



# A Babel Fish from the Swamp of POX

API Days Mediterranea

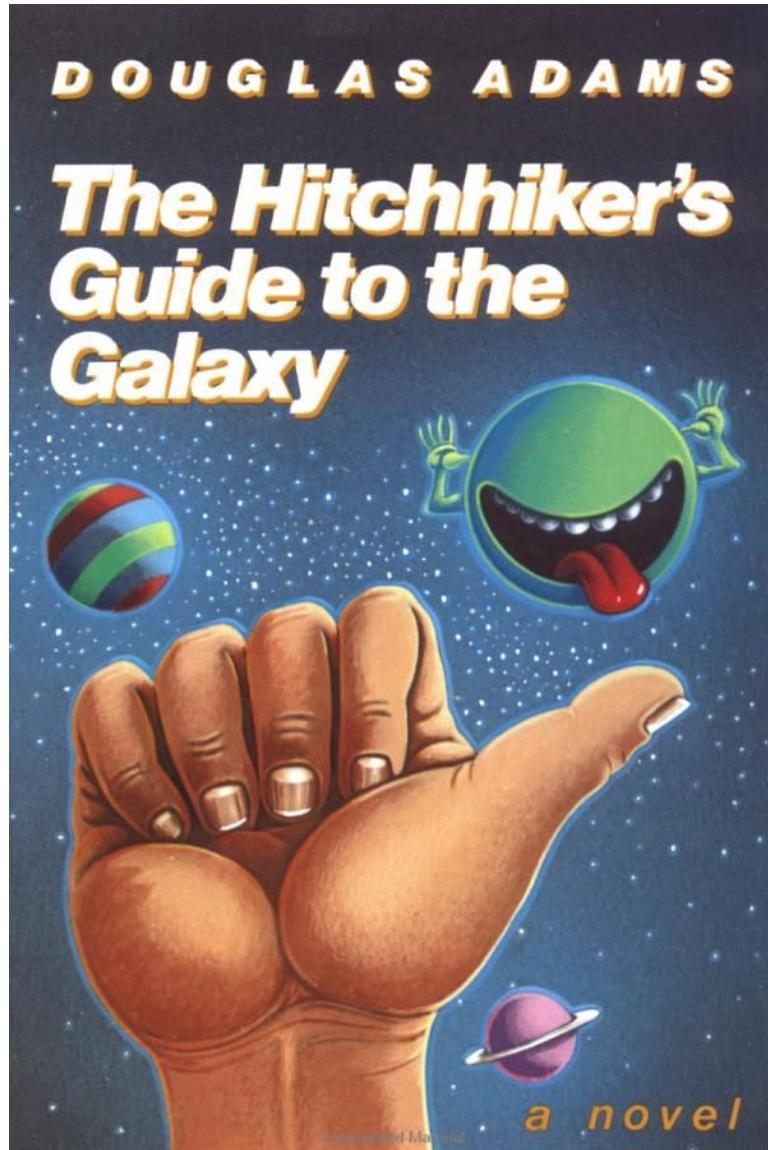
Ross Garrett

[rgarrett@axway.com](mailto:rgarrett@axway.com)

