

Real World **REST** &

Hands-On

Hypermedia

@dylanbeattie

DOTNEXT



Dylan Beattie

 @dylanbeattie

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

Slides & code:

www.github.com/dylanbeattie/dotnext/

я не говорю по-русски



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.



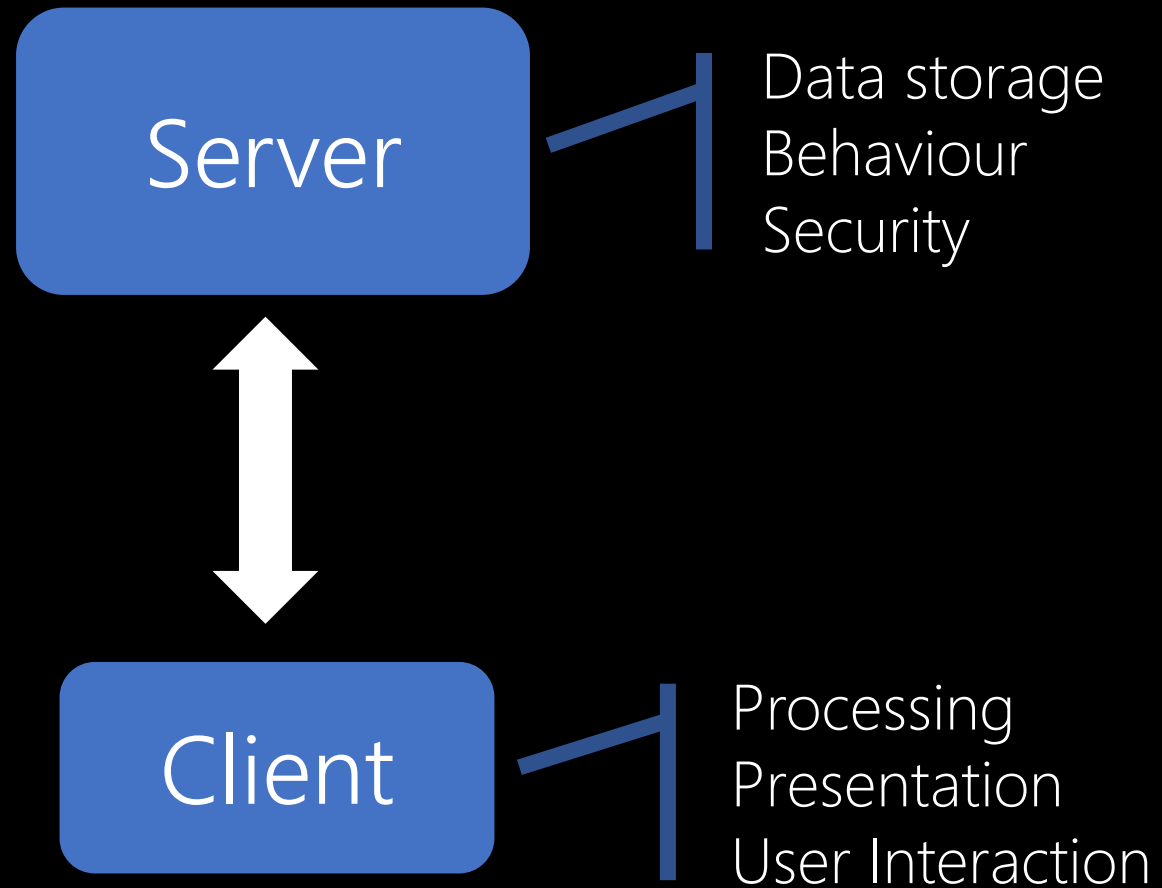
Many of the constraints are
directly opposed to
short-term efficiency."

Representational State Transfer (REST)

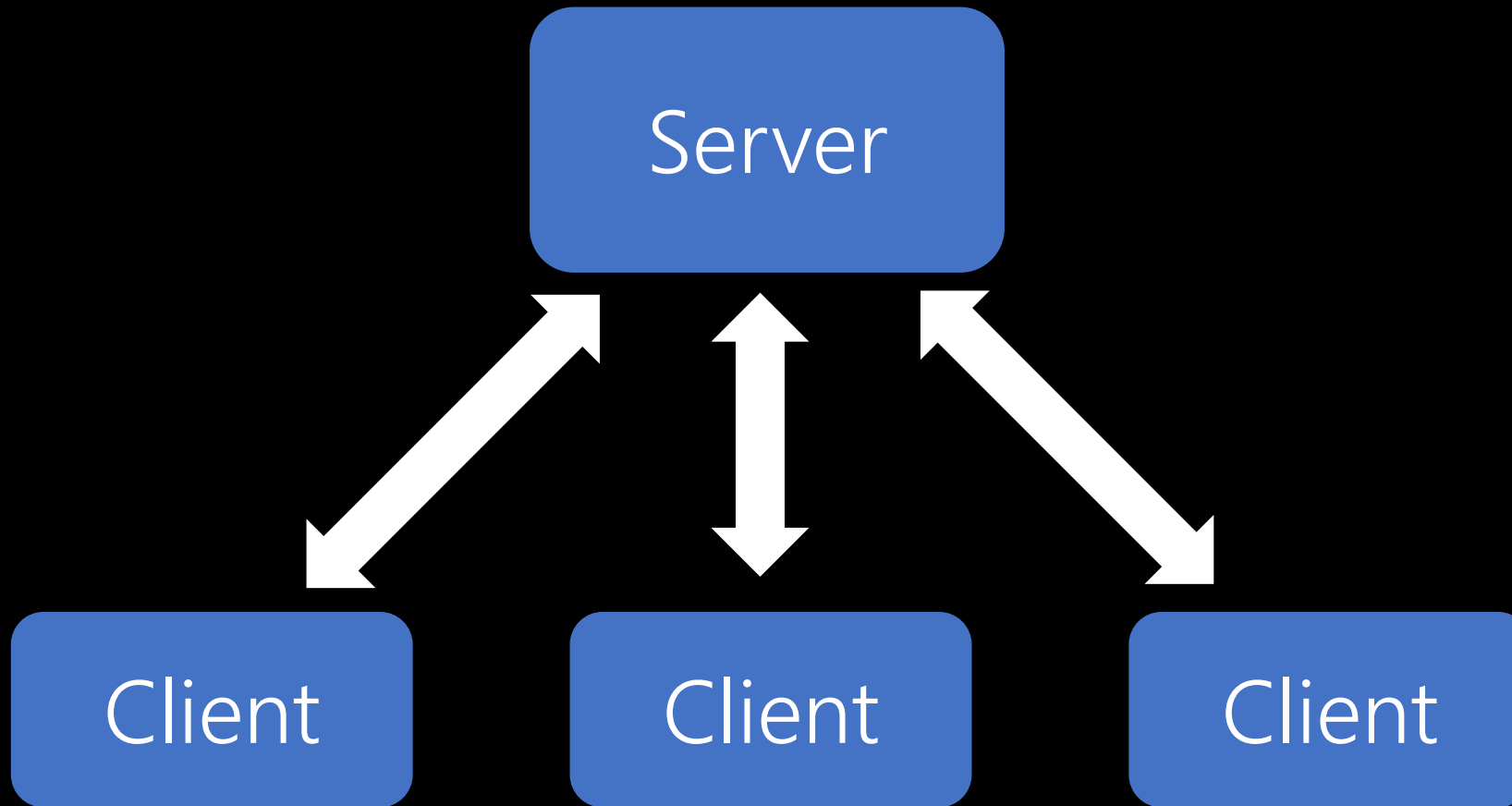
A coordinated set of
architectural constraints...

