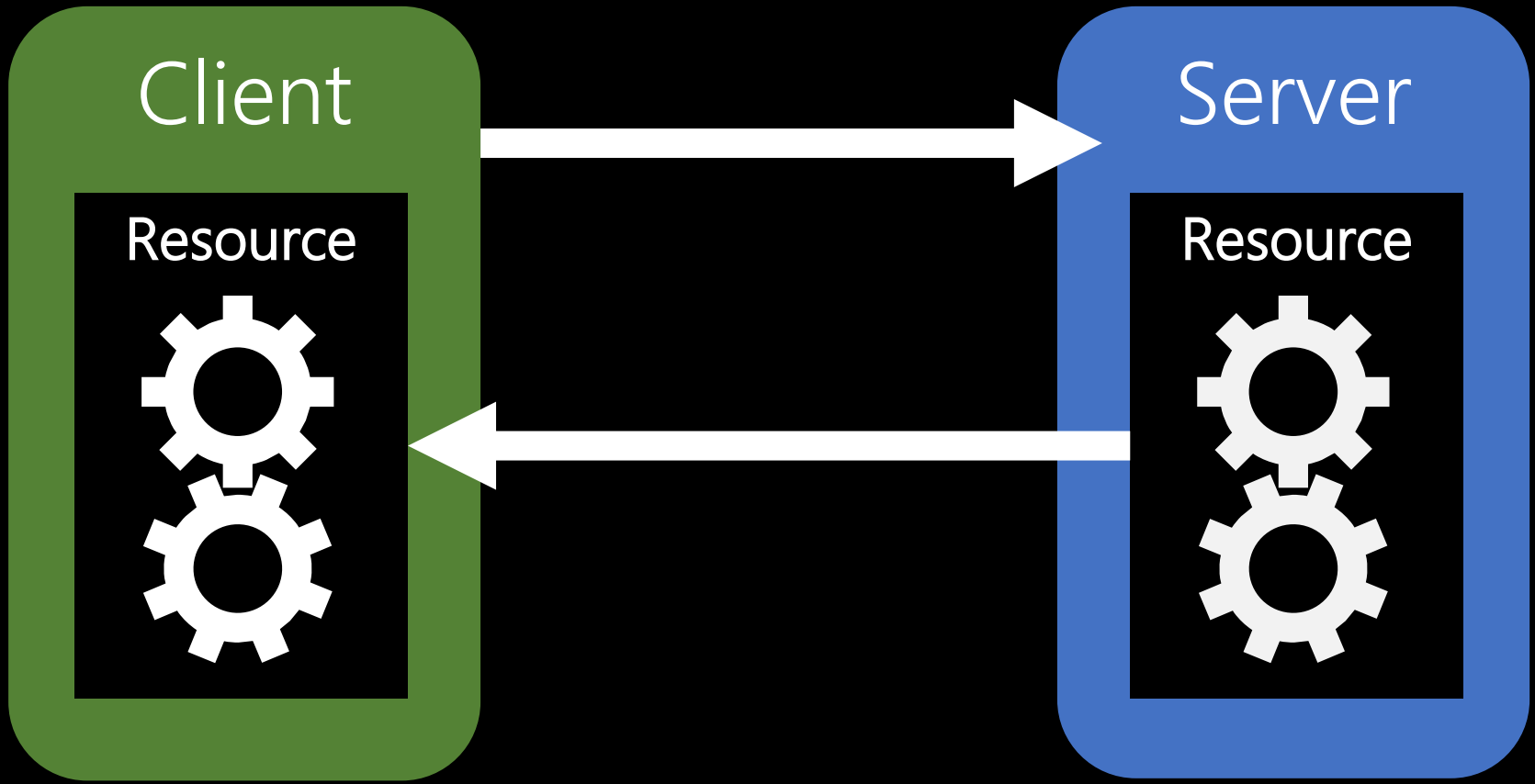
4. Layered System (continued)



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

Module 1:

Hypermedia as the Engine of Application State

```
GET /profiles HTTP/1.1
200 OK
{
  "_links" : { … },
  "items" : {
    { "username":"ironman", "name":"Tony Stark" }
    { "username":"blackwidow", "name":"Natasha Romanoff" },
    { "username":"spidey", "name":"Peter Parker" },

    { /* +2 million users! Wow! */ },

    { "username":"ducky", "name":"Howard the Duck" }
  }
]
```

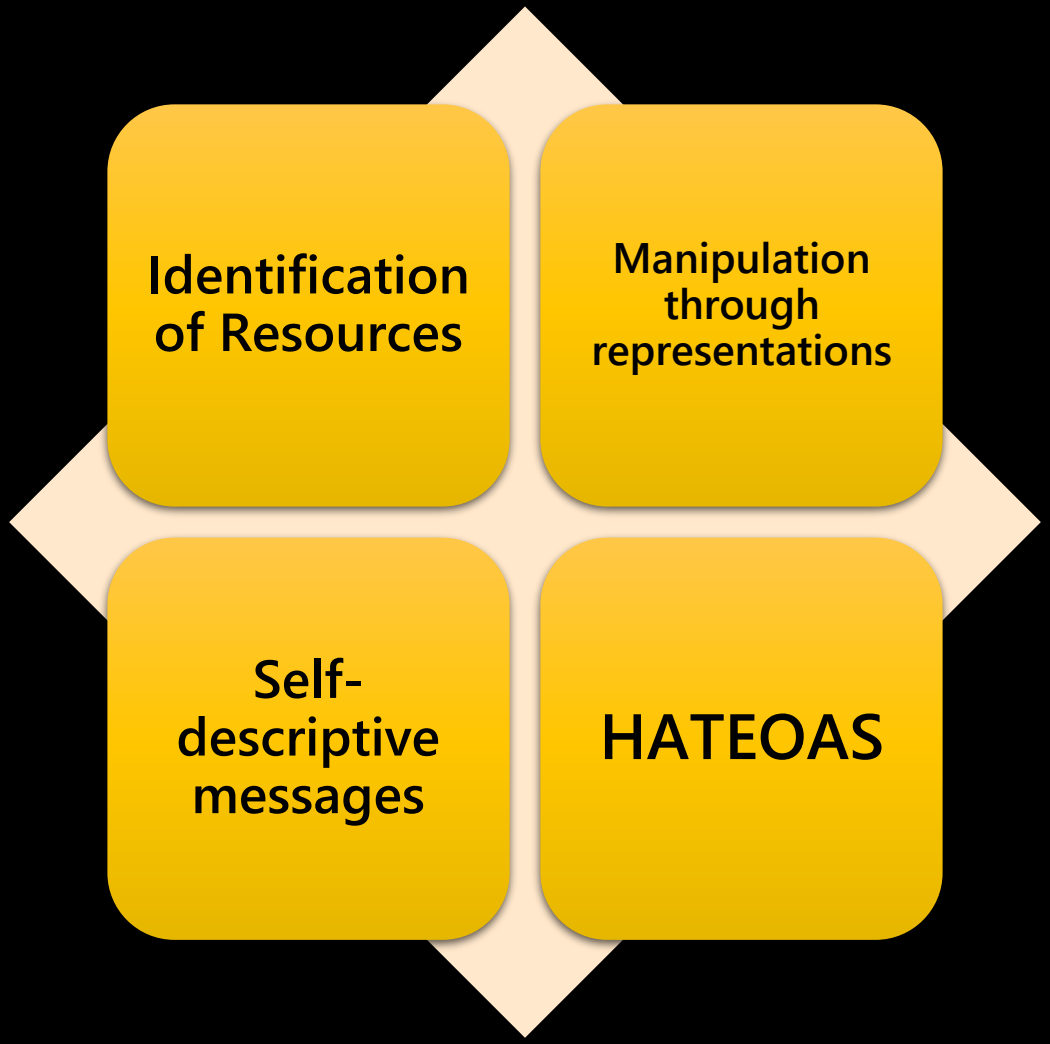
"Your API is using all our bandwidth!"

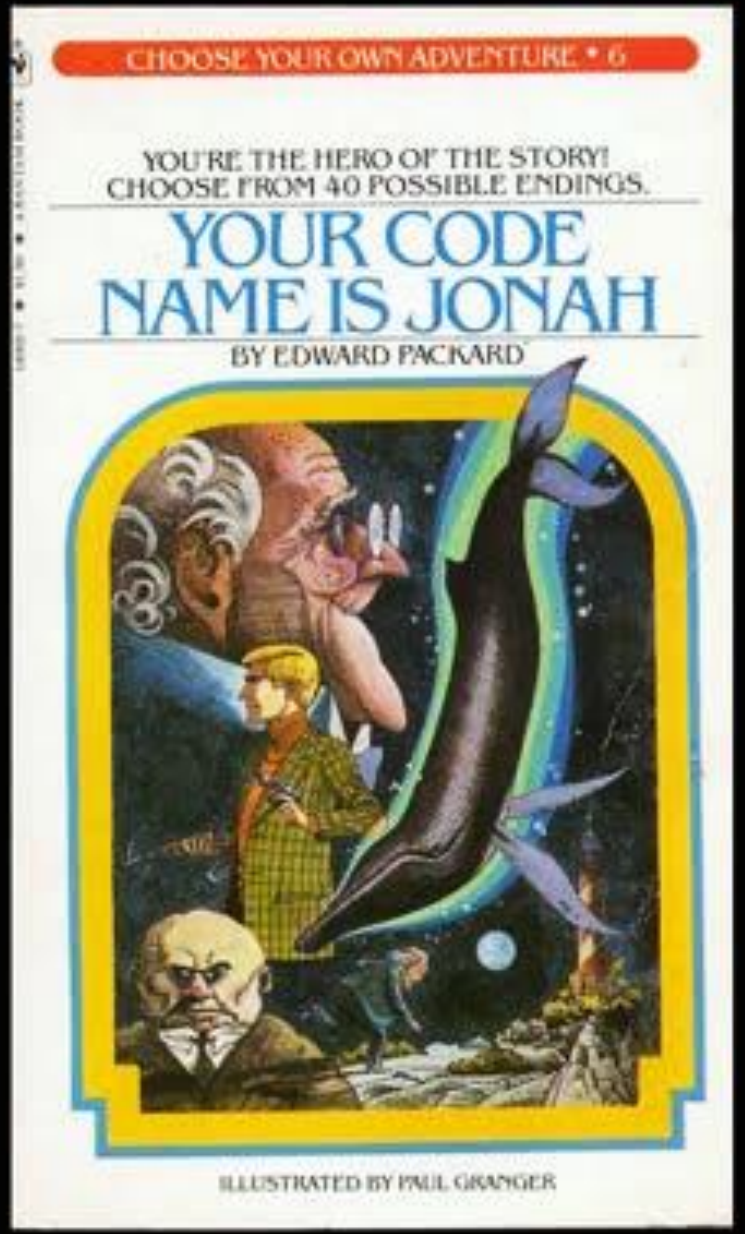
"Your stupid API is overloading our database!"

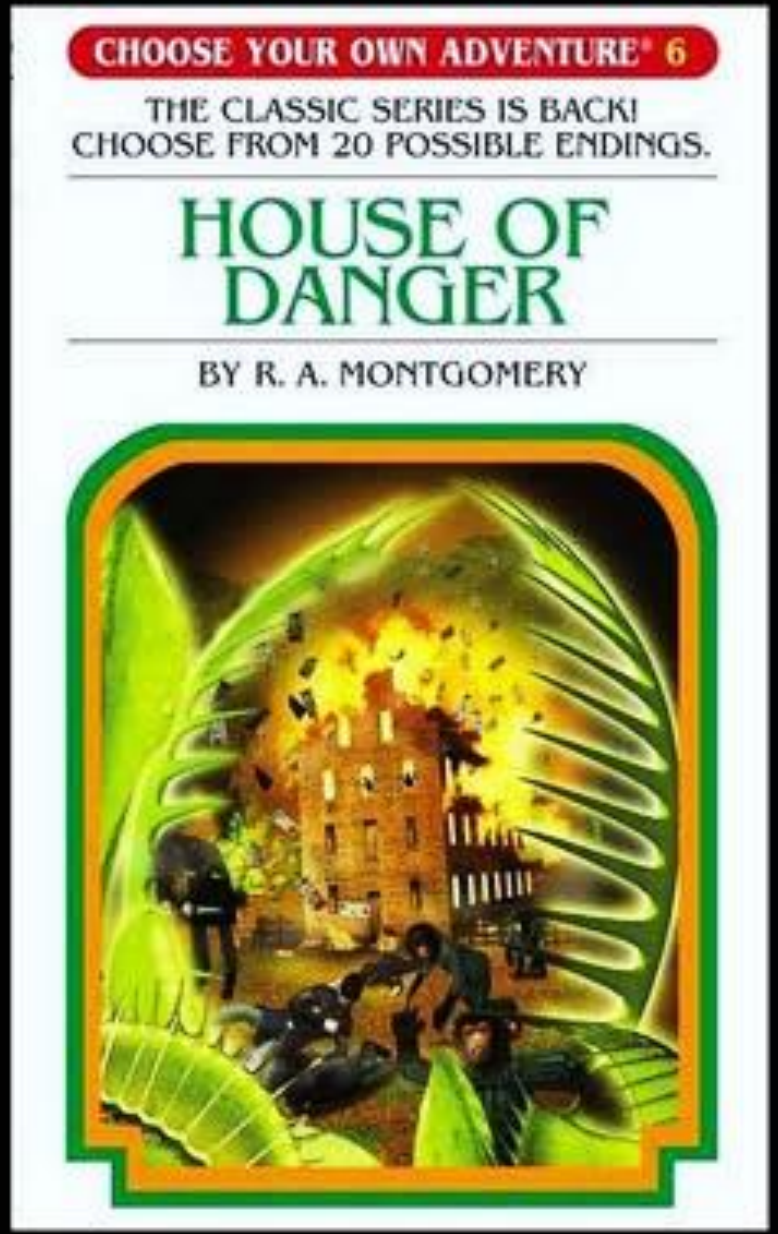
```
GET /profiles?page=1 HTTP/1.1
200 OK
{
  "_links" : { … },
  "items" : [
    { "username":"ironman", "name":"Tony Stark" }
    { "username":"spidey", "name":"Peter Parker" },
    { "username":"blackwidow", "name": "Natasha Romanoff" },
    { "username":"cap", "name":"Steve Rogers" },
    { "username":"storm", "name":"Ororo Munroe" }
  ]
}
```

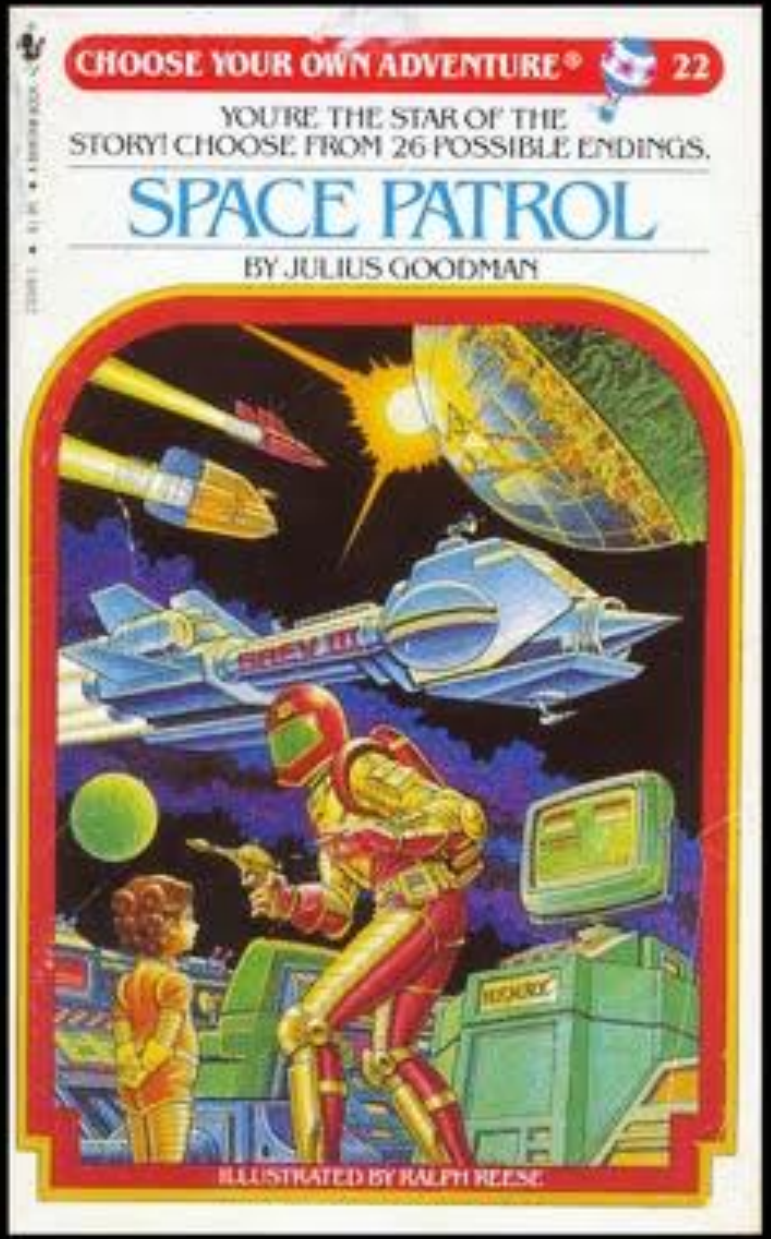
```
GET /profiles?page=2 HTTP/1.1
200 OK

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

GET /profiles?page=4 HTTP/1.1
200 OK

GET /profiles?page=5 HTTP/1.1
204 No Content
```

5. Uniform Interface









CHOOSE YOUR OWN ADVENTURE! #SELASDP

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

HANDS-ON HYPERMEDIA

SELA DEVELOPER PRACTISE

It's Monday night in Tel Aviv... you've finished dinner, and you're deciding what to do later. You're going to a REST workshop at the Sela Developer Practise tomorrow, so you should probably get some sleep... but some friends are meeting up at Beerbazaar in Habima Square for drinks later, and you just found out Malox are playing tonight down at Rothschild 12. What are you going to do?