@gssor

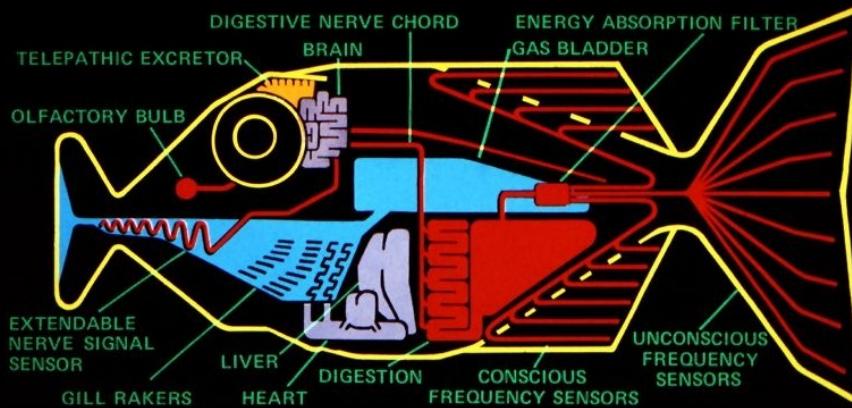
*May 6th, 2015*

# A Babel Fish?



# A Babel Fish?

## The Babel Fish



The Babel fish is small, yellow, leech-like, and probably the oddest thing in the universe. It feeds on brain wave energy, absorbing all unconscious frequencies and then excreting telepathically a matrix formed from the conscious frequencies and nerve signals picked up from the speech centres of the brain, the practical upshot of which is that if you stick one in your ear, you can instantly understand anything said to you in any form of language: the speech you hear decodes the brain wave matrix. It is a universal translator which simultaneously translates from one spoken language to another. It takes the brainwaves of the other body and what they are thinking then transmits the thoughts to the speech centres of the host's brain, the speech heard by the ear decodes the brainwave matrix. When inserted into the ear, its nutrition processes convert unconscious sound waves into conscious brain waves, neatly crossing the language divide between any species.

# A Swamp of POX?

Glory of REST



Level 3: Hypermedia Controls

Level 2: HTTP Verbs

Level 1: Resources

Level 0: The Swamp of POX



# Why are we here?

## WorldWideWeb:

Proposal for a HyperText Project

*T. Berners-Lee / CN, R. Cailliau / ECP*

### Abstract:

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will. Potentially, HyperText provides a single user-interface to many large classes of stored information such as reports, notes, databases, computer documentation and on-line systems help. We propose the implementation of a simple scheme to incorporate several different servers of machine-stored information already available at CERN, including an analysis of the requirements for information access needs by experiments.

### Introduction

The current incompatibilities of the platforms and tools make it impossible to access existing information through a common interface, leading to waste of time, frustration and obsolete answers to simple data lookup. There is a potential large benefit from the integration of a variety of systems in a way which allows a user to follow links pointing from one piece of information to another one. This forming of a web of information nodes rather than a hierarchical tree or an ordered list is the basic concept behind HyperText.

# The Web

*“We propose the implementation of a simple scheme to incorporate several different servers of machine-stored information already available.”*

- Berners-Lee / Cailliau, 1990

# The Web

*“... web of nodes in which the user can browse at will.”*

*- Berners-Lee / Cailliau, 1990*

# HTTP...

# Tell me what HTTP is?



# Tell me what HTTP is?

## Hypertext Transfer Protocol

# Tell me what HTTP is?

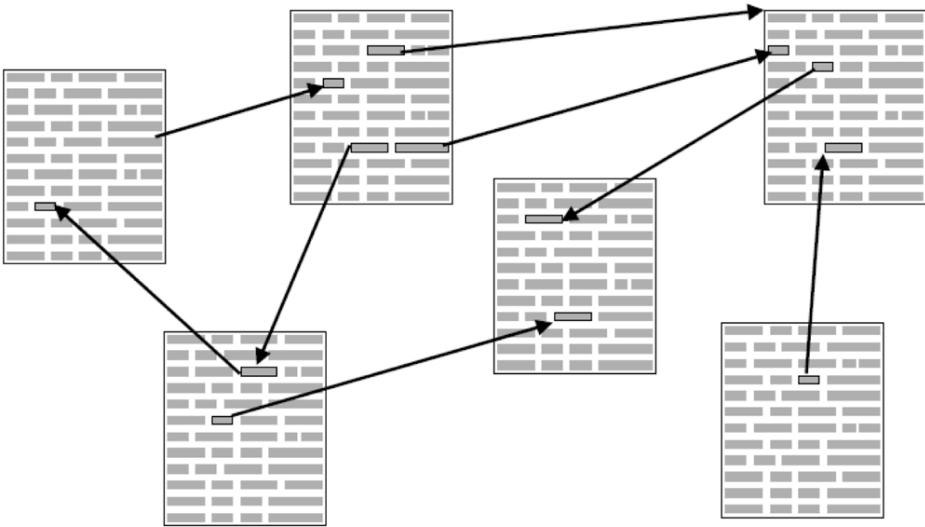
## How-To-Transfer Protocol

????