...intended to promote
software longevity and
independent evolution

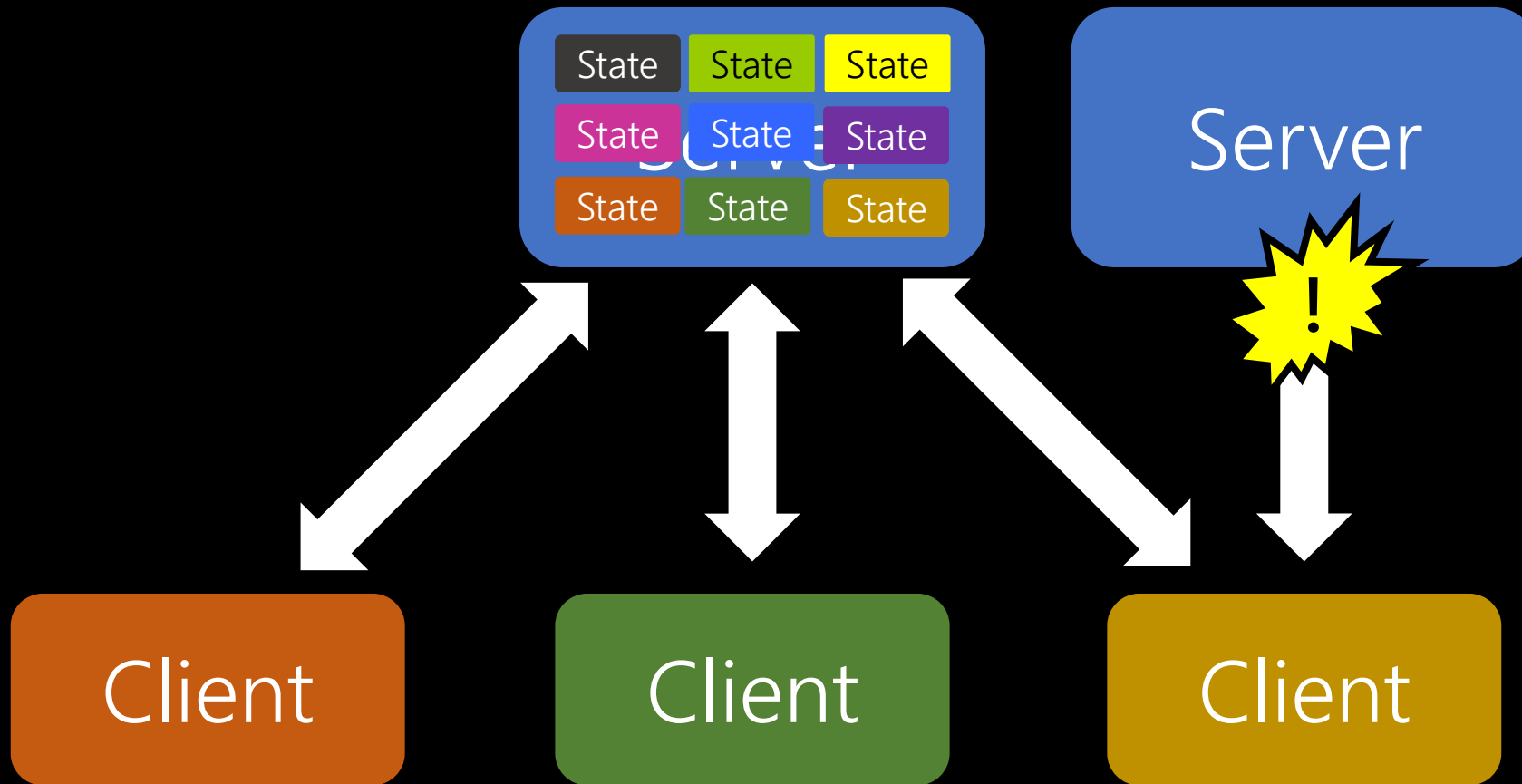