To go home and get some sleep, turn to page 24

To go to Beer Bazaar and try some Israeli craft beers, turn to page 41

To head down to Rothschild 12 and dance to Malox, turn to page 73

```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
{
  "_links" : {



  },
  "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" }
  ]
}
```

```
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" }
  ]
}
```

```
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" }
  ]
}
```

Module 1:

Hypermedia as the Engine of Application State DEMO

Interactive API Explorer

Dylan @ Gmail

www.herobook.local/explorer/explorer.html#/profiles?

Server:

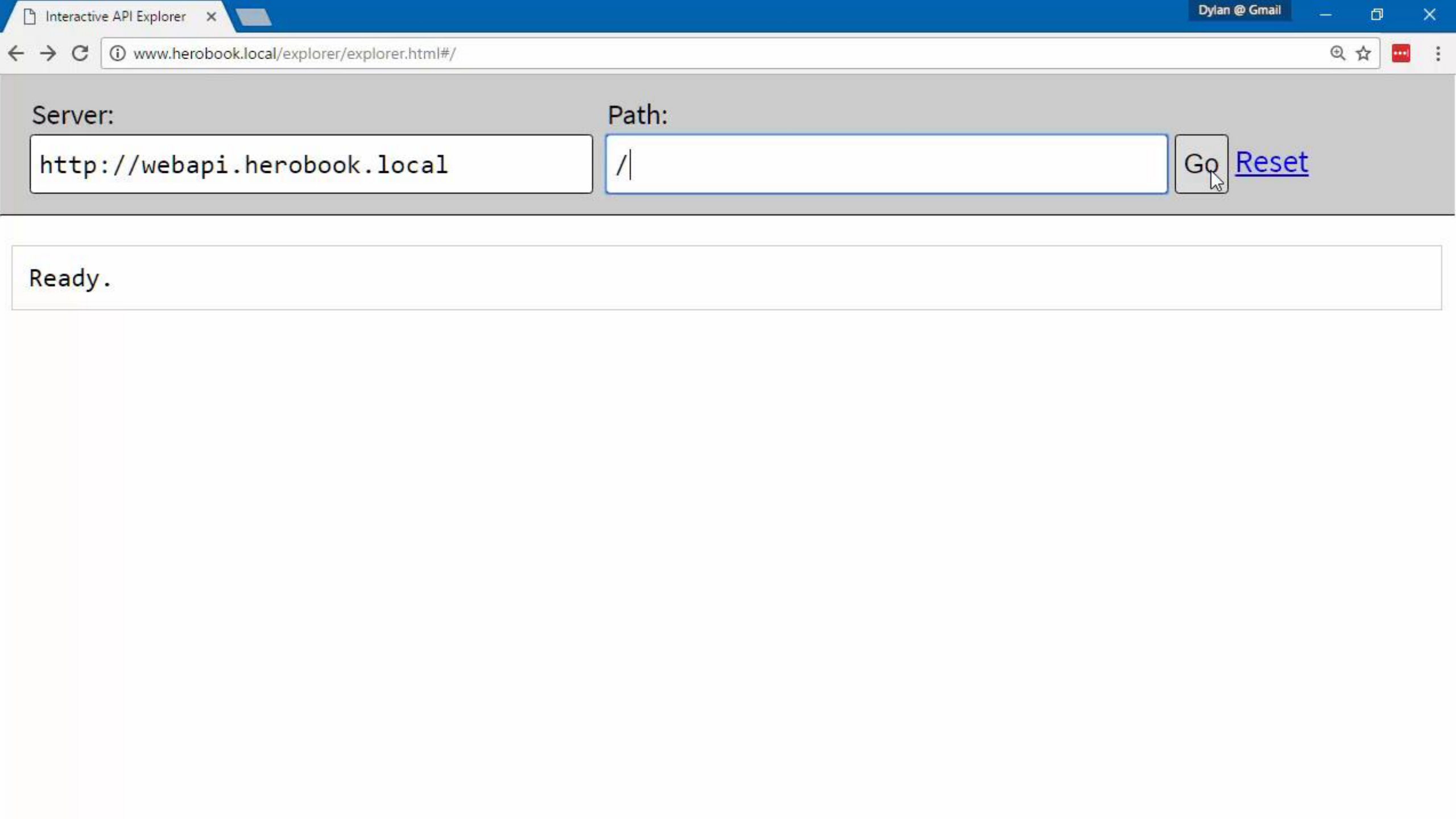
http://webapi.herobook.local

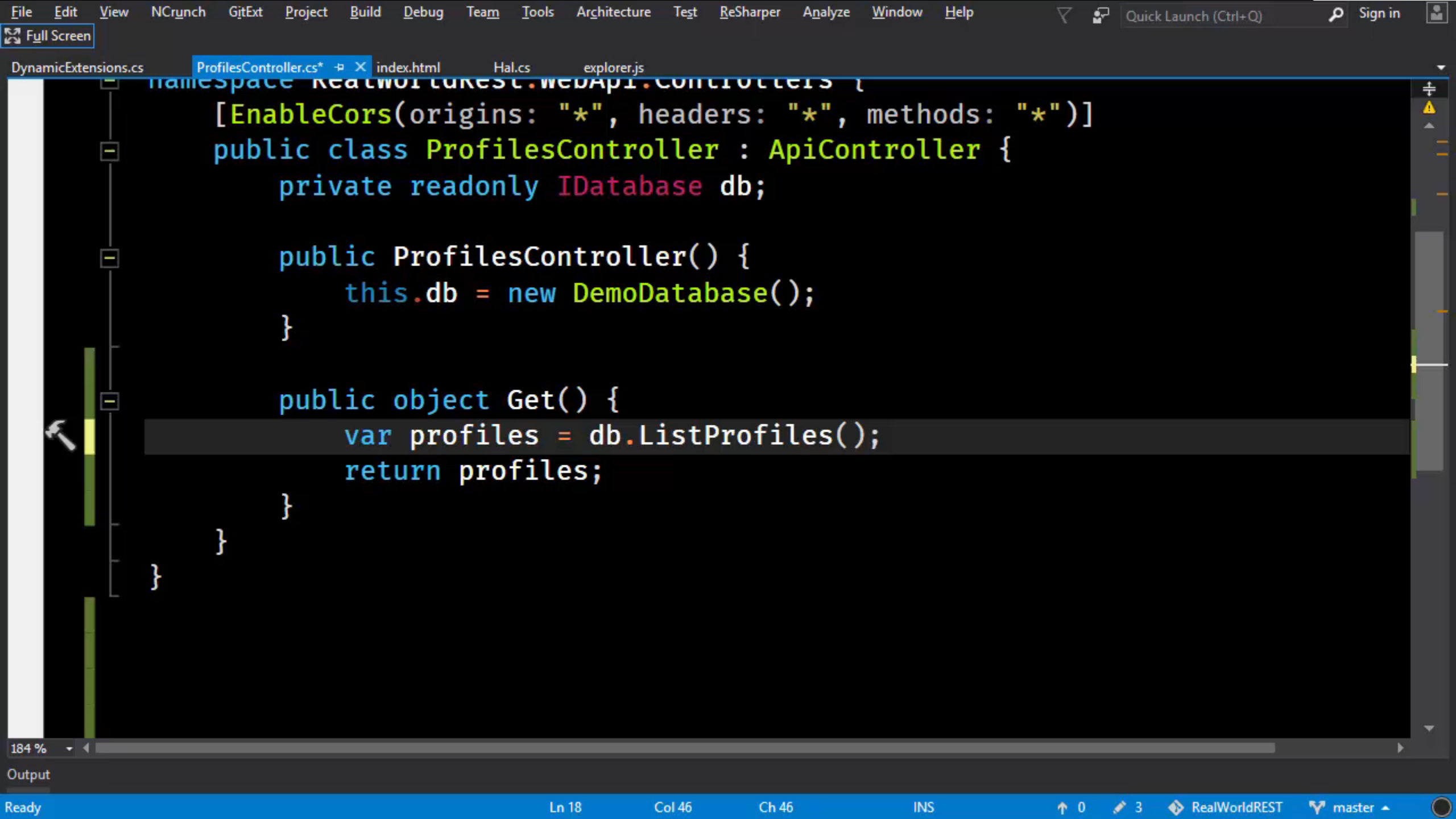
Path:

/

Go Reset

Ready.


Interactive API Explorer
Dylan @ Gmail
www.herobook.local/explorer/explorer.html#/
Server:
http://webapi.herobook.local
Path:
/
Go
Reset
Ready.


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs*
index.html
Hal.cs
explorer.js
namespace RealWorldRest.WebApi.Controllers {
[EnableCors(origins: "*", headers: "*", methods: "*")]
public class ProfilesController : ApiController {
private readonly IDatabase db;
public ProfilesController() {
this.db = new DemoDatabase();
}
public object Get() {
var profiles = db.ListProfiles();
return profiles;
}
}
}
184 %
Output
Ready
Ln 18
Col 46
Ch 46
INS
0
3
RealWorldREST
master