# HYPertext

# Hypertext

Ted Nelson coins the word “*Hypertext in A File Structure for the Complex, the Changing, and the Indeterminate*”



20th National Conference,  
New York,  
Association for Computing Machinery,  
1965.

# Hypertext

**Hypertext is text which contains links to other texts.**

# **And by extension...**

**HyperMedia is a term used for  
hypertext which is not constrained  
to be text**

# Consider a Web Application

## <http://www.facebook.com>



Facebook helps you connect and share with the people in your life.



Email or Phone  Password   
 Keep me logged in [Forgotten your password?](#)

## Create an account

It's free and always will be.

First name  Surname   
Email or mobile number   
Re-enter email or mobile number   
New password

### Birthday

Day  Month  Year

Why do I need to provide my date of birth?

Female  Male

By clicking Create an account, you agree to our [Terms](#) and that you have read our [Data Policy](#), including our [Cookie Use](#).

# The Web is more than just HTTP

# The Web is more than just HTTP

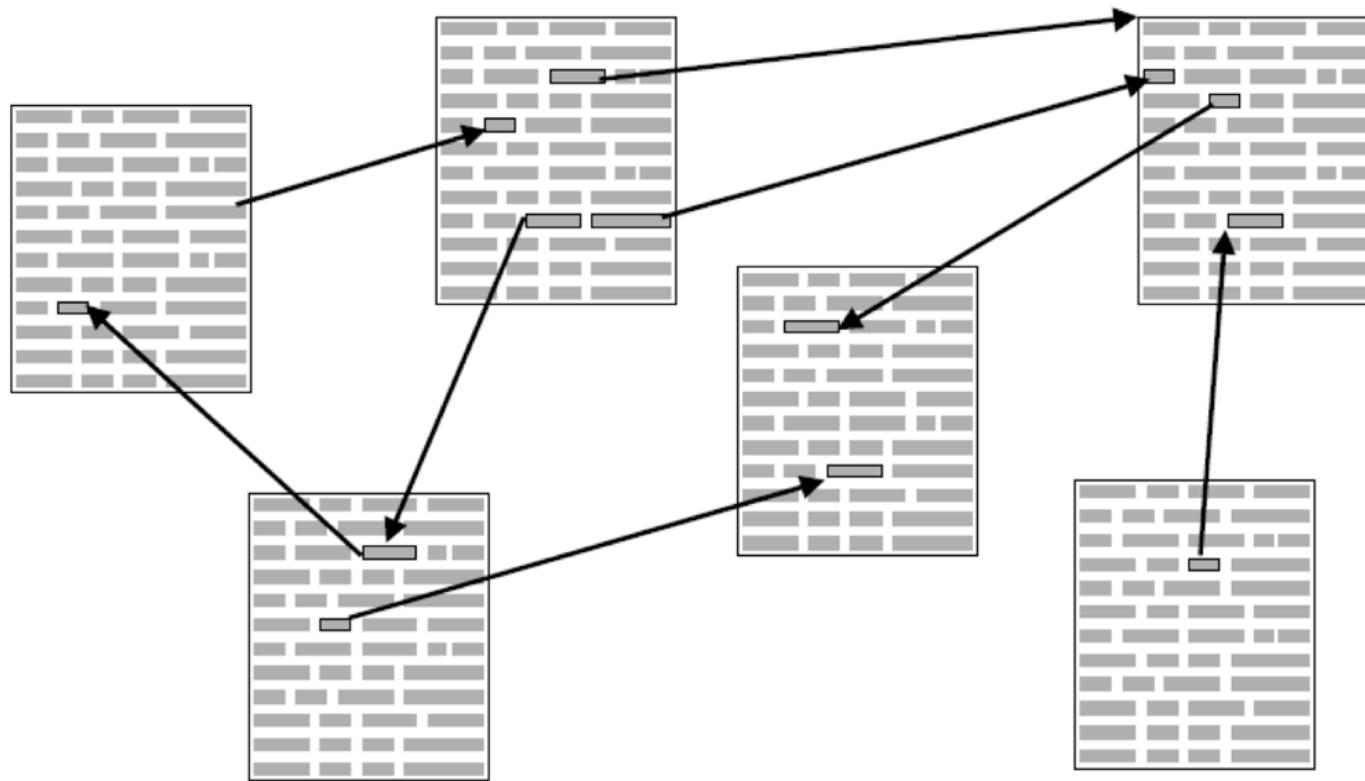
- **HTTP**
- **URI**
- **HTML**

We need to use ALL of the Web

# **But have we forgotten this in a World of APIs?**

# ~~How-To-Transfer Protocol~~

# The Web



# A Reminder

Glory of REST



Level 3: Hypermedia Controls

Level 2: HTTP Verbs

Level 1: Resources

Level 0: The Swamp of POX

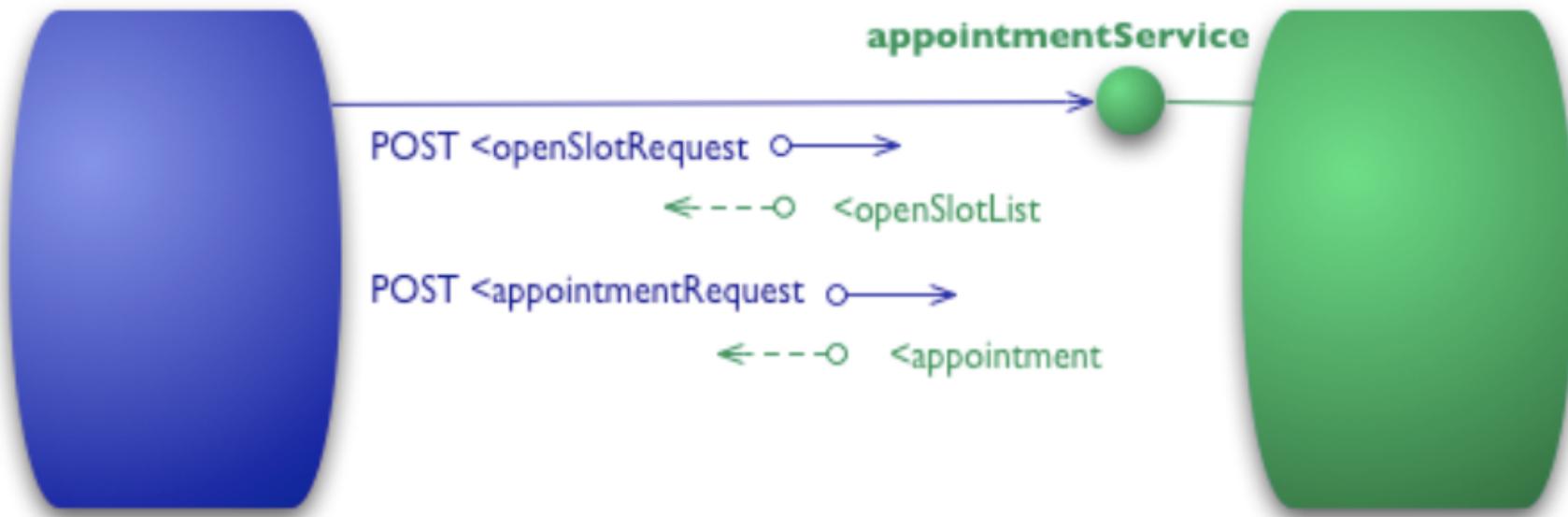




From the beginning then...