1. Client–Server



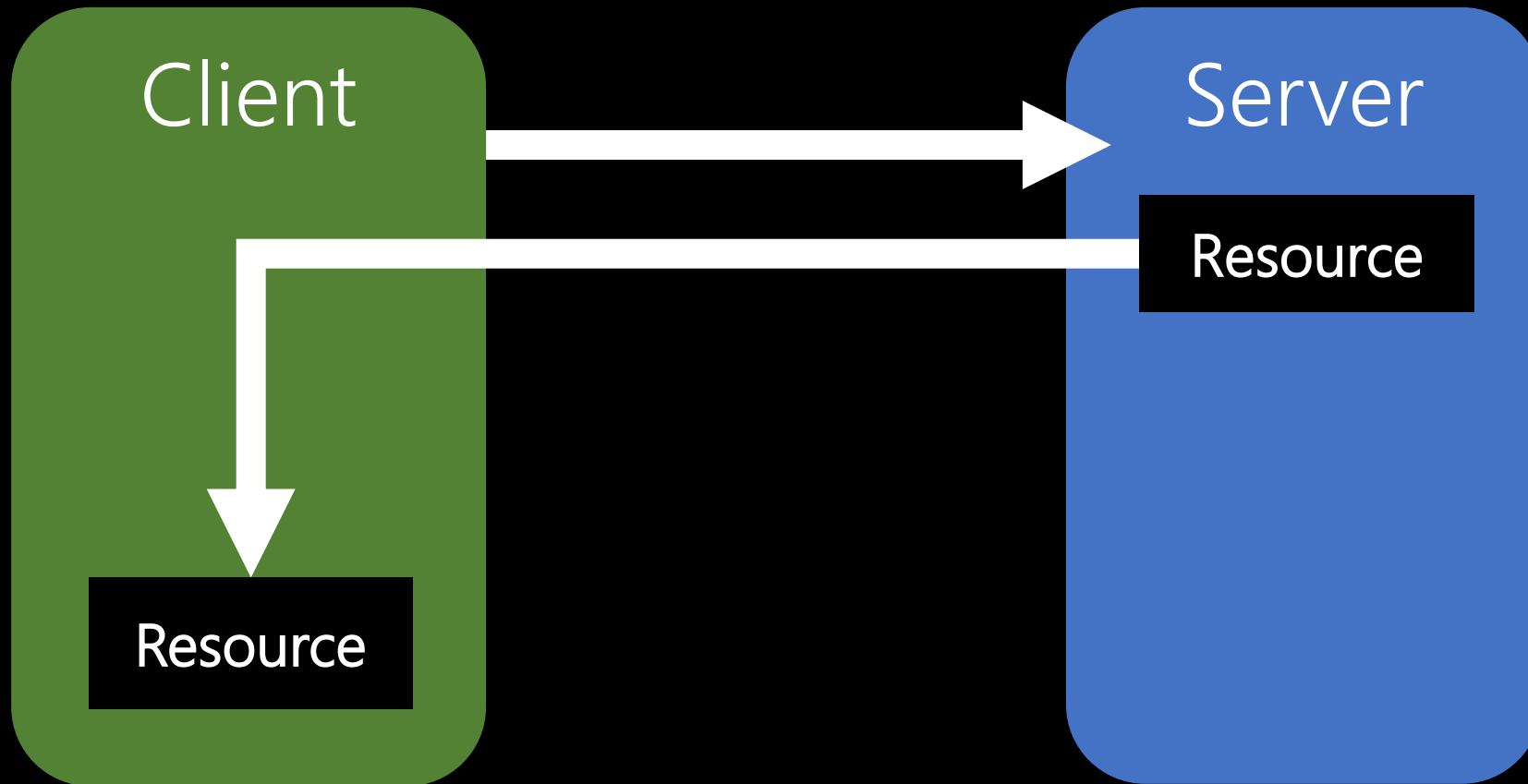
1. Client–Server



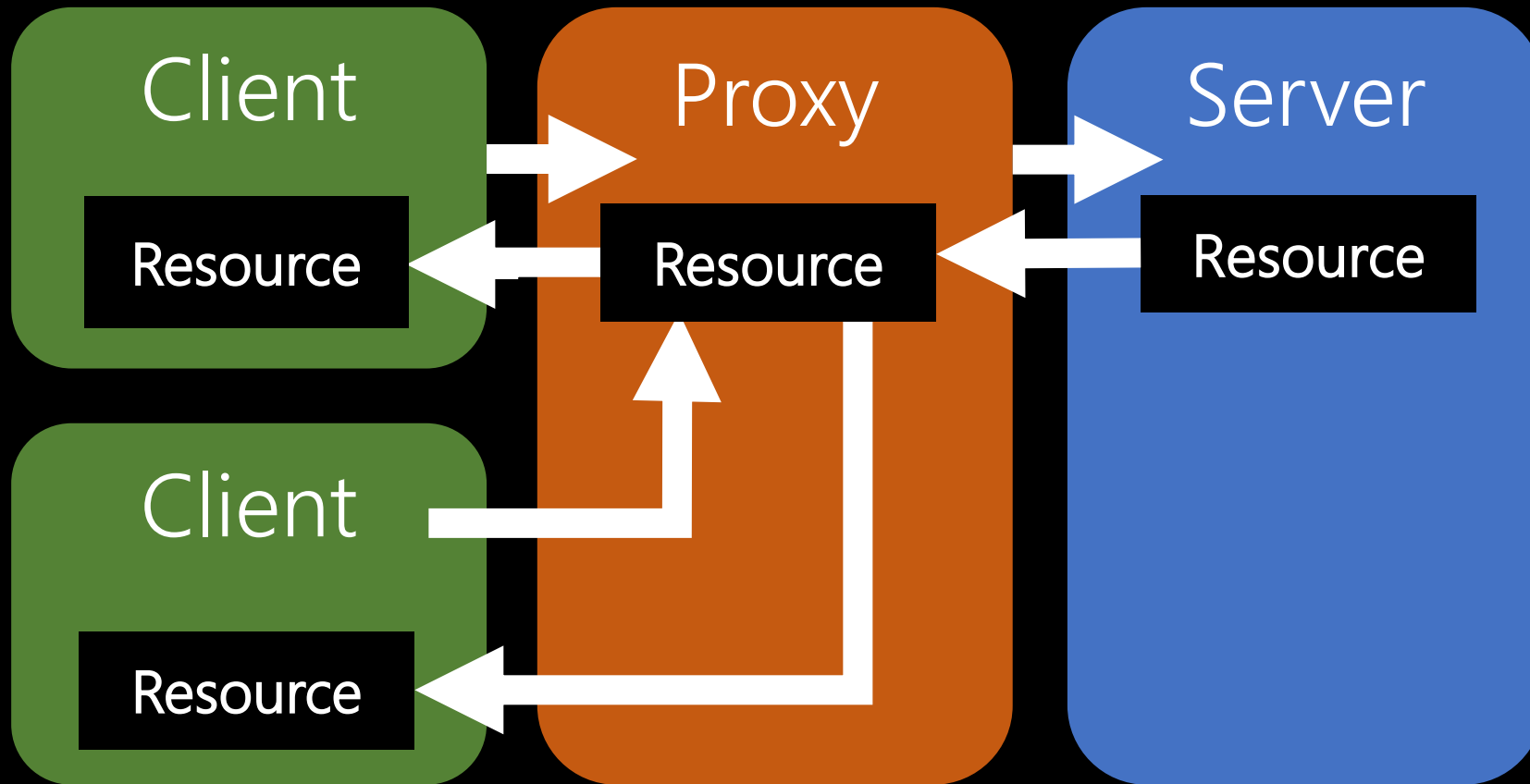