Interactive API Explorer

Dylan @ Gmail

www.herobook.local/explorer/explorer.html#/profiles

Server: http://webapi.herobook.local

Path: /profiles

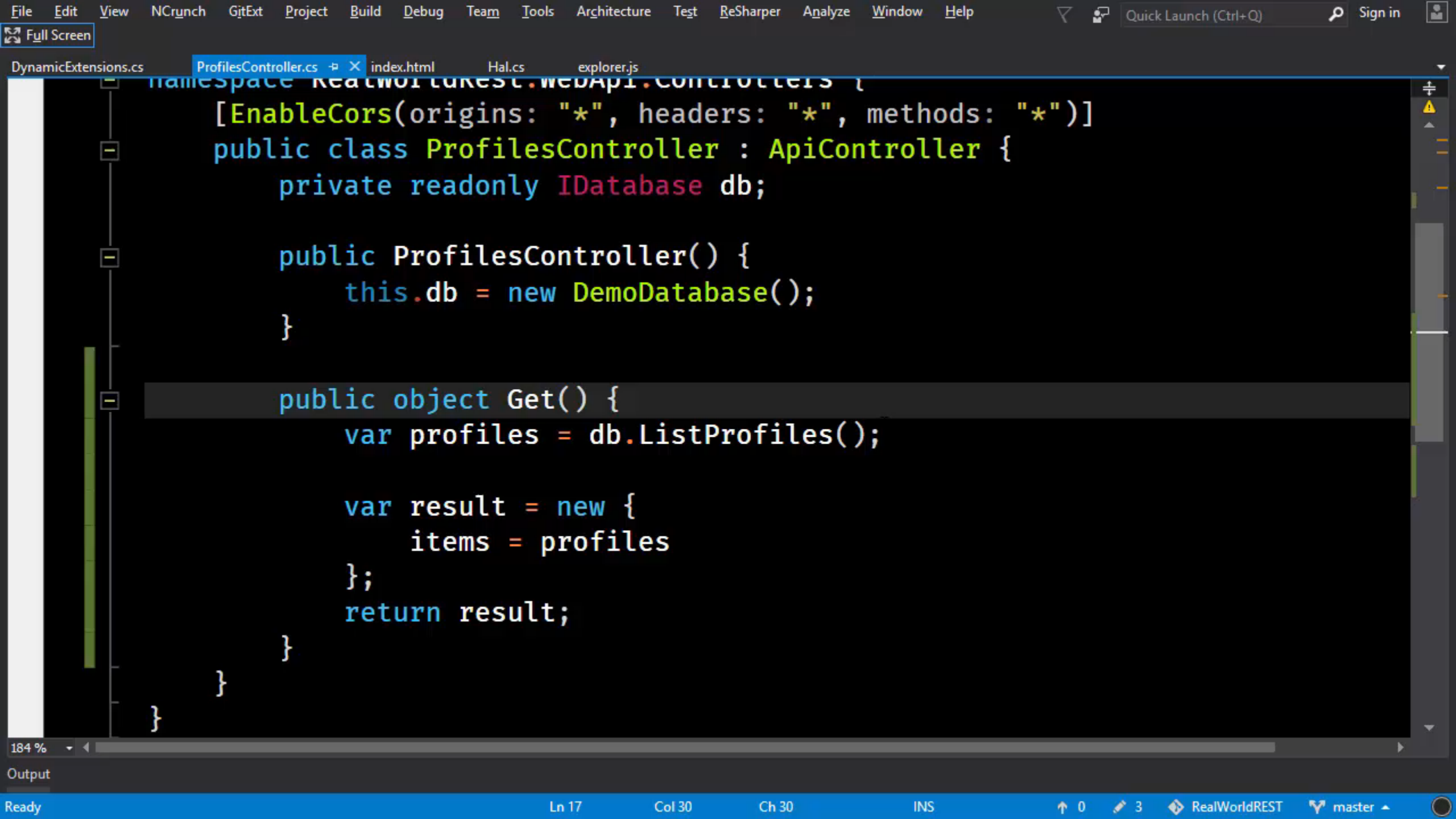
Go Reset

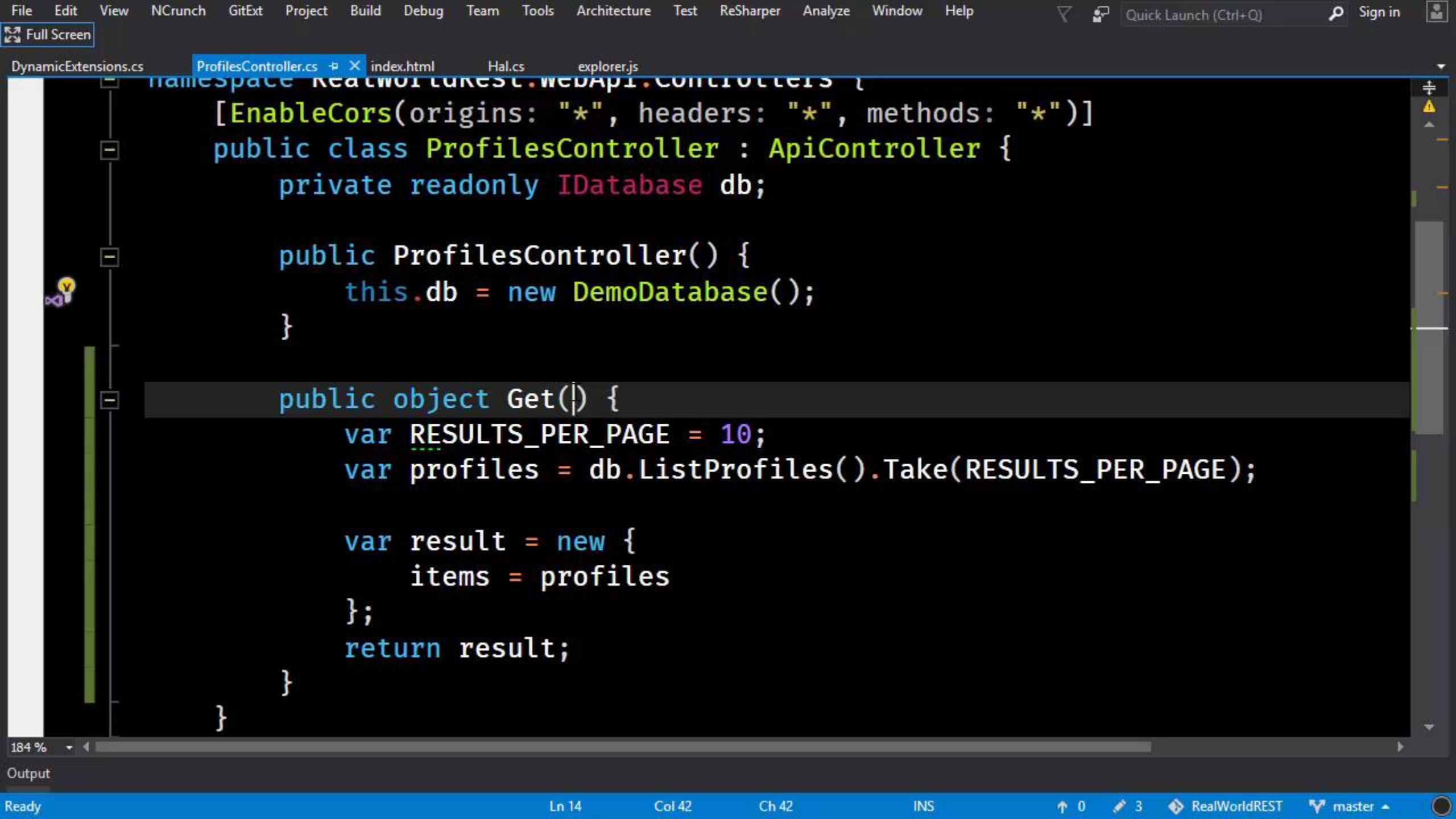
GET http://webapi.herobook.local/profiles

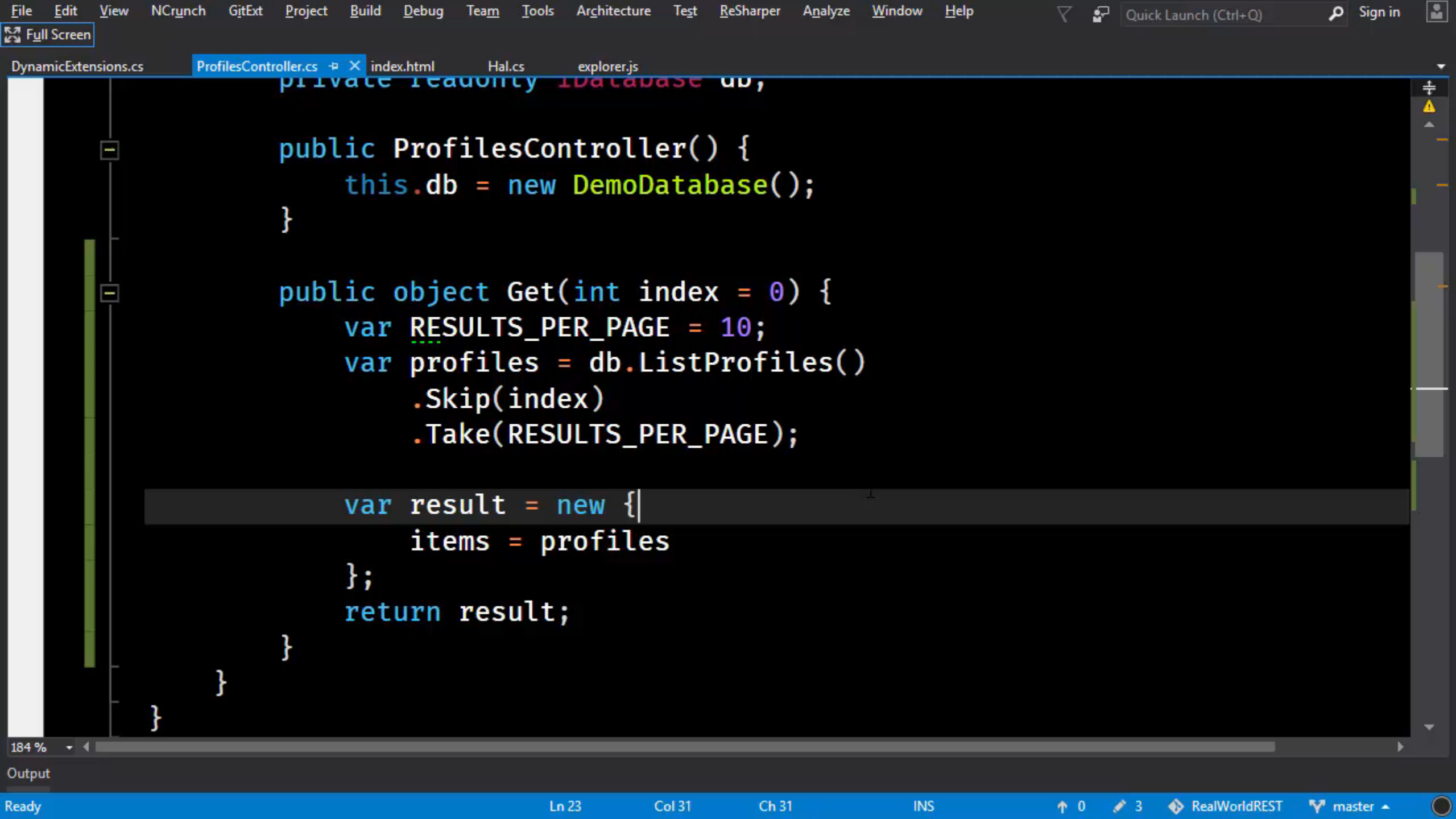
200 success

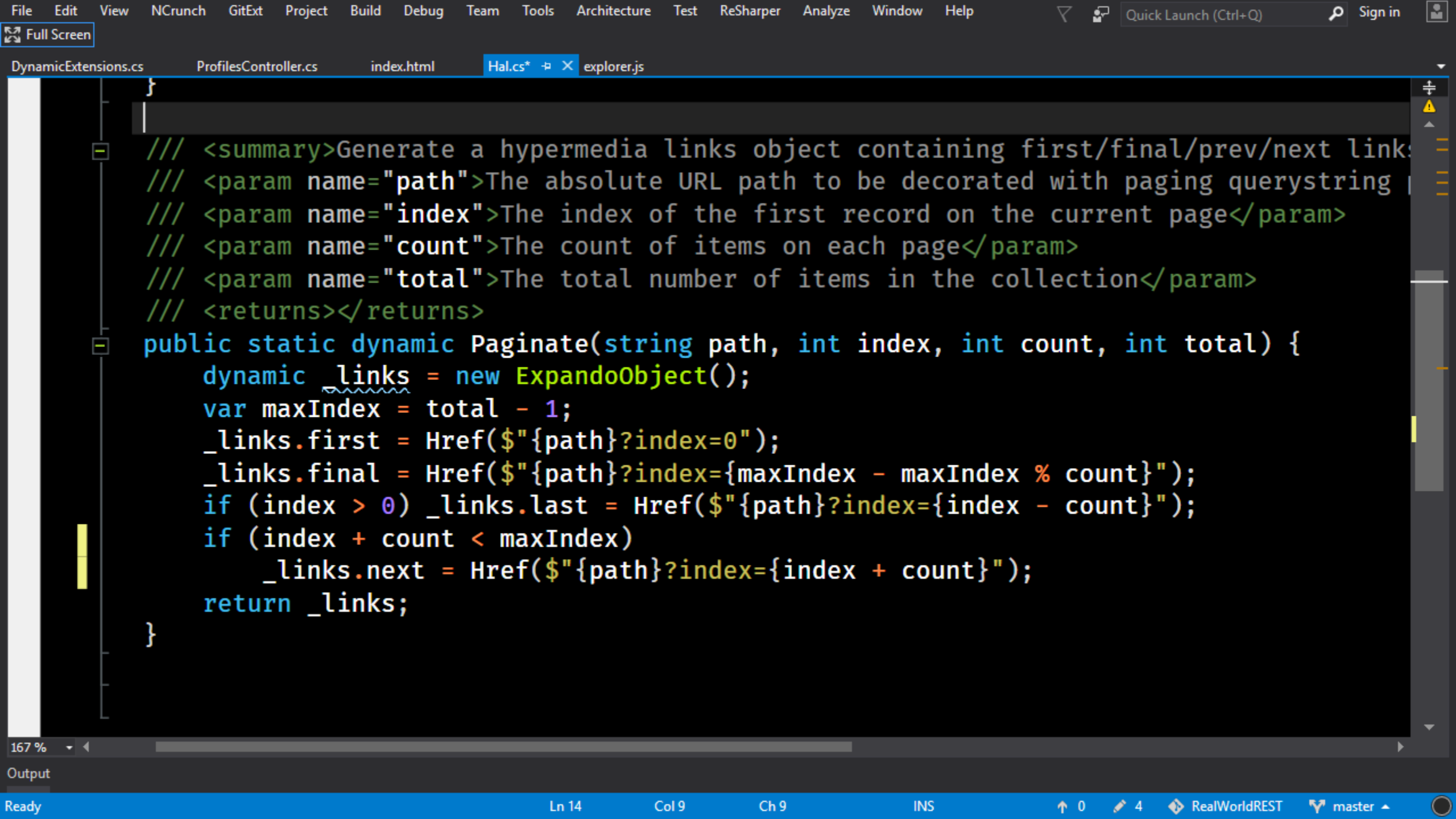
```
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Cache-Control: no-cache
Expires: -1
```

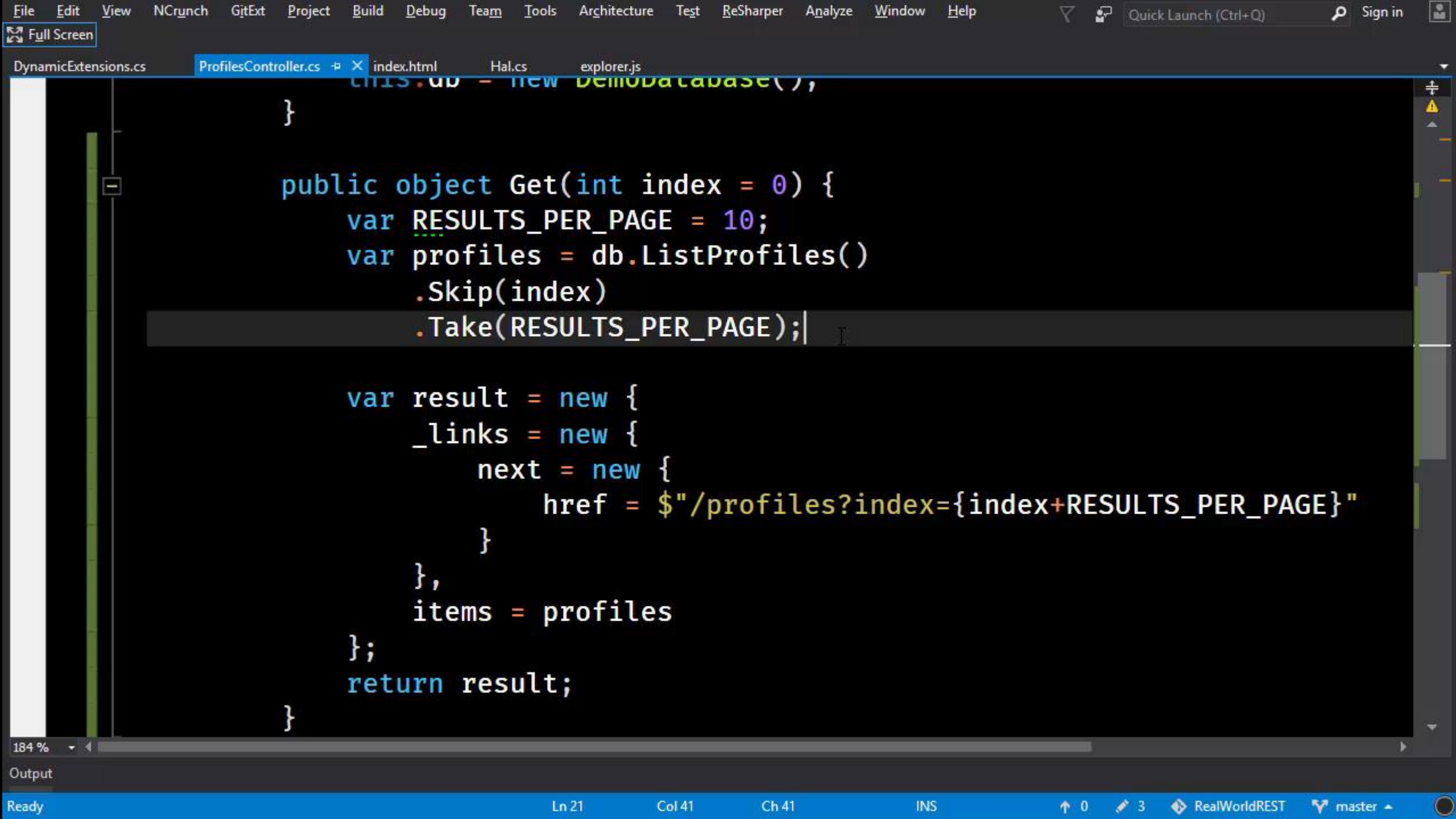

```
[
  {
    "Name": "Captain Lea Corbin",
    "Username": "4d"
  },
  {
    "Name": "Steve Austin",
    "Username": "6milliondollarman"
  },
```