# The Swamp of POX

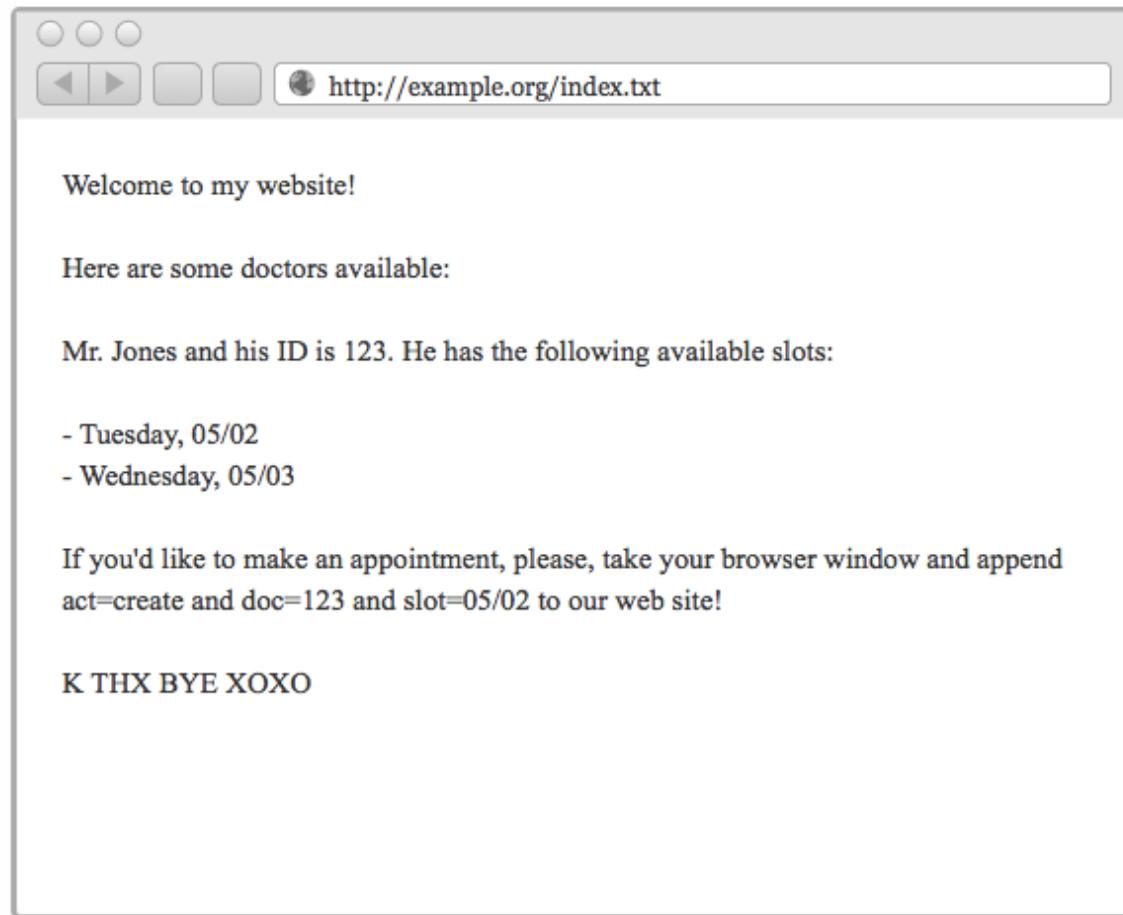


*Figure 2: An example interaction at Level 0*

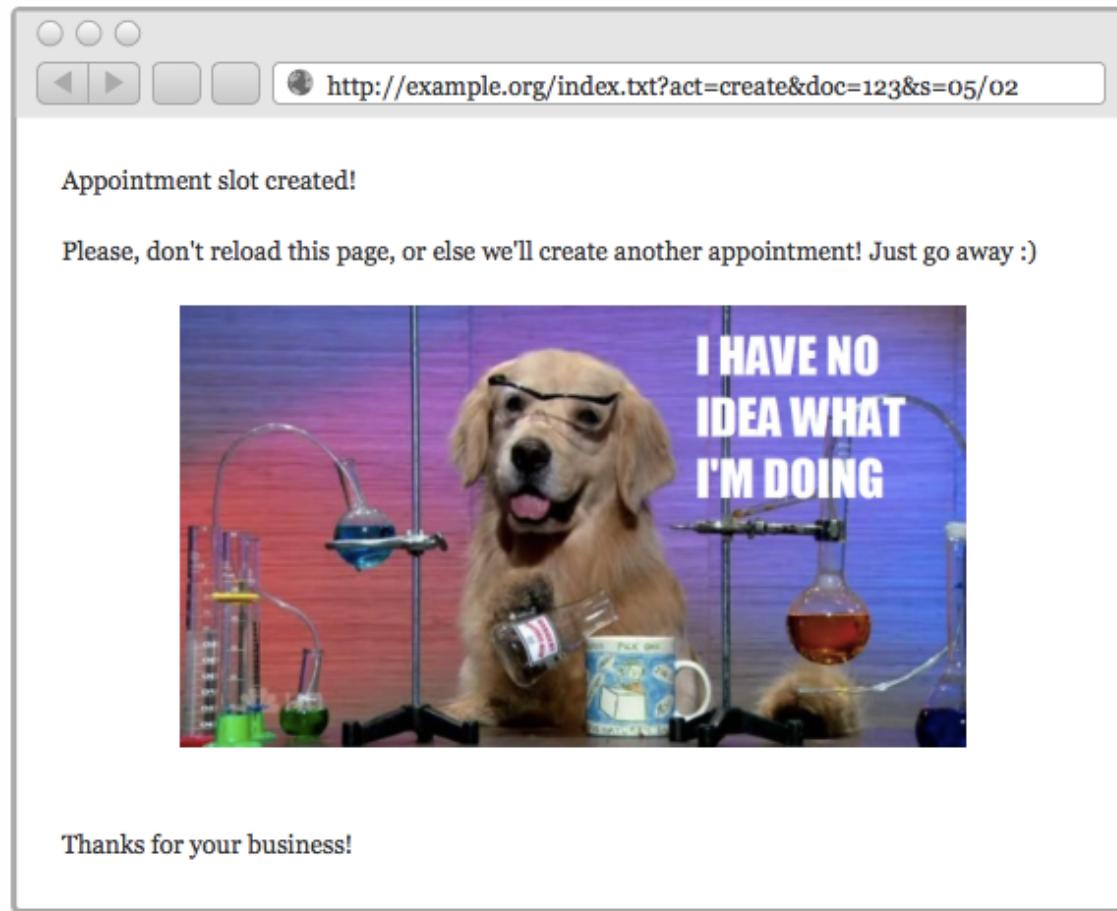
# A example POX API

```
1. POST / HTTP/1.1
2. Host: www.example.org
3. Content-Type: application/soap+xml; charset=utf-8
4. Content-Length: 299
5. SOAPAction: "http://www.w3.org/2003/05/soap-envelope"
6. <?xml version="1.0"?>
7. <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
8.   <soap:Body>
9.     <m:ScheduleAppointment xmlns:m="http://www.example.org/doctors">
10.      <m:DoctorId>123</m:DoctorId>
11.    </m:ScheduleAppointment>
12.  </soap:Body>
13. </soap:Envelope>
```