2. Stateless



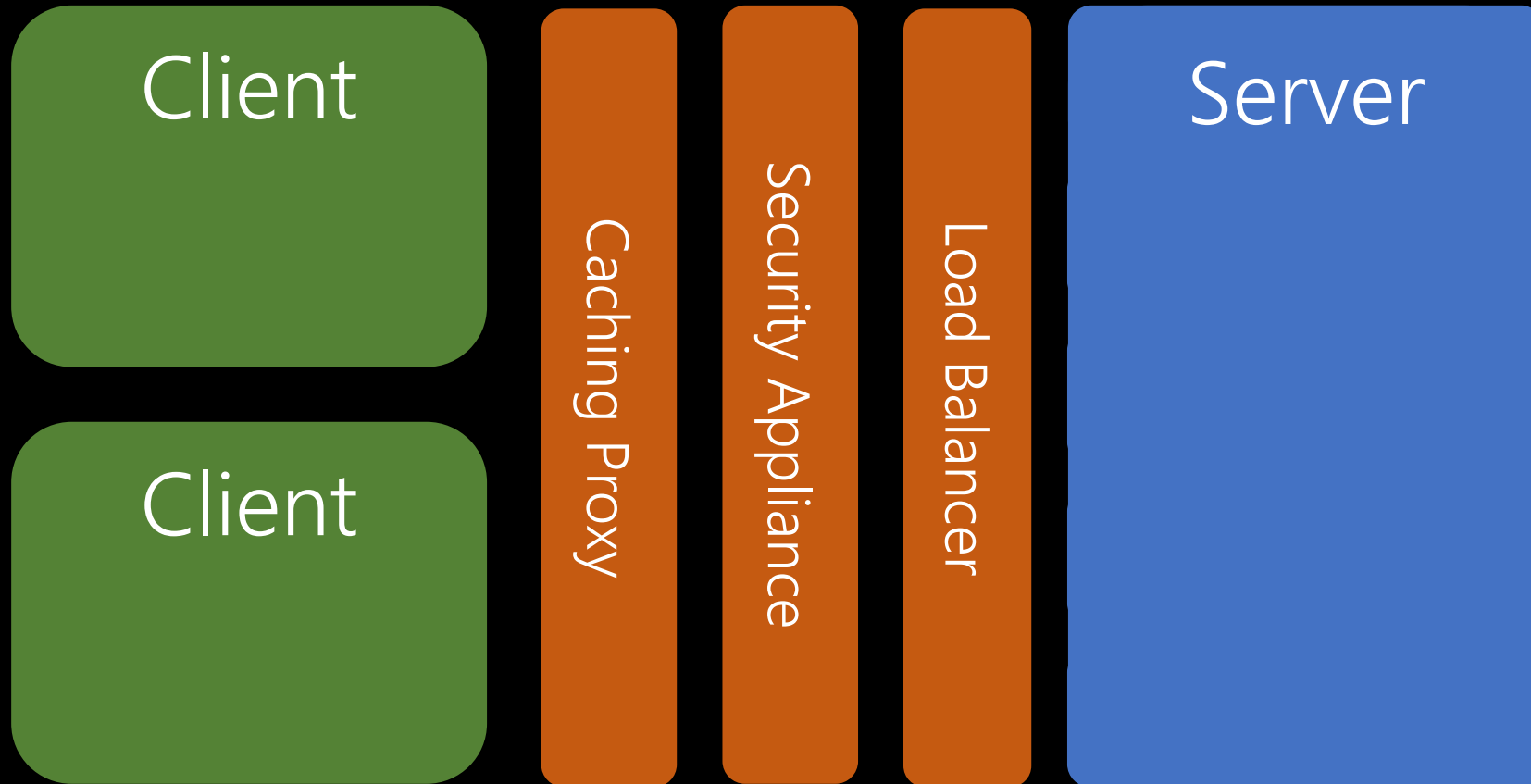
3. Cacheable



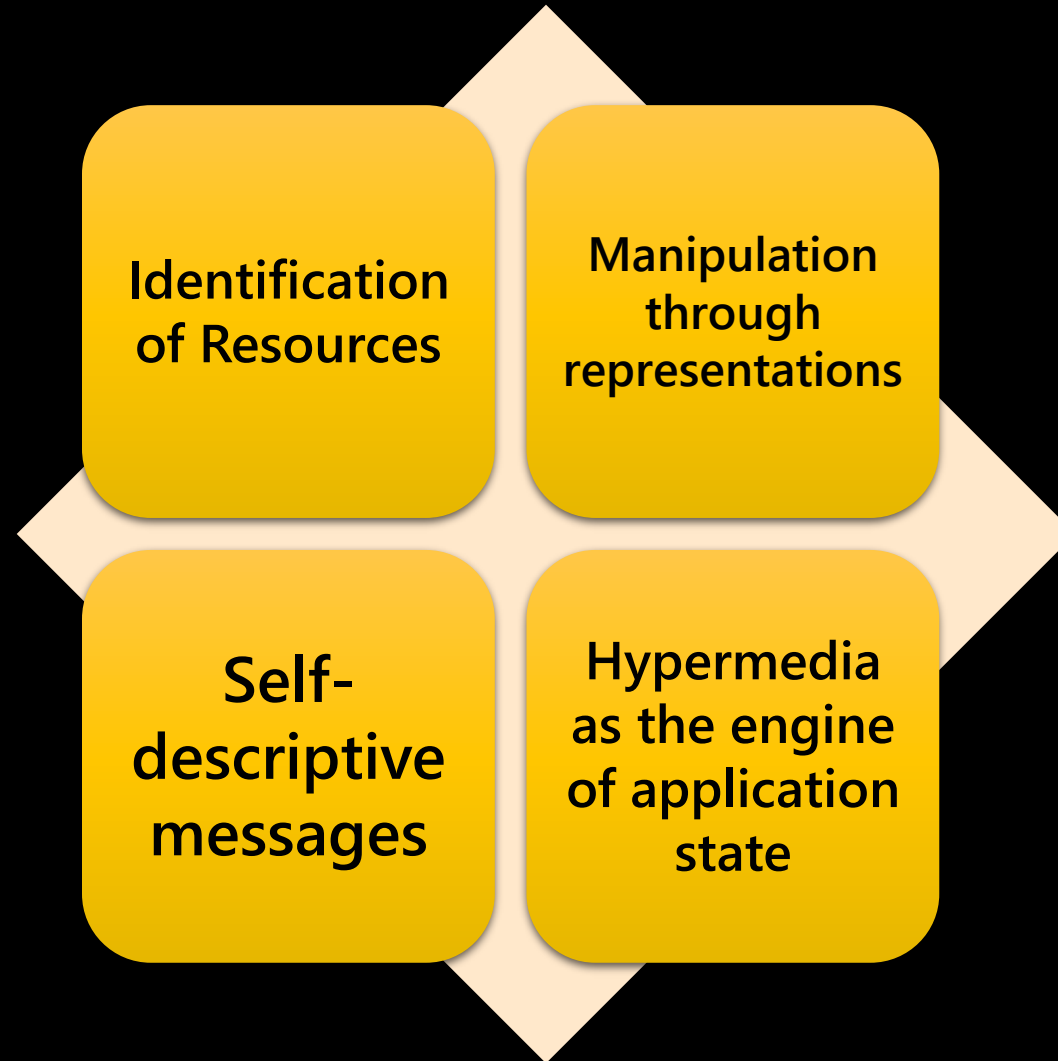