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs
index.html
Hal.cs
explorer.js
namespace RealWorldRest.WebApi.Controllers {
[EnableCors(origins: "*", headers: "*", methods: "*")]
public class ProfilesController : ApiController {
private readonly IDatabase db;
public ProfilesController() {
this.db = new DemoDatabase();
}
public object Get() {
var profiles = db.ListProfiles();
var result = new {
items = profiles
};
return result;
}
}
}
184 %
Output
Ready
Ln 17
Col 30
Ch 30
INS
0
3
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs
index.html
Hal.cs
explorer.js
[EnableCors(origins: "*", headers: "*", methods: "*")]
public class ProfilesController : ApiController {
private readonly IDatabase db;
public ProfilesController() {
this.db = new DemoDatabase();
}
public object Get() {
var RESULTS_PER_PAGE = 10;
var profiles = db.ListProfiles().Take(RESULTS_PER_PAGE);
var result = new {
items = profiles
};
return result;
}
}
184 %
Output
Ready
Ln 14
Col 42
Ch 42
INS
0
3
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs
index.html
Hal.cs
explorer.js
private readonly IDatabase db;
public ProfilesController() {
this.db = new DemoDatabase();
}
public object Get(int index = 0) {
var RESULTS_PER_PAGE = 10;
var profiles = db.ListProfiles()
.Skip(index)
.Take(RESULTS_PER_PAGE);
var result = new {
items = profiles
};
return result;
}
}
}
184 %
Output
Ready
Ln 23
Col 31
Ch 31
INS
0
3
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs
index.html
Hal.cs*
explorer.js
}
/// <summary>Generate a hypermedia links object containing first/final/prev/next link
/// <param name="path">The absolute URL path to be decorated with paging querystring
/// <param name="index">The index of the first record on the current page</param>
/// <param name="count">The count of items on each page</param>
/// <param name="total">The total number of items in the collection</param>
/// <returns></returns>
public static dynamic Paginate(string path, int index, int count, int total) {
dynamic _links = new ExpandoObject();
var maxIndex = total - 1;
_links.first = Href($"{path}?index=0");
_links.final = Href($"{path}?index={maxIndex - maxIndex % count}");
if (index > 0) _links.last = Href($"{path}?index={index - count}");
if (index + count < maxIndex)
_links.next = Href($"{path}?index={index + count}");
return _links;
}
167 %
Output
Ready
Ln 14
Col 9
Ch 9
INS
0
4
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesController.cs
index.html
Hal.cs
explorer.js
this.db = new DemoDatabase();
}
public object Get(int index = 0) {
var RESULTS_PER_PAGE = 10;
var profiles = db.ListProfiles()
.Skip(index)
.Take(RESULTS_PER_PAGE);
var result = new {
_links = new {
next = new {
href = $"/profiles?index={index+RESULTS_PER_PAGE}"
}
},
items = profiles
};
return result;
}
184 %
Output
Ready
Ln 21
Col 41
Ch 41
INS
0
3
RealWorldREST
master

- Workshop: Hypermedia as the Engine of Application State

Code:

http://www.github.com/dylanbeattie/sela2017workshop/

Follow instructions in workshop.pdf

Module 1: Hypermedia as the Engine of Application State

Discussion Points

- Do linked resources have to be on the same server?
- How could you design a partition scheme that would split your user profile data across multiple database servers?

Module 2:

Resource and

Resource Composition

```
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",    "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"
     }
  ]
}
```

```
GET /profiles/ironman HTTP/1.1

200 OK
Content-Type: application/json

{
  "username": "ironman",
  "name": "Tony Stark",
  "friends" : [
    { "username":"hulk", "name":"Bruce Banner", "updates" : [ { . . . },    { . . . }, { . . . } ],
      "friends" : [ { . . . },  { . . . }, { . . . } ],

    },
    { "username":"spidey", "name":"Peter Parker", "updates" : [ { . . . },  { . . . }, { . . . } ]
      "friends" : [ { . . . },  { . . . }, { . . . } ],
    },
    { "username":"blackwidow", "name":"Ната́ша Романов", "updates" : [ { . . . },      { . . . } , { . . . } ]
      "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" : [ { . . . },   { . . . }, { . . . } ],
      "friends" : [
          { "updates" : { . . . }  },
          { "updates" : { . . . }  },
          { "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 DATA!

"Your API is using all our bandwidth... AGAIN!"

"Your API is overloading the database. Again.

...I told you we should have just stuck with 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"
}
```

Module 2:

Demo:

Resources and Navigation Properties

Decompositing Aggregate Resources

Module 2:

Workshop

Module 2: Hypermedia as the Engine of Application State

Discussion Points

- Are there situations where this might be the best solution?
- What are the similiarities between this and eager vs lazy loading in an object-relational mapper?

Module 3:

Sub-resources and Resource Expansion

```
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!"

"Our web server crashed because the IIS logs filled up the C: drive!
We never had this problem with PHP..."

```
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" }
  },
  "username" : "ironman",
  "name" : "Tony Stark"









}
```

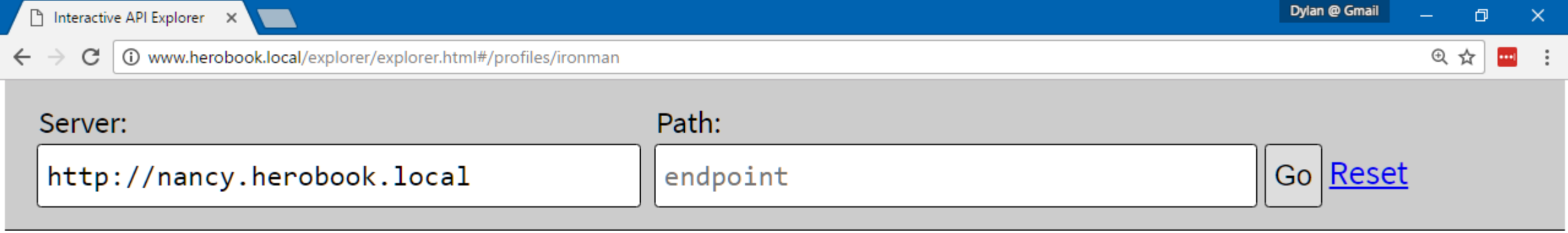
```
GET /profiles/ironman?expand=updates HTTP/1.1
200 OK
Content-Type: application/json
{
  "_links": {
    "self"    : { "href": "/profiles/ironman" },
    "friends" : { "href": "/profiles/ironman/friends" },
    "photos"  : { "href": "/profiles/ironman/photos" },
    "updates" : { "href": "/profiles/ironman/updates"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

Module 3:

Demo:

Resource Expansion


Interactive API Explorer
Dylan @ Gmail
www.herobook.local/explorer/explorer.html#/profiles/ironman
Server:
http://nancy.herobook.local
Path:
endpoint
Go
Reset
Ready.

Interactive API Explorer

Dylan @ Gmail

www.herobook.local/explorer/explorer.html#/profiles/ironman

Server:

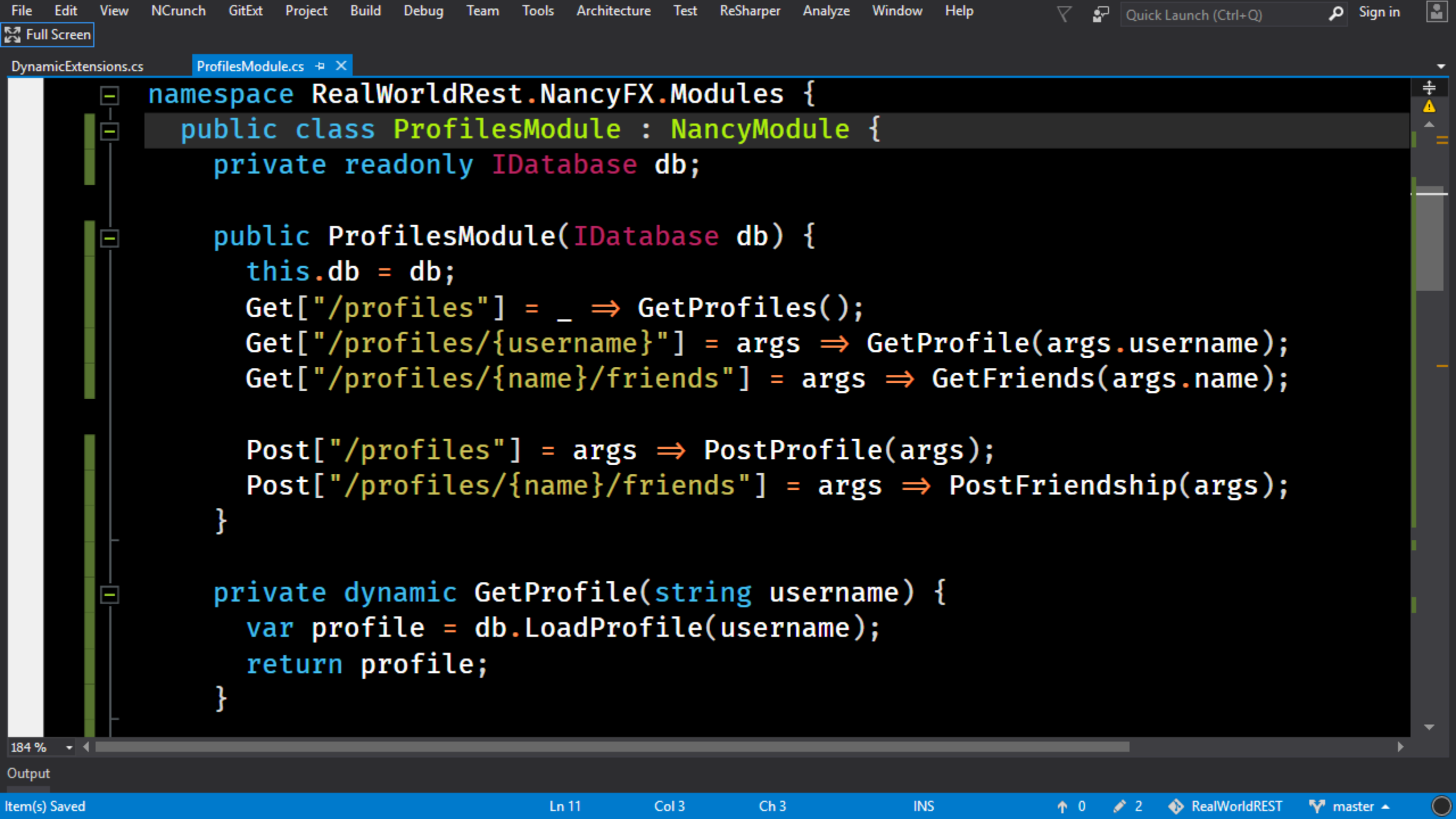
http://nancy.herobook.local

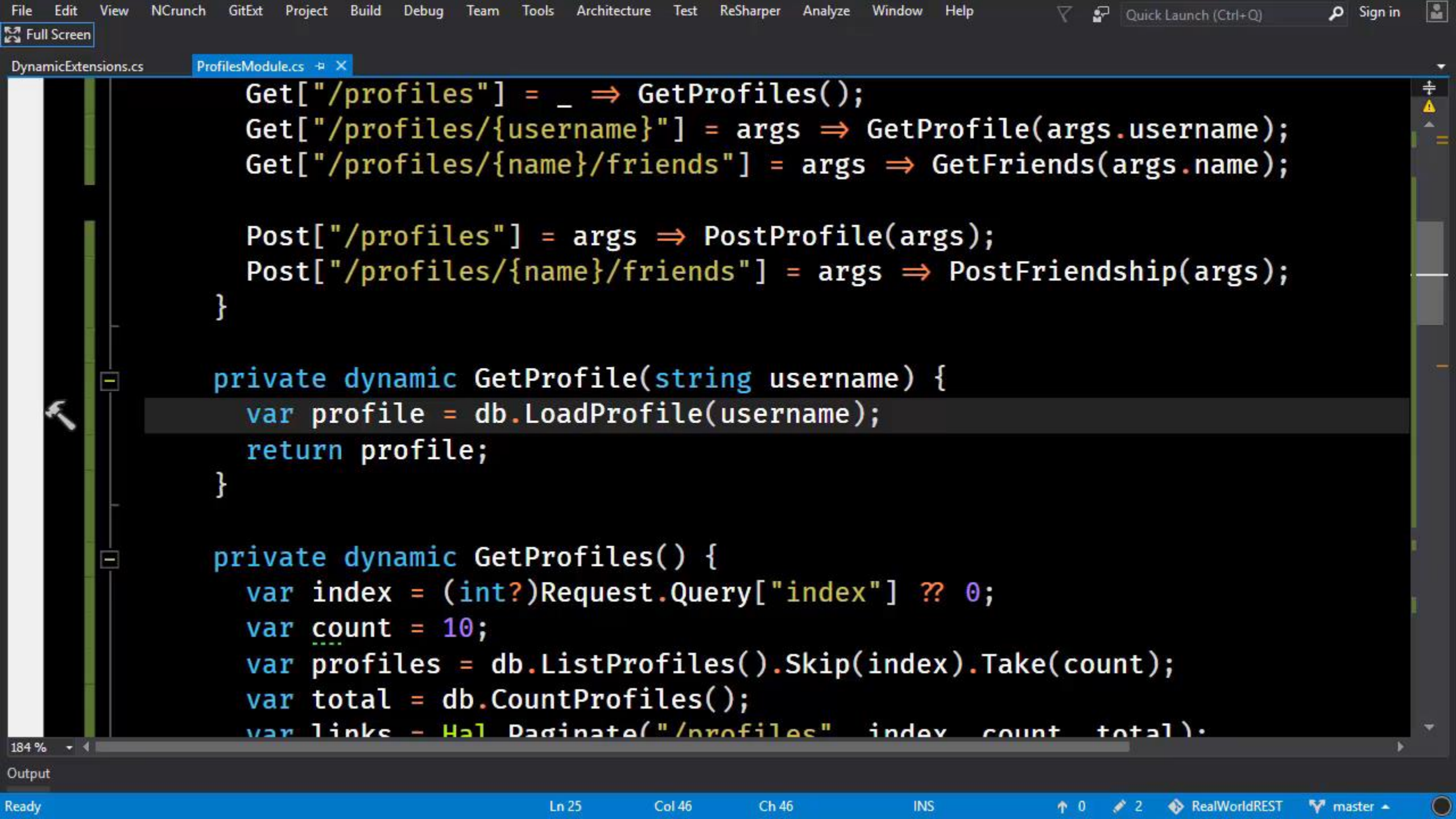
Path:

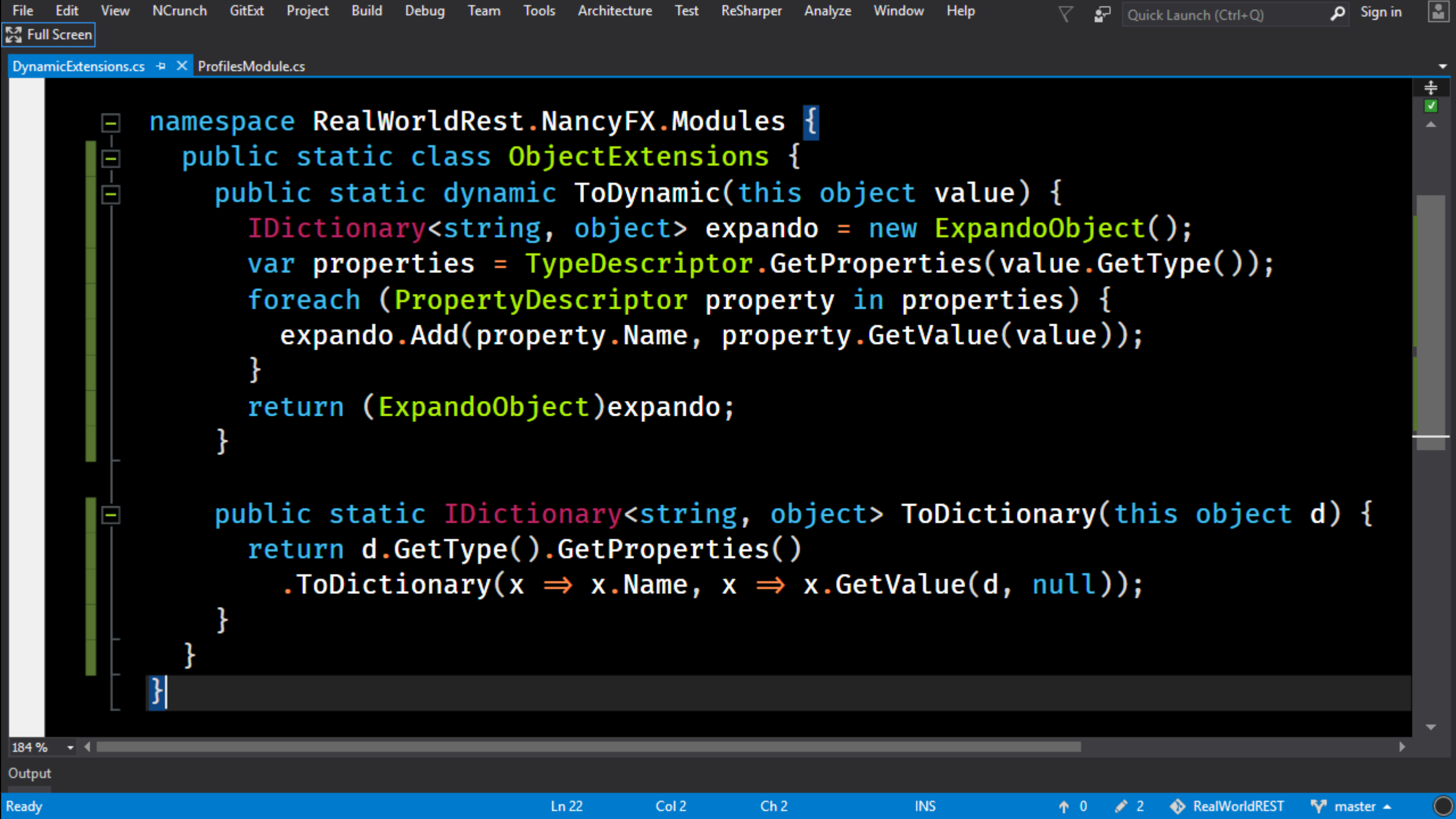
/profiles/ironman

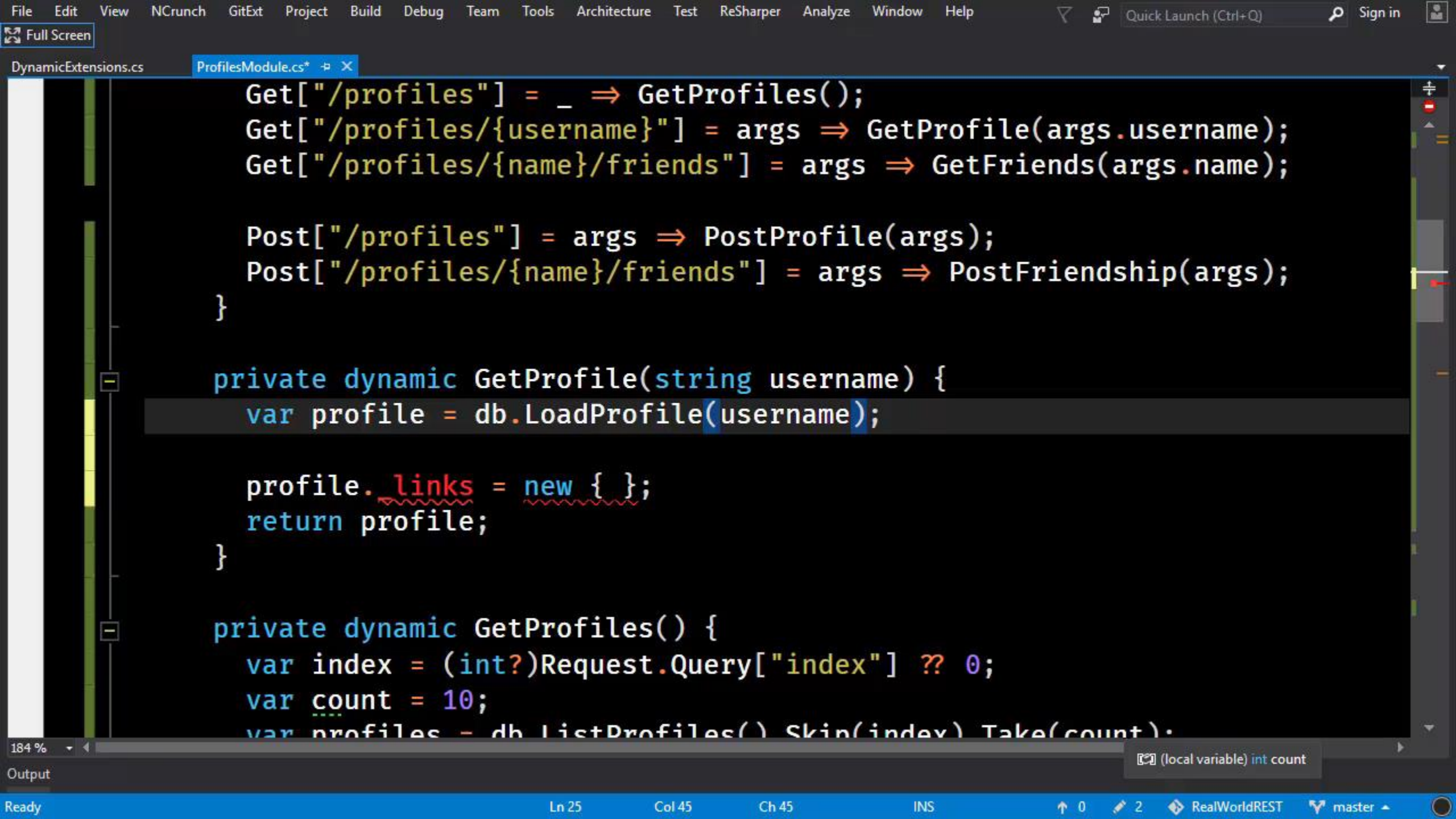
Go Reset

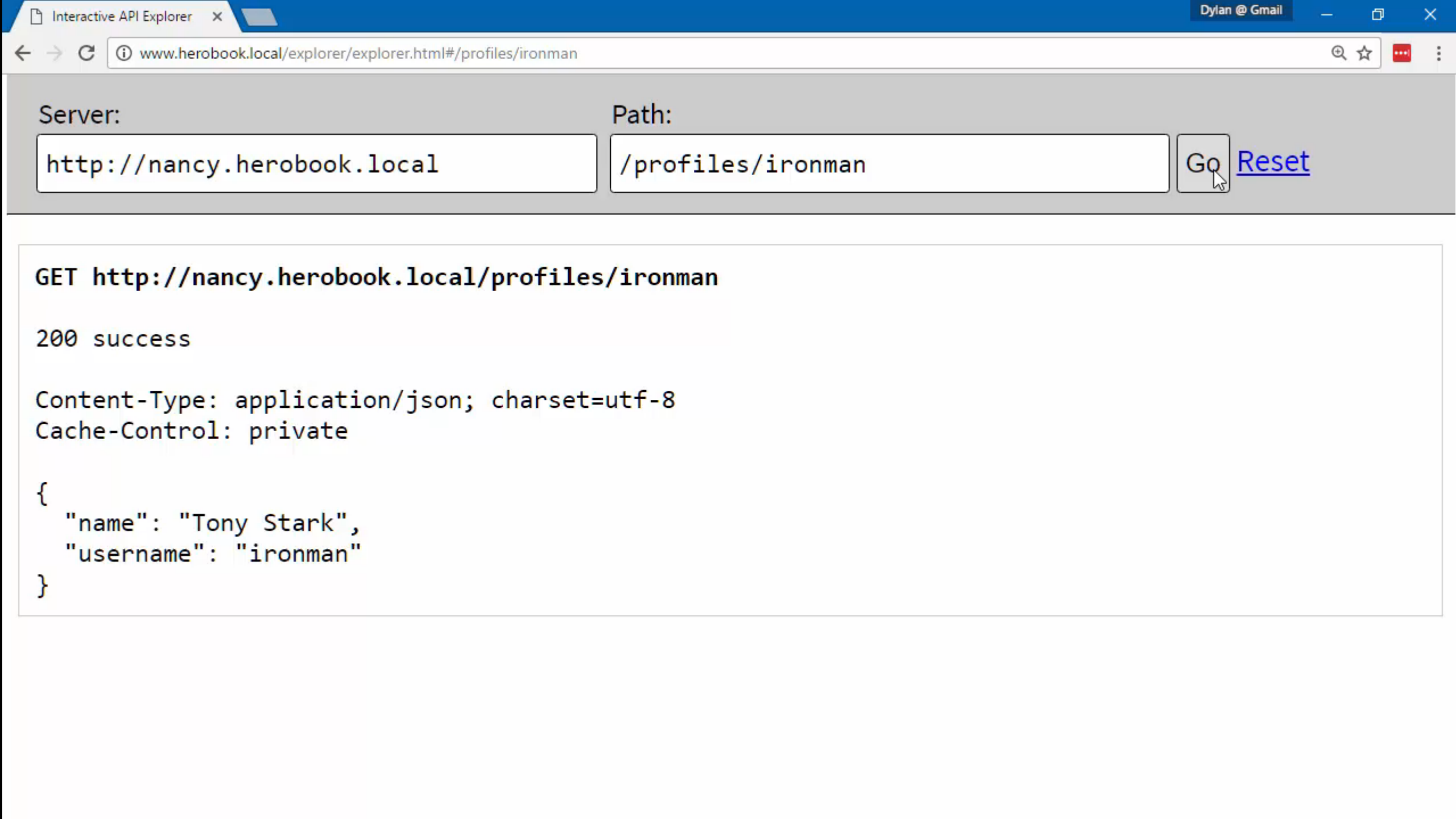
Ready.

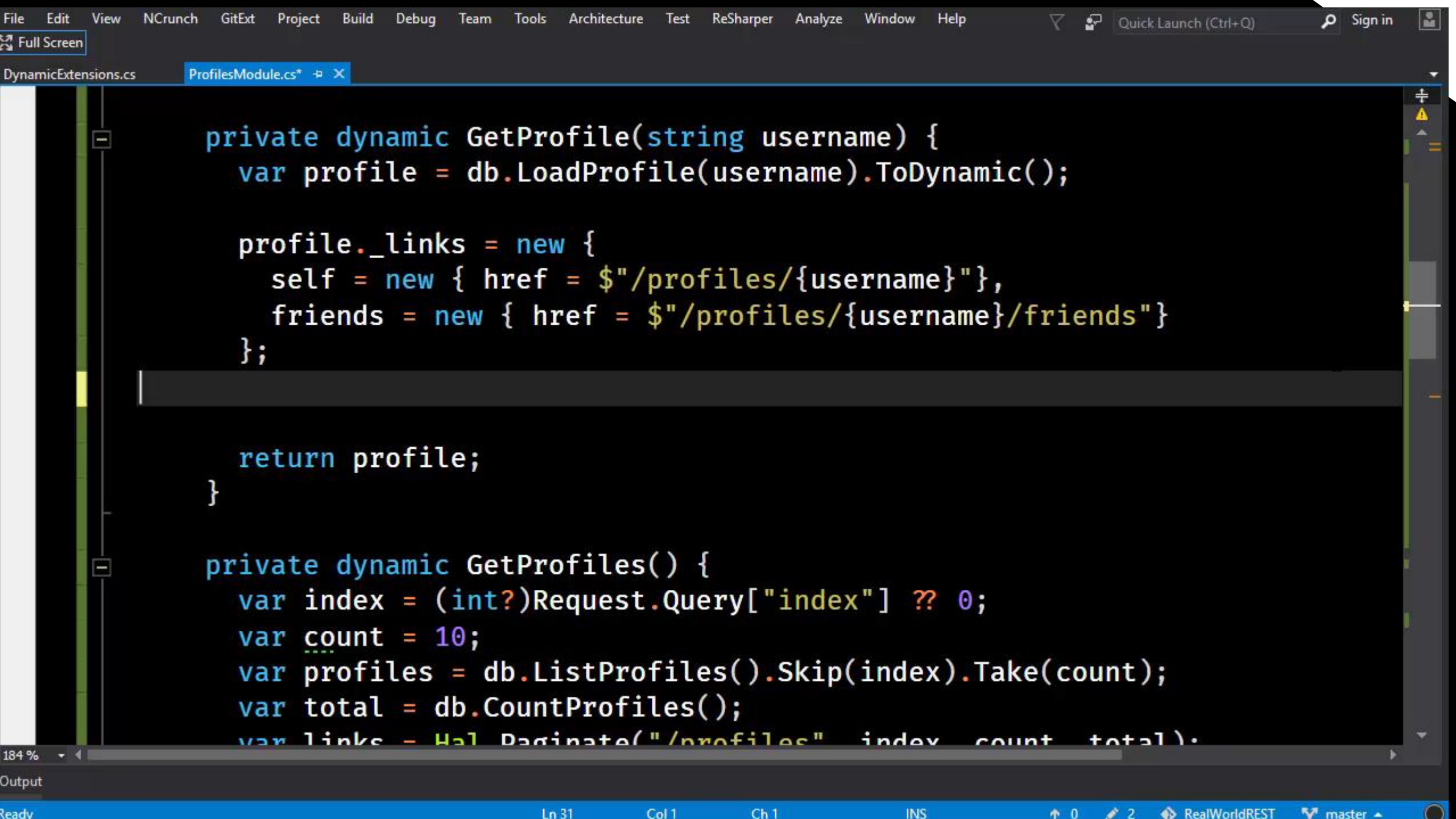

File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesModule.cs
namespace RealWorldRest.NancyFX.Modules {
public class ProfilesModule : NancyModule {
private readonly IDatabase 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
Item(s) Saved
Ln 11
Col 3
Ch 3
INS
0
2
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesModule.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();
184 %
Output
Ready
Ln 25
Col 46
Ch 46
INS
0
2
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesModule.cs
namespace RealWorldRest.NancyFX.Modules {
public static class ObjectExtensions {
public static dynamic ToDynamic(this object value) {
IDictionary<string, object> expando = new ExpandoObject();
var properties = TypeDescriptor.GetProperties(value.GetType());
foreach (PropertyDescriptor property in properties) {
expando.Add(property.Name, property.GetValue(value));
}
return (ExpandoObject)expando;
}
public static IDictionary<string, object> ToDictionary(this object d) {
return d.GetType().GetProperties()
.ToDictionary(x ⇒ x.Name, x ⇒ x.GetValue(d, null));
}
}
}
184 %
Output
Ready
Ln 22
Col 2
Ch 2
INS
0
2
RealWorldREST
master


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesModule.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;
184 %
(local variable) int count
Output
Ready
Ln 25
Col 45
Ch 45
INS
0
2
RealWorldREST
master


Interactive API Explorer
Dylan @ Gmail
www.herobook.local/explorer/explorer.html#/profiles/ironman
Server:
http://nancy.herobook.local
Path:
/profiles/ironman
Go
Reset
GET http://nancy.herobook.local/profiles/ironman
200 success
Content-Type: application/json; charset=utf-8
Cache-Control: private
{
"name": "Tony Stark",
"username": "ironman"
}


File Edit View NCrunch GitExt Project Build Debug Team Tools Architecture Test ReSharper Analyze Window Help
Quick Launch (Ctrl+Q)
Sign in
Full Screen
DynamicExtensions.cs
ProfilesModule.cs*
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();
184 %
Output
Ready
Ln 31
Col 1
Ch 1
INS
RealWorldREST
master

Module 3:

Workshop Exercise:

Resource Expansion

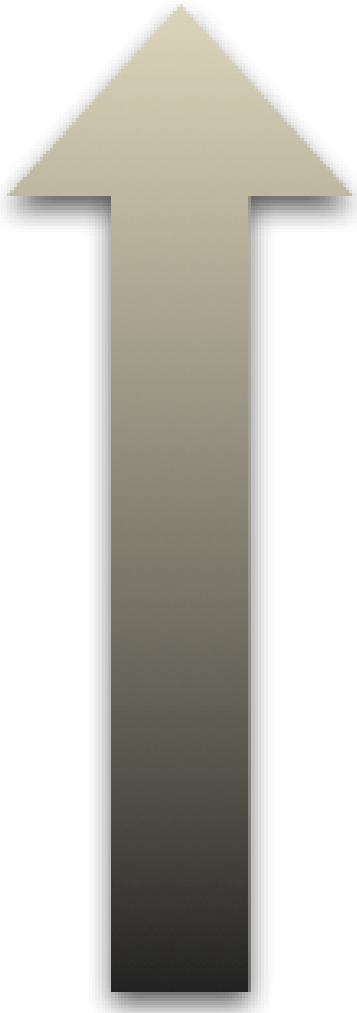
Module 4:

Going beyond GET:

Supporting Actions with Hypermedia

The Richardson Maturity Model

Glory of REST

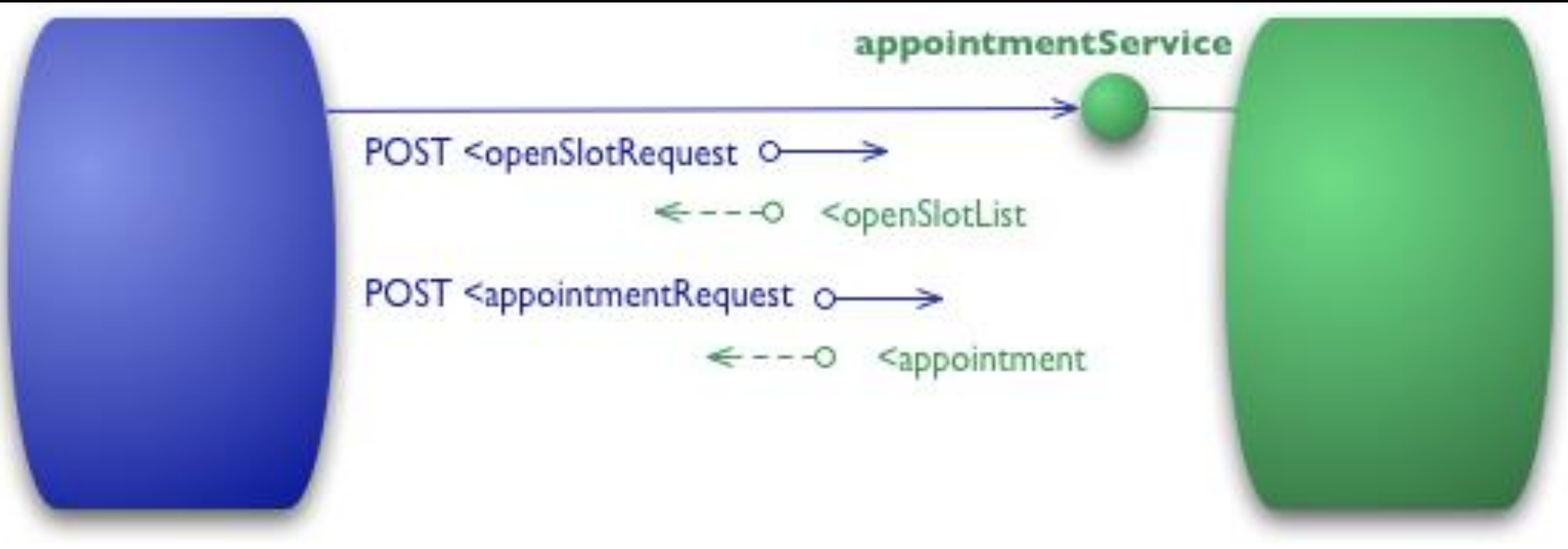


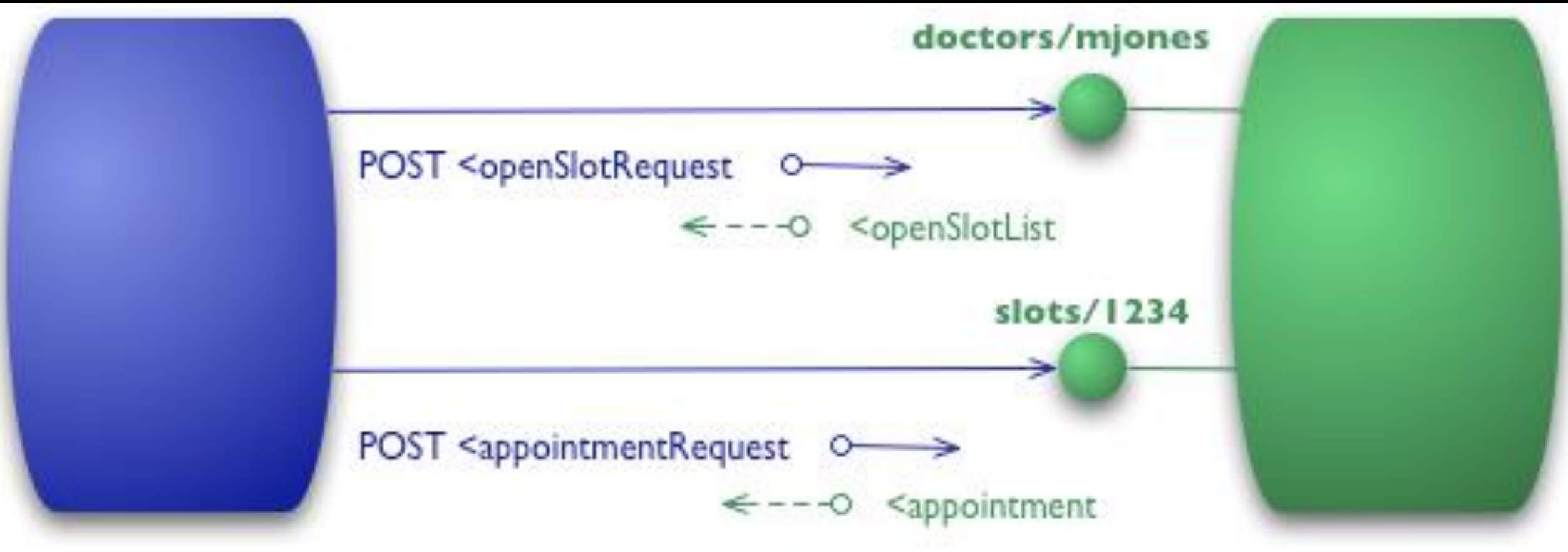
Level 3: Hypermedia Controls

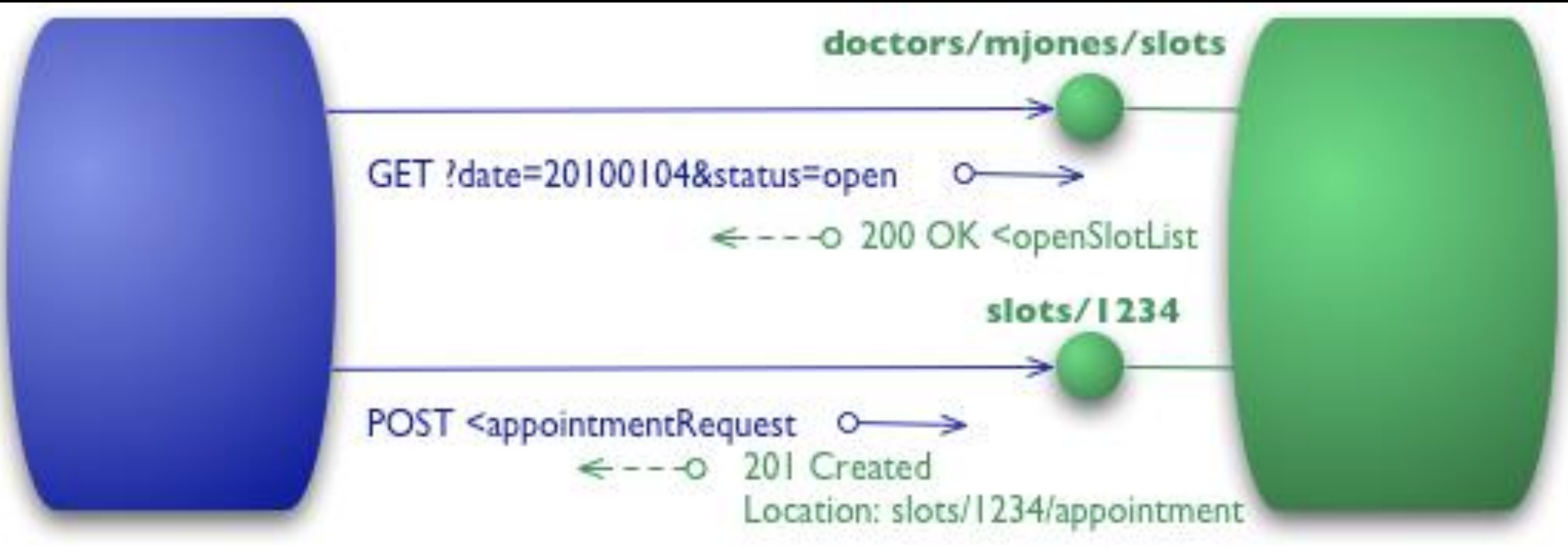
Level 2: HTTP Verbs

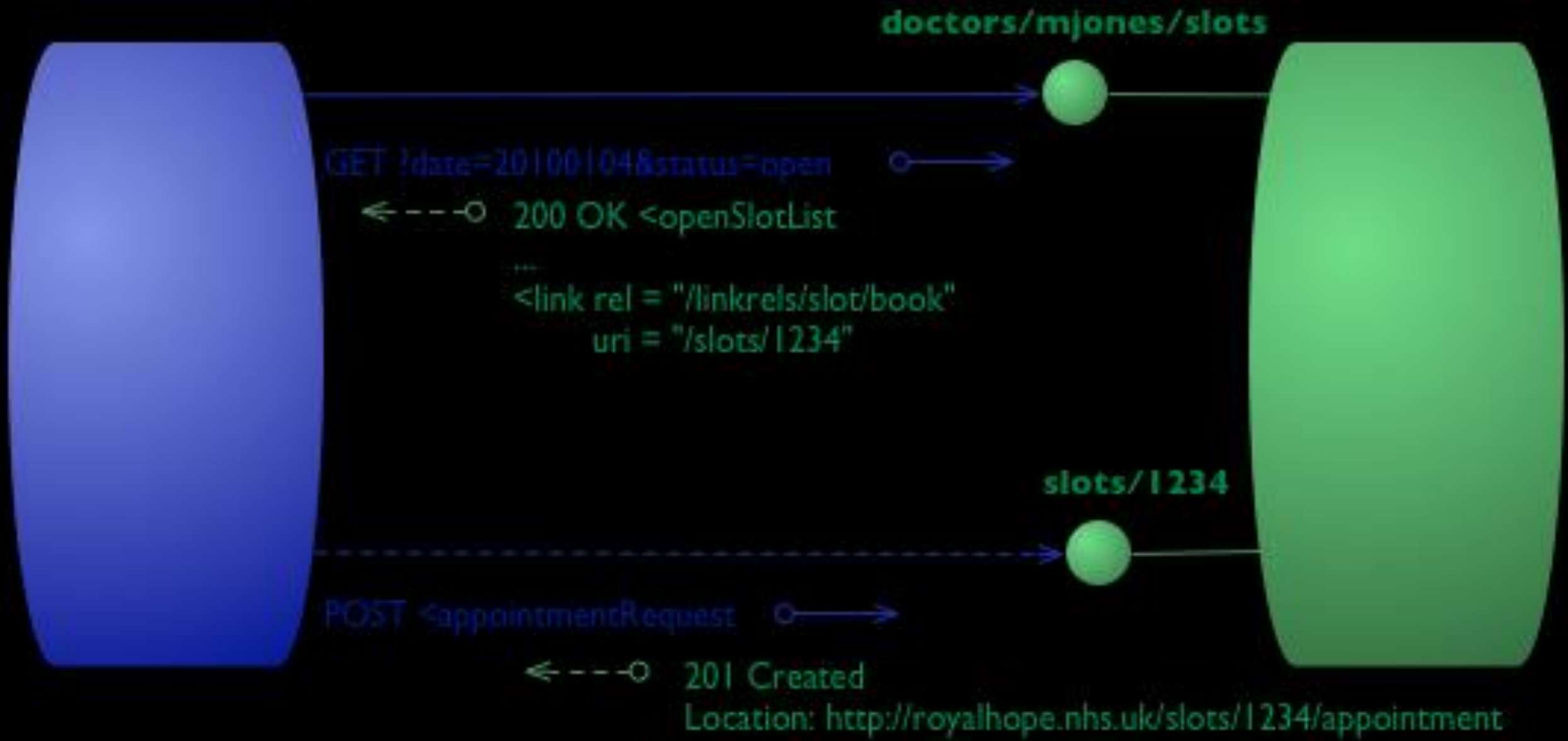
Level 1: Resources

Level 0: The Swamp of POX


appointmentService
POST <openSlotRequest
<openSlotList
POST <appointmentRequest
<appointment


doctors/mjones
POST <openSlotRequest
<openSlotList
slots/1234
POST <appointmentRequest
<appointment


doctors/mjones/slots
GET ?date=20100104&status=open
200 OK <openSlotList
slots/1234
POST <appointmentRequest
201 Created
Location: slots/1234/appointment


doctors/mjones/slots
GET ?date=20100104&status=open
200 OK <openSlotList
...
<link rel = "/linkrels/slot/book"
uri = "/slots/1234"
slots/1234
POST <appointmentRequest
201 Created
Location: http://royalhope.nhs.uk/slots/1234/appointment

```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
{

  "items" : [
    { "username": "ironman", "name": "Tony Stark" },
    { "username": "blackwidow", "name": "Natasha Romanoff" }
  ]
}
```

```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/hal+json
{
  "_actions" : {
    "create" : {
      "name": "Create a Profile",
      "type": "application/json",
      "href" : "/profiles",
      "method": "POST"
    }
  },
  "items" : [
    { "username": "ironman", "name": "Tony Stark" },
    { "username": "blackwidow", "name": "Natasha Romanoff" }
  ]
}
```

```
GET /profiles HTTP/1.1
200 OK
Content-Type: application/vnd.herobook.hal+json
{
  "_actions" : {
    "create" : {
      "name": "Create a Profile",
      "type": "application/json",
      "href" : "/profiles",
      "method": "POST"
    }
  },
  "items" : [
    { "username": "ironman", "name": "Tony Stark" },
    { "username": "blackwidow", "name": "Natasha Romanoff" }
  ]
}
```

Module 4:

Walkthrough:

Supporting Actions with Hypermedia

Module 4:

Workshop Exercise:

Supporting Actions with Hypermedia

Module 5:

Implementing

HTTP PATCH

```
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 location?

"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 Method for HTTP" - http://tools.ietf.org/html/rfc5789