# Looking at this as a Web Page



# Looking at this as a Web Page



# Level 1 - Resources

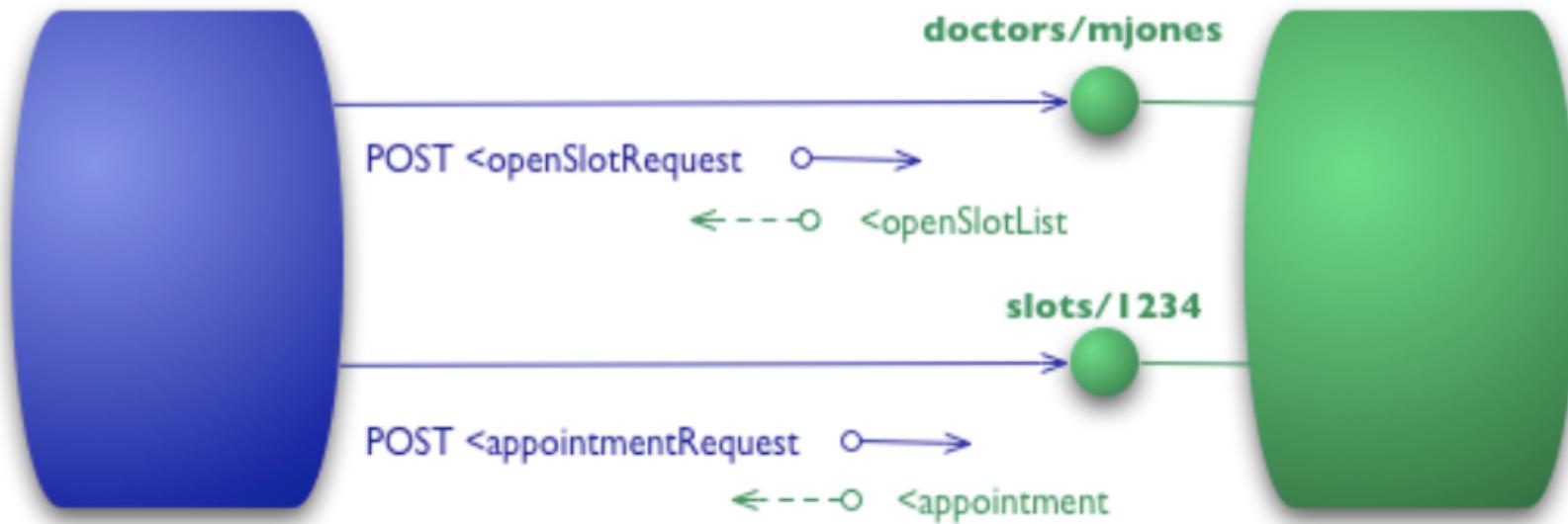
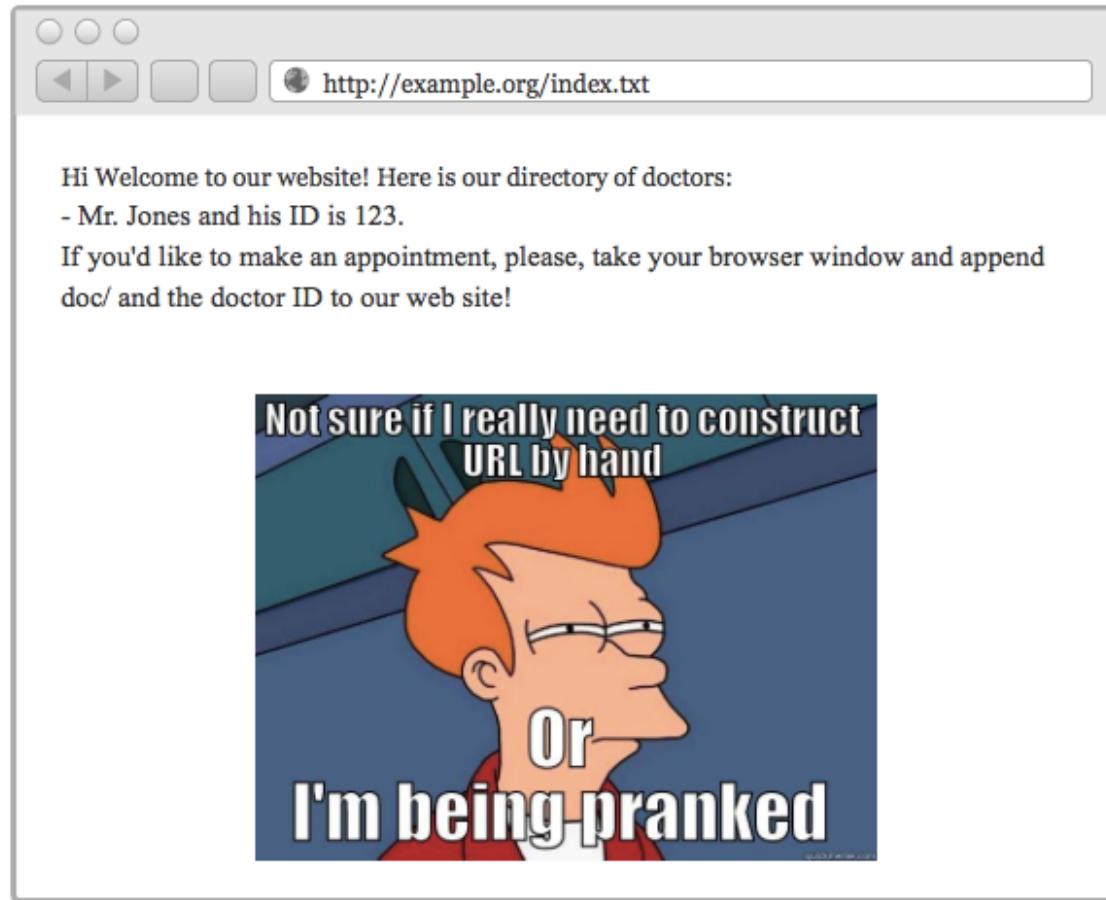