4. Layered System



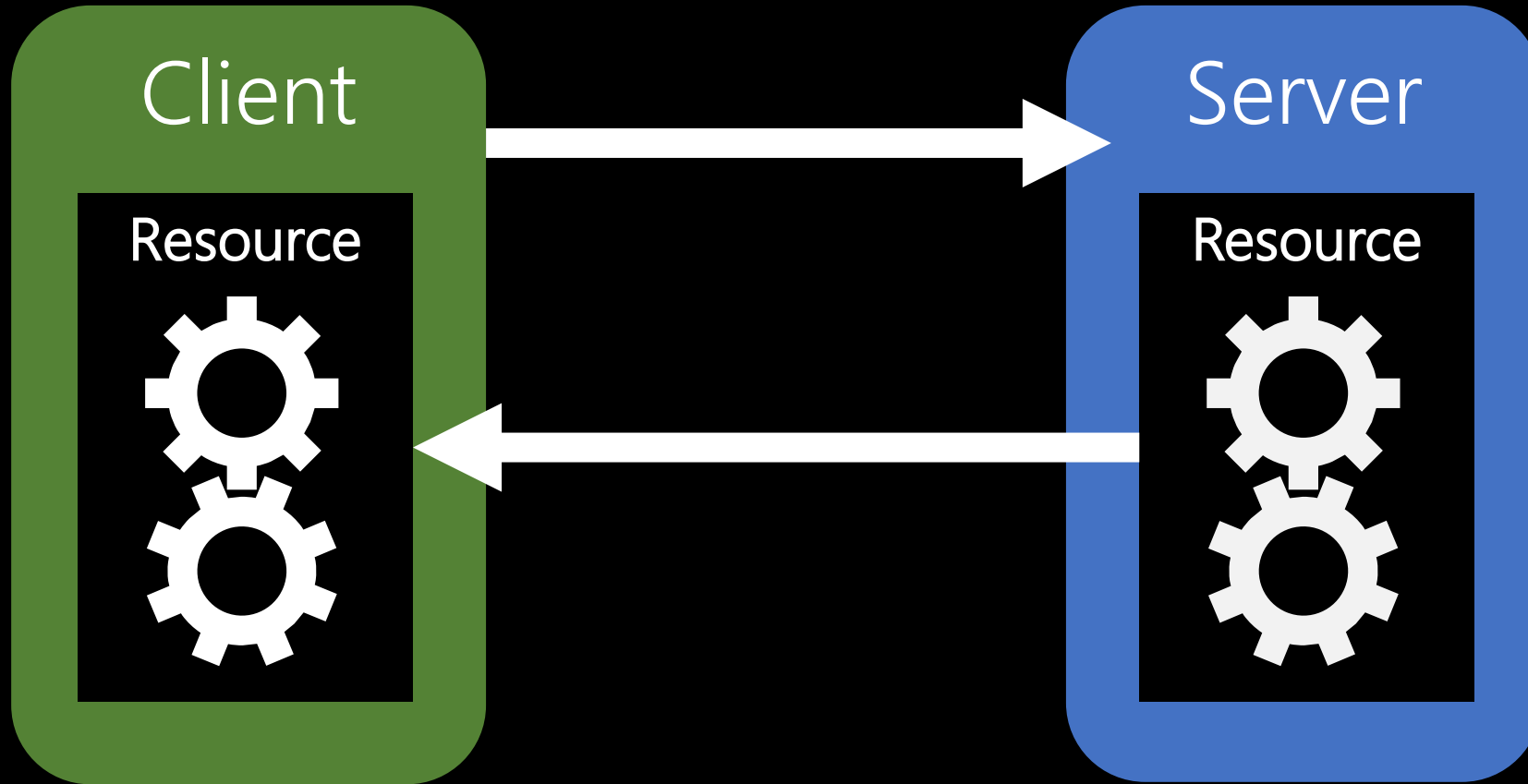
4. Layered System (continued)



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



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" }
  }
}
```

**“Ваш API
забил весь
наш канал!”**

**“Ваш идиотский
API перегружает
нашу базу
данных!”**

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=4 HTTP/1.1

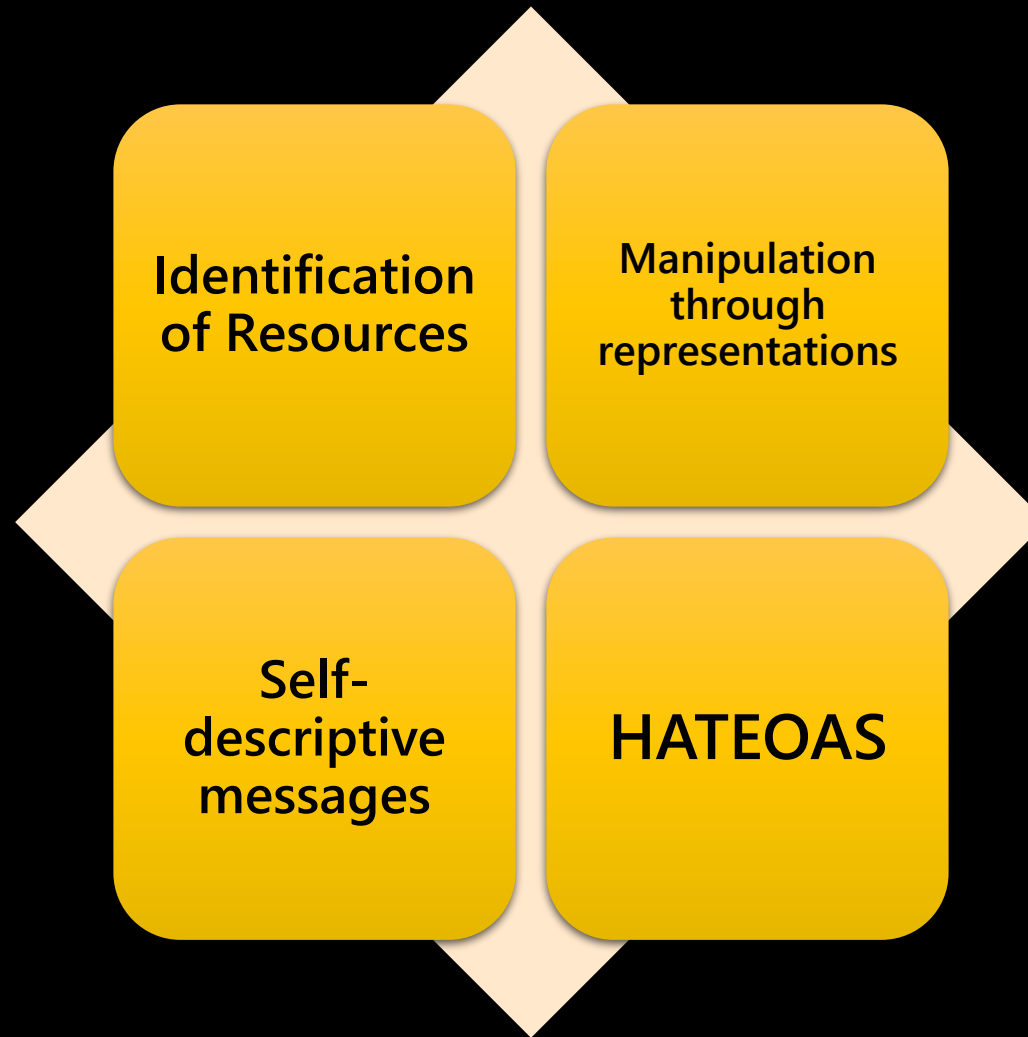
200 OK

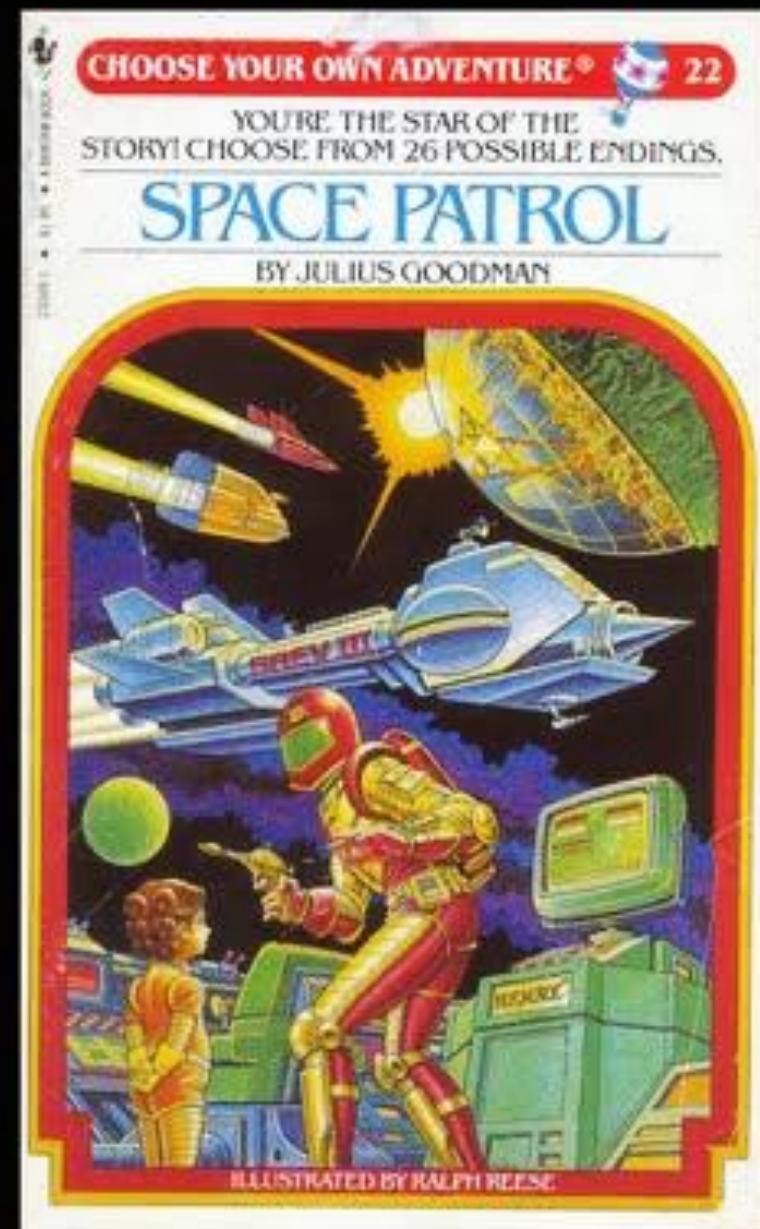
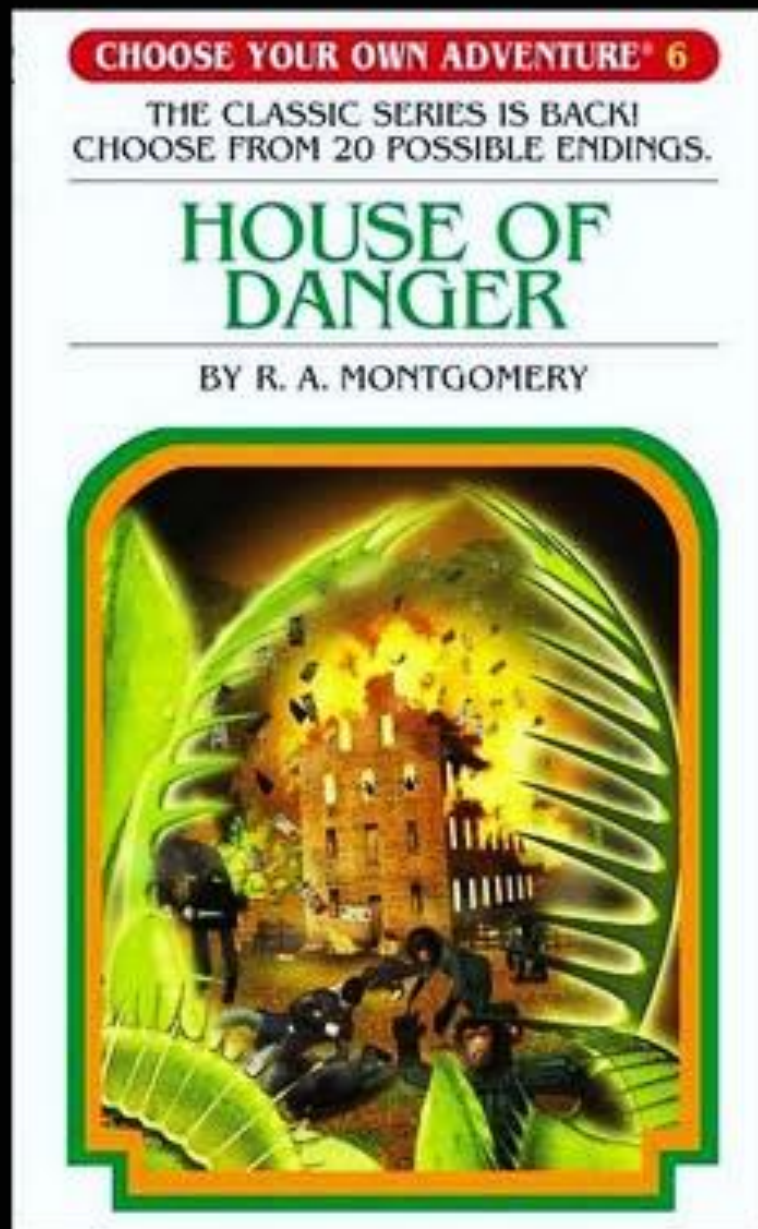
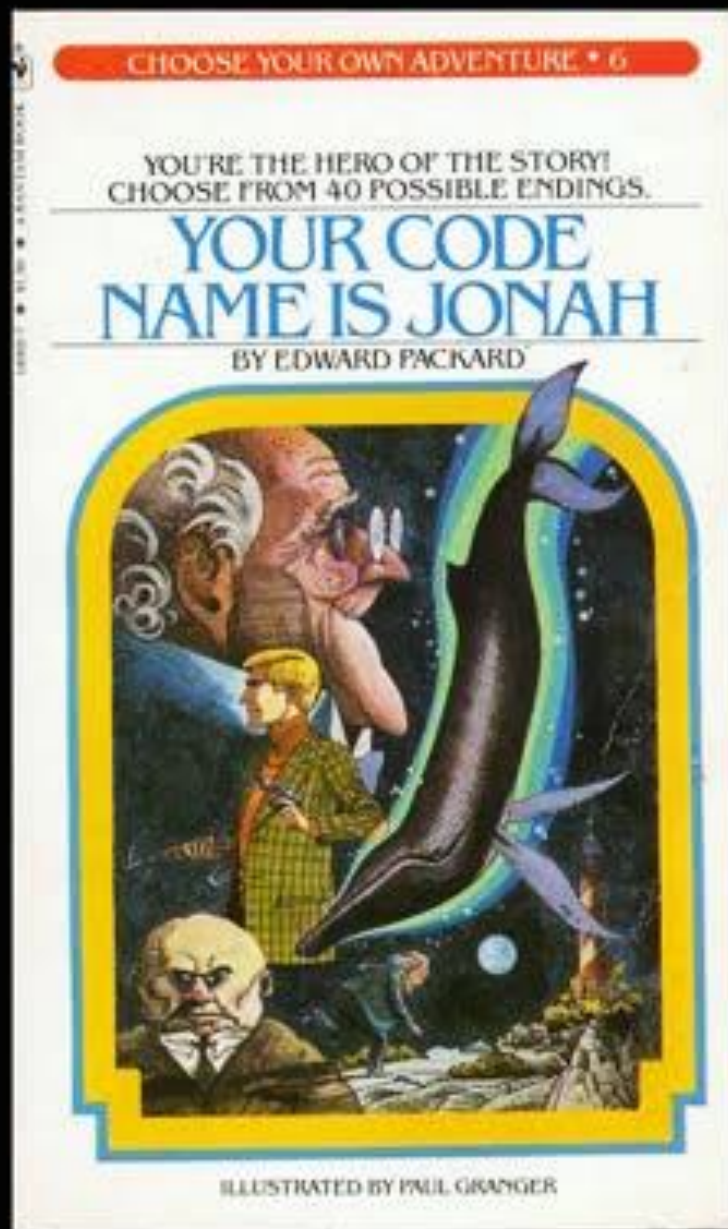
GET /profiles?page=5 HTTP/1.1

204 No Content

NOT REST

5. Uniform Interface





CHOOSE YOUR OWN ADVENTURE! #DOTNEXT

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

HANDS-ON HYPERMEDIA

ST. PETERSBURG



You have had an AWESOME first day at DotNext in Saint Petersburg. Now you are at the conference after-party. It has been a busy day; maybe you should go to your room and sleep? But some people are going out to have sushi – or you could stay at the party and watch Eugene do karaoke to Slim Shady?

To go to your room and sleep, turn to page 23

To go out and eat sushi, turn to page 41

To stay at the party and watch karaoke, turn to page 52

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":"Натáша 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":"Натáша 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":"Натáша Романов" },
    { "username":"pepper", "name":"Pepper Potts" },
    { "username":"storm", "name":"Ororo Munroe" }
  ]
}
```


Enough talk already...
SHOW ME THE

CODE!

<https://github.com/tonsky/FiraCode>

`==`

`=`

`!=`

`≠`

`=>`

`⇒`

`...`

`⋮`

`<=`

`≤`

`>=`

`≥`

Server:

Path:

[Reset](#)

Ready.

Server:

Path:

[Reset](#)

Ready.

```
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;  
        }  
    }  
}
```

Server:

Path:

[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"
  },
]
```

```
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;  
        }  
    }  
}
```


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 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

```
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;
```

```
}
```

```
}
```

```
}
```

```
    }  
  
    /// <summary>Generate a hypermedia links object containing first/final/prev/next links  
    /// <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;  
    }
```

Full Screen

DynamicExtensions.cs

ProfilesController.cs

index.html

Hal.cs

explorer.js

```
        this.db = new DemodbDatabase();  
    }  
  
    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

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": "Натáша 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": "Натáша 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": "Натáша 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": "Натáша 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": "Натáша 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":"Натáша 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":"Натáша Ромáнов", "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": "Наташа 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!

"Ваш API
забил весь
наш канал...
ОПЯТЬ!"

"Ваш API
перегружает базу
данных. Опять!"

...Я говорил, нам нужно
было держаться РНР...

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

"Я должен сделать
50 вызовов API
просто чтобы
выдать одну веб
страницу!!"

"Наш веб сервер
упал из-за того, что
IIS-логи заполнили
драйв C:!"

У нас никогда не было
таких проблем с 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"
      }
    ]
  }
}
```

DEMO #2

- Resource Expansion



Server:

Path:

[Reset](#)

Ready.

Server:

Path:

[Reset](#)

Ready.

```
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;  
        }  
    }  
}
```


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();  
    var links = Html.Paginate("/profiles", index, count, total);
```

184 %

Output

Ready

Ln 25

Col 46

Ch 46

INS

↑ 0

✎ 2

RealWorldREST

master

```
namespace RealWorldRest.NancyFX.Modules {  
    public static class ObjectExtensions {  
        public static dynamic ToDynamic(this object value) {  
            IDictionary<string, object> expando = new ExpandableObject();  
            var properties = TypeDescriptor.GetProperties(value.GetType());  
            foreach (PropertyDescriptor property in properties) {  
                expando.Add(property.Name, property.GetValue(value));  
            }  
            return (ExpandableObject)expando;  
        }  
  
        public static IDictionary<string, object> ToDictionary(this object d) {  
            return d.GetType().GetProperties()  
                .ToDictionary(x => x.Name, x => x.GetValue(d, null));  
        }  
    }  
}
```

DynamicExtensions.cs

ProfilesModule.cs* X

```
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 = db.ListProfiles().Skip(index).Take(count);
```

184 %

Output

(local variable) int count

Server:

Path:

[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"
}
```

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.Paginate("/profiles", index, count, total);  
}
```

184 %

Output

Ready

Ln 31

Col 1

Ch 1

INS

↑ 0

✎ 2

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

"Почему мне
приходится
засовывать в PUT
весь профайл
лишь для того,
чтобы обновить
координаты?"

"Мои обновления
происходят с
ошибкой 409
Conflict!
Помогите!"


```
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

PATCH /conferences/dotnext HTTP/1.1

Content-Type: image/epic.selfie+png



202 Accepted

Вопросы?

Спасибо!

Slides & code:

www.github.com/dylanbeattie/dotnext/

@dylanbeattie

dylan@dylanbeattie.net

DOTNEXT