```
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 Method for HTTP" - http://tools.ietf.org/html/rfc5789

```
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

Module 5:

Walkthrough:

Introducing HTTP PATCH

Module 5:

Workshop exercise:

Introducing HTTP PATCH

DEMO:

Tools, Tips

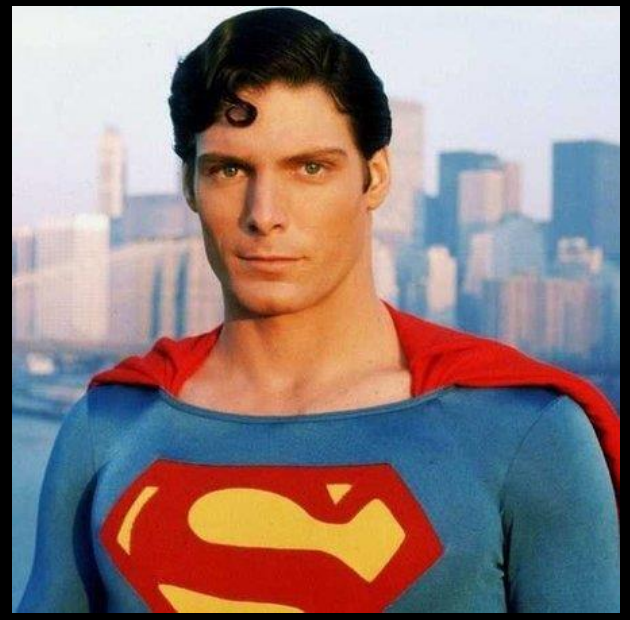
and Tricks

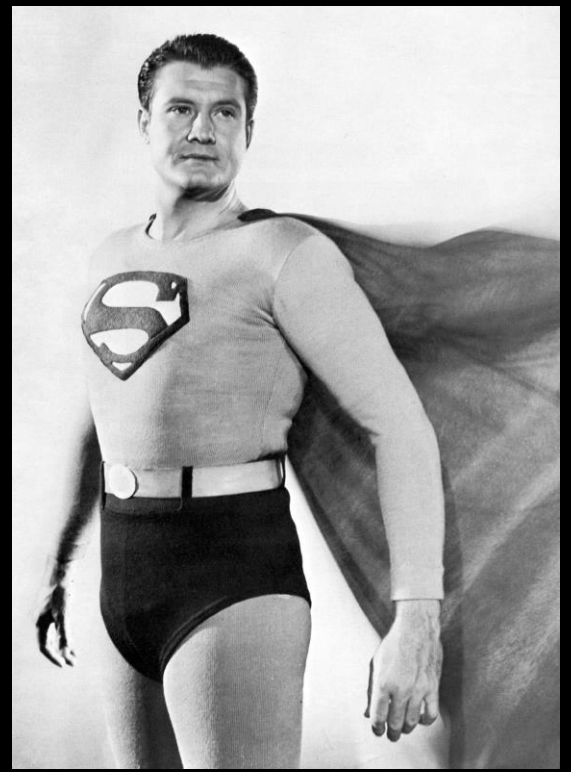
LUNCH

Module 6:

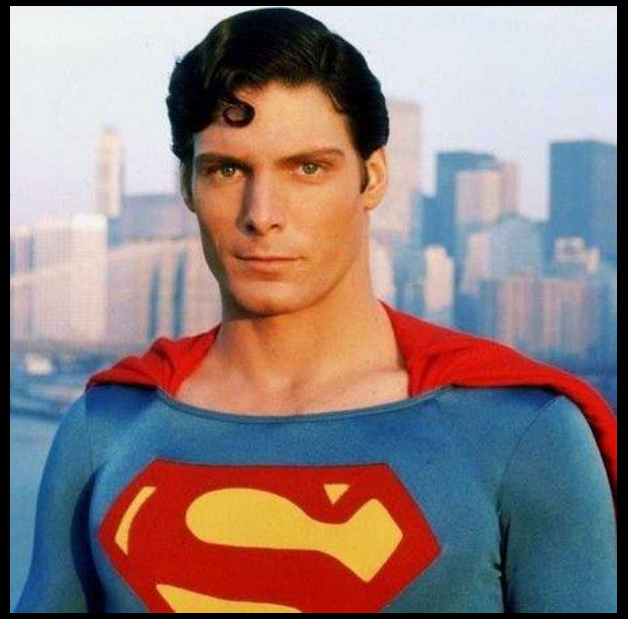
Content Negotiation

Resources and Representations

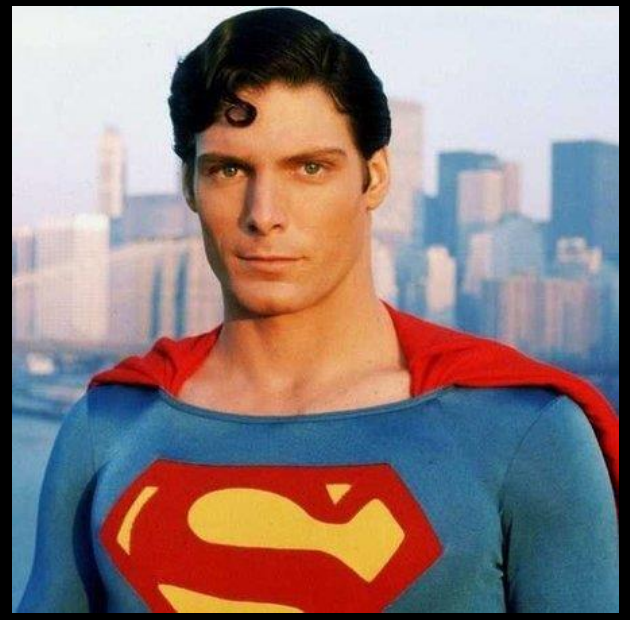




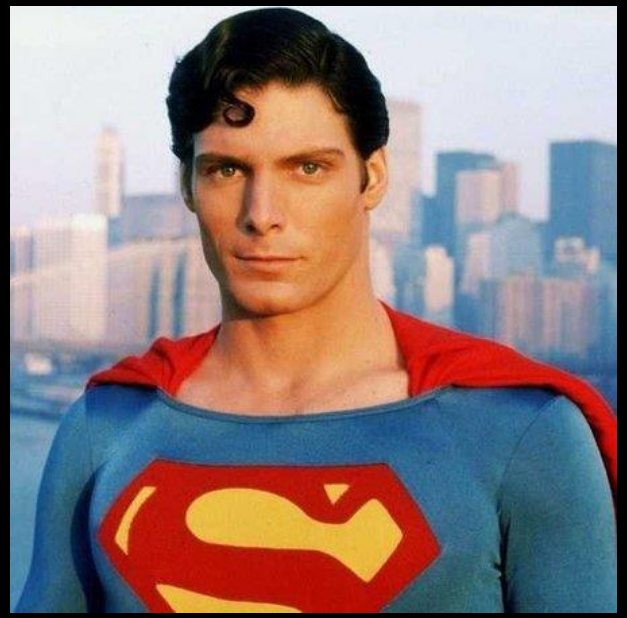
'concept'



Representation
(image/jpeg)



Resource
("this picture of Superman")



Representation
(image/png)

Module 6:

Walkthrough:

Resources and Representations

Module 6:

Coding Exercise:

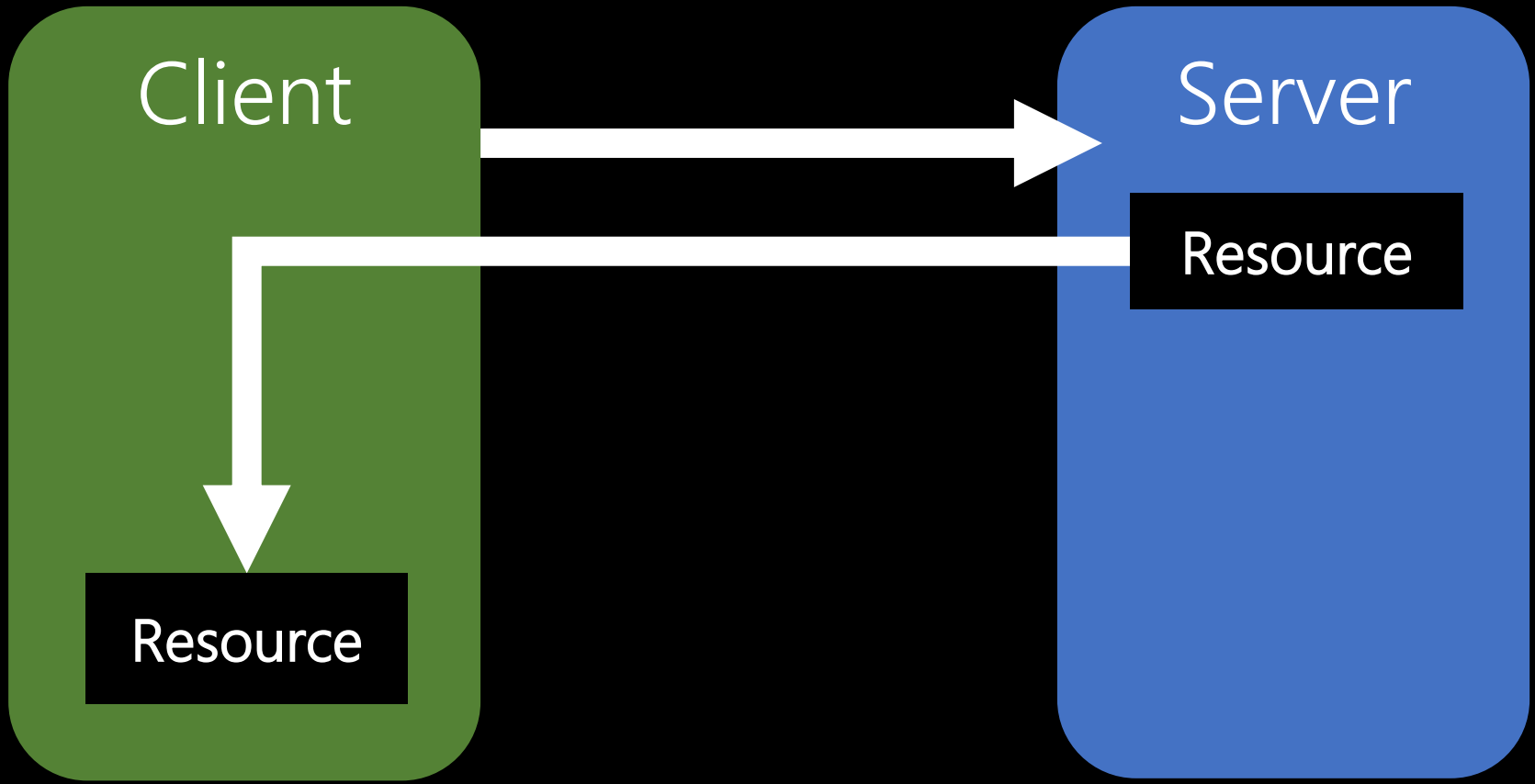
Resources and Representations

Module 7:

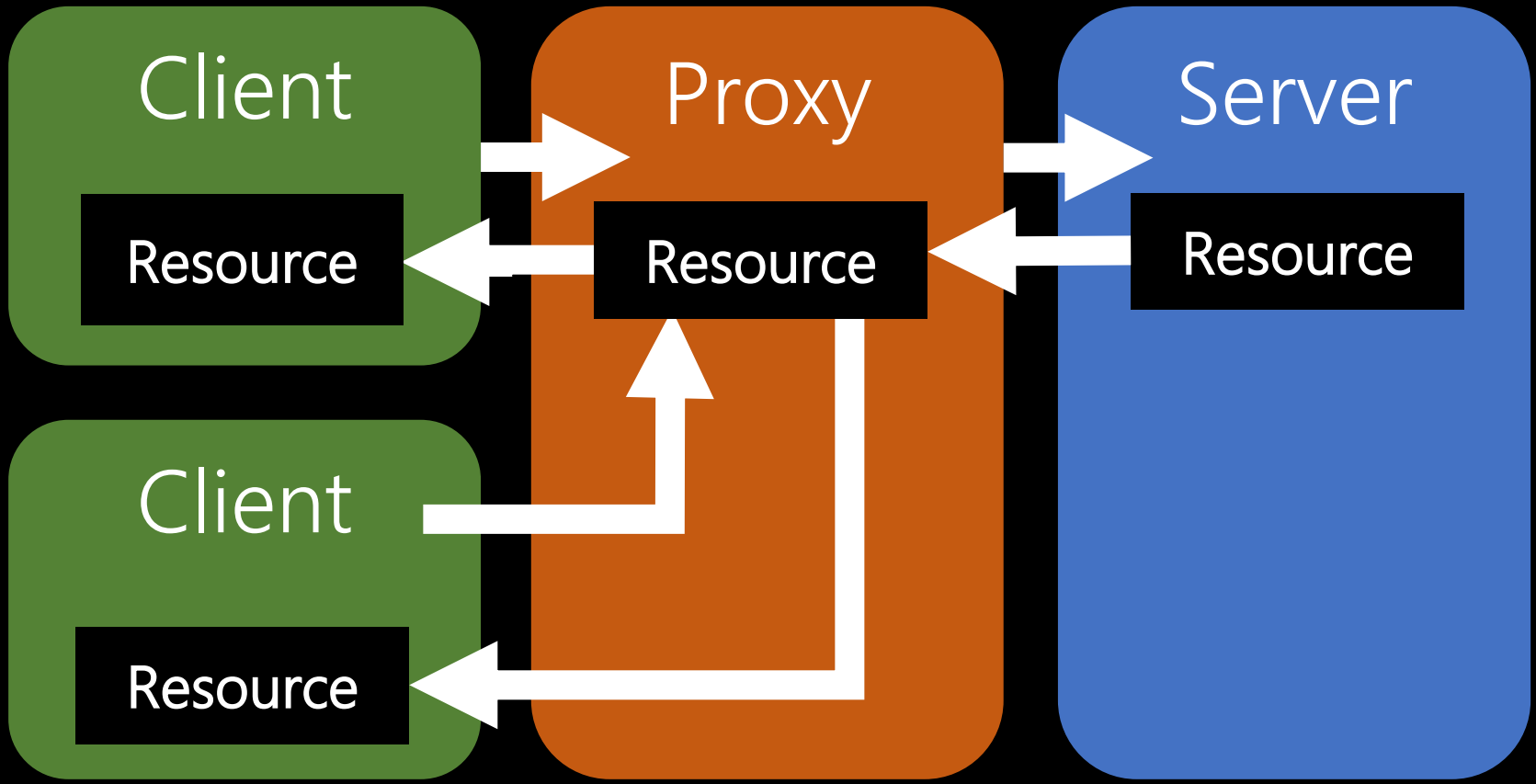
Scaling REST APIs:

Caching and Layering

3. Cacheable



4. Layered System



```
GET /profiles/ironman HTTP/1.1
Accept: application/json


200 OK
Content-Type: application/json



{
  "name" : "Tony Stark",
  "username: " "ironman"
}
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json


200 OK
Content-Type: application/json
Date: Tue, 22 May 2017 12:45:22 GMT


{
  "name" : "Tony Stark",
  "username: " "ironman"
}
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json


200 OK
Content-Type: application/json
Date: Tue, 22 May 2017 12:45:22 GMT
Expires: Tue, 22 May 2017 13:45:22 GMT

{
  "name" : "Tony Stark",
  "username: " "ironman"
}
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
If-Modified-Since: Tue, 22 May 2017 12:45:22 GMT
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
If-Modified-Since: Tue, 22 May 2017 12:45:22 GMT

304 Not Modified
Date: Tue, 22 May 2017 12:55:22 GMT
Expires: Tue, 22 May 2017 13:45:22 GMT
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json


200 OK
Content-Type: application/json

{
  "name" : "Tony Stark",
  "username: " "ironman"
}
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json


200 OK
Content-Type: application/json
Etag: "771320498712341098"
{
  "name" : "Tony Stark",
  "username: " "ironman"
}
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
If-None-Match: "771320498712341098"
```

```
GET /profiles/ironman HTTP/1.1
Accept: application/json
If-None-Match: "771320498712341098"

304 Not Modified
```

```
PUT /profiles/ironman HTTP/1.1
Content-Type: application/json


{
  "name" : "Tony Stark",
  "username" : "ironman",
  "location" : "Tel Aviv, Israel"
}
```

```
PUT /profiles/ironman HTTP/1.1
Content-Type: application/json
If-None-Match: "1678267862877782"

{
  "name" : "Tony Stark",
  "username" : "ironman",
  "location" : "Tel Aviv, Israel"
}
```

```
PUT /profiles/ironman HTTP/1.1
Content-Type: application/json
If-None-Match: "1678267862877782"