Figure 3: Level 1 adds resources

# An example Level 1 API

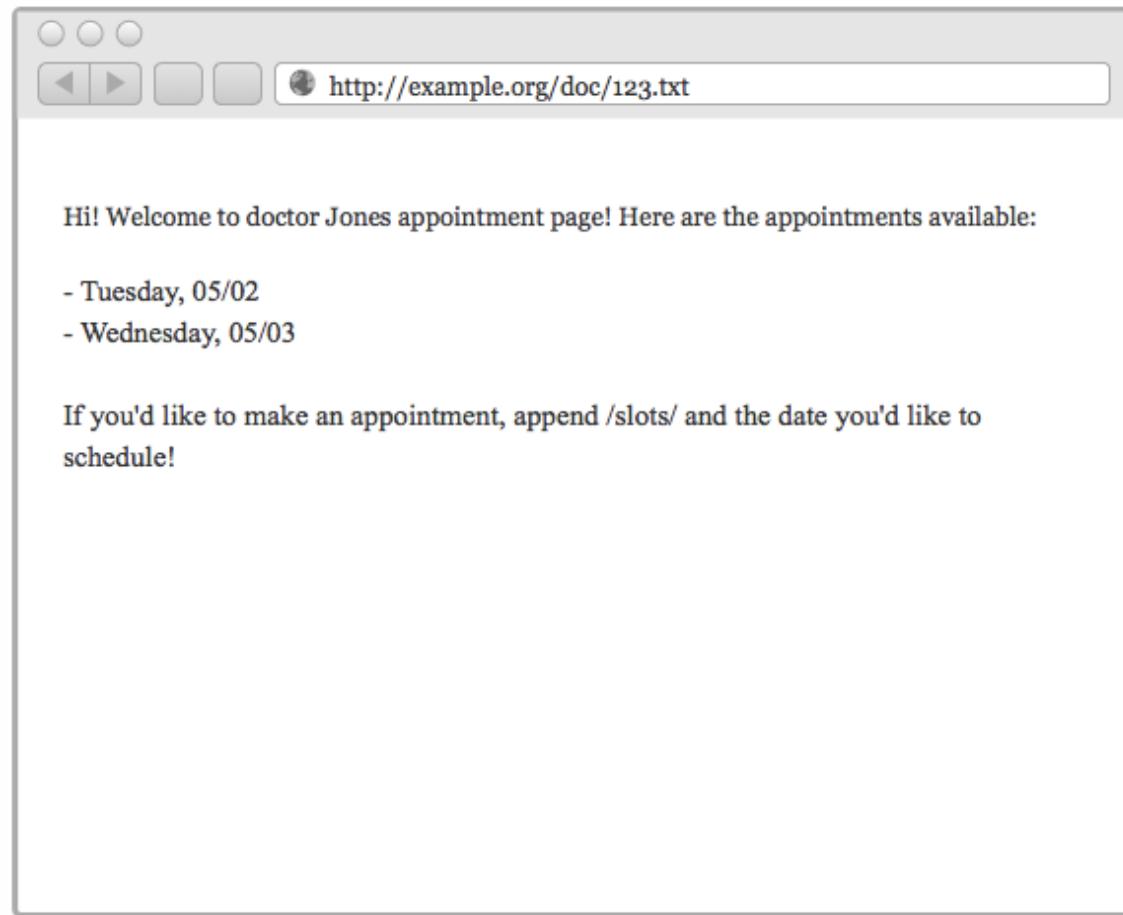
1. GET /doc/123/slots/05/02.txt?action=create HTTP