{
  "name" : "Tony Stark",
  "username" : "ironman",
  "location" : "Tel Aviv, Israel"
}
```

409 Conflict

Module 7:

Walkthrough:

Caching and Layering

Module 7:

Workshop Exercise:

Caching and Layering

Module 8:

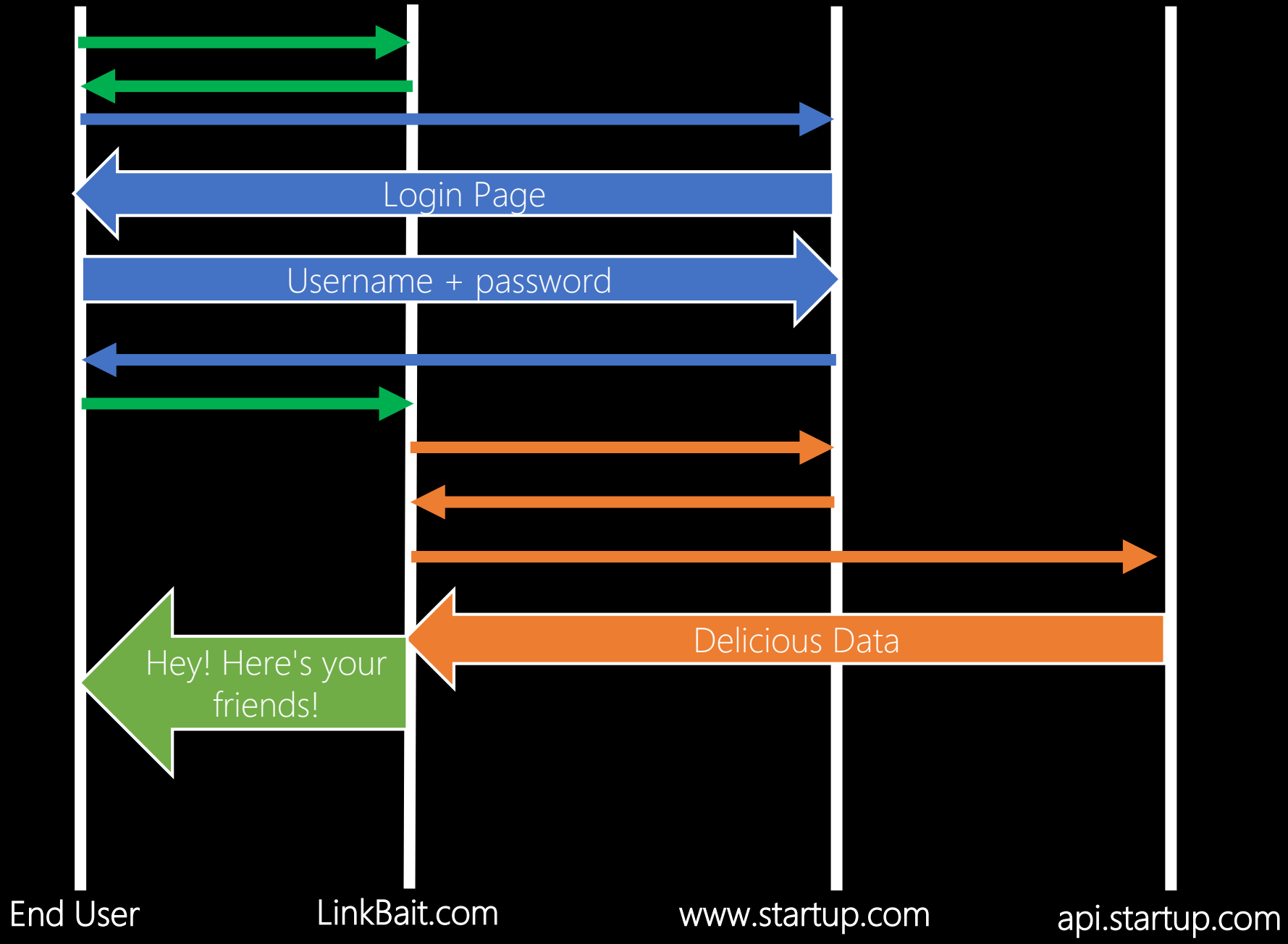
Security:

Authentication and Authorization

"We hired the CFO's teenage son to build us a mobile app"

"We just found out LinkBait.com is asking for our users' login details!"


157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223


Login Page
Username + password
Delicious Data
Hey! Here's your friends!
End User
LinkBait.com
www.startup.com
api.startup.com

Module 8:

Walkthrough:

Security and Authentication

Module 8:

Workshop:

Security and Authentication

Module 9:

Managing Changes:

API Versioning

"We need to expose forenames and surname separately"

"We're replacing hometown with an ISO3166 country code"

```
GET /profiles/ironman HTTP/1.1
200 OK
Content-Type: application/hal+json
{
  "name": { "firstname" : "Tony", "lastname" : "Stark" },
  "username" : "ironman",
  "height": 192,
  "weight": 85,
  "location" : { "lat": 34.02, "lon": -118.77 },
  "status": "Out saving the world. Again.",
  "hometown": {
    "city": "Malibu",
    "country": "US"
  },
  "email": "tony@stark.com",
  "website": "www.ironman.com",
  "birthdate": "1972-01-24",
  "last_modified": "2015-08-12T19:45:43.511Z"
}
```

"Hey! Your API broke our app!"

"Our website just stopped working"

"None of our reports work!"

“YOU BROKE THE INTERNET!”

Oops.

The easiest thing to do is

...never break anything!

...but : "REST is software design on

the scale of

decades"

```
GET /v2/profiles/ironman
Accept: application/json


200 OK
{ ... }
```

```
GET /profiles/ironman
Accept: application/vnd.myapi.v2+json
```

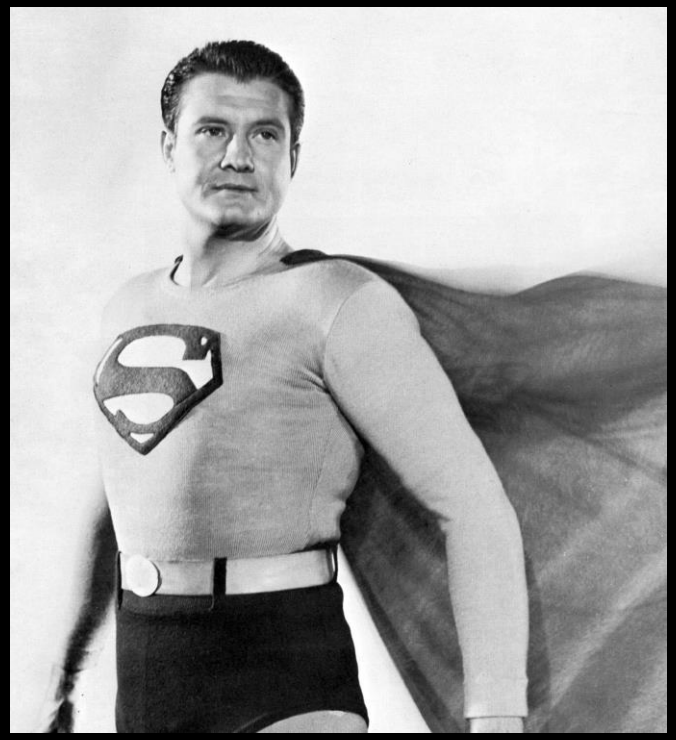
```
200 OK
{ ... }
```

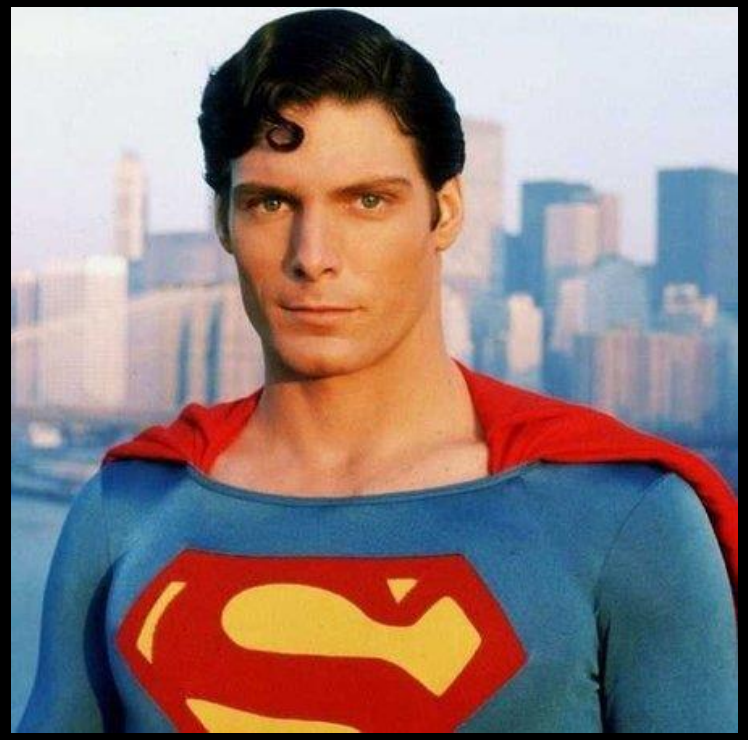
```
GET /profiles/ironman
Accept: application/json
X-MyApi-Version: 2


200 OK
{ ... }
```

"is version 2 of a profile the same *resource* as version 1?"

"Of course they are! They represent the same person!"

"Don't be daft. They're completely different."





"Of course they are! They represent the same person!"

```
GET /api/profiles
Accept: application/vnd.myapi.v2+json
```

"Don't be daft. They're completely different."

```
GET /api/v2/profiles
Accept: application/json
```

"Oh, forget ReST! Let's just use custom headers"

```
GET /api/profiles
X-MyApi-Version: 2
```

...and on the server?

Branch the codebase for separate versions?

One codebase, many versions?

```
[VersionedRoute("api/profiles/{name}", 2)]
[Route("api/v2/profiles/{name}")]
public Profile GetV2(string name) {

    /* write your program here! */

}
```

Module 9:

Walkthrough:

API Versioning

Module 9:

Workshop Exercise:

API Versioning

Module 10:

Testing and Monitoring Hypermedia APIs

http://memory-alpha.wikia.com/wiki/Force_of_Nature_(episode)?file=Warp_core_maintenance.jpg

NANCY

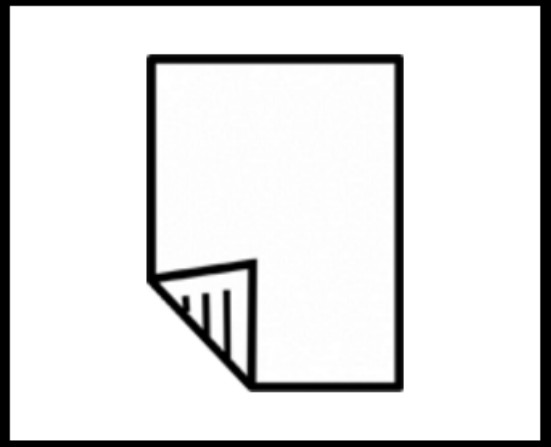
Dashboard

Information

Information about Nancy's current runtime configuration. A useful starting point for diagnosing a misbehaving site.

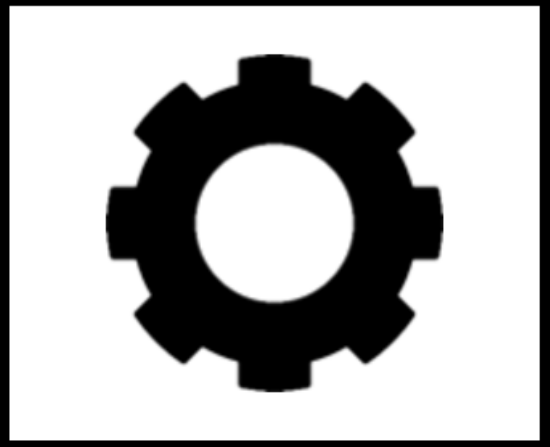
Interactive Diagnostics

Interactive diagnostics allow you to poke and prod at the guts of your Nancy site, from querying configuration, to clearing caches, through to executing parts of the pipeline and visualising the results.



Request Tracing

Provides tracing information about requests including request/response headers and a tracelog of each request.



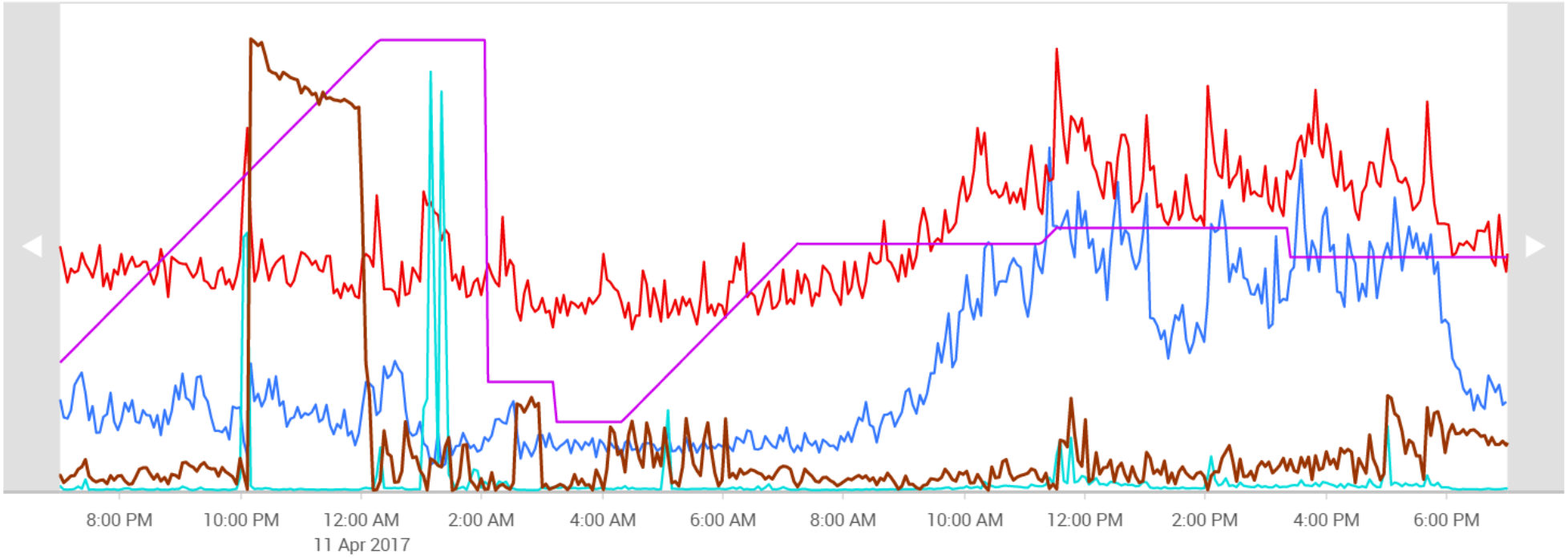
Settings

Configure some of Nancy's runtime static configuration.

Analysis Graph

Time range: 24 hours | Mon, Apr 10 19:00 | Tue, Apr 11 19:00 | COMPARE BASELINE

Last 1h | Last 6h | Last 24h | Last 7 days | Today | This week | Export (csv)



SQL Server: processor time [redacted]\(local)
User connections [redacted](local)
Buffer page life expectancy [redacted](local)
Latch wait time [redacted](local)
SQL Server: free memory [redacted]\(local)

Show:	SQL Server: processor time	For:	[redacted]spotmain.co	(local)
And:	User connections	For:	[redacted]spotmain.co	(local)
And:	Buffer page life expectancy	For:	*(As above)*	*(As above)*

Description | Statistics

FREE MEMORY

Module 10:

Walkthrough:

Testing and Monitoring Hypermedia APIs

Module 10:

Exercise:

Testing and Monitoring Hypermedia APIs

DISCUSSION

API Strategy

Questions?

Questions?

Thank you!

www.github.com/dylanbeattie/sela2017/

@dylanbeattie

dylan@dylanbeattie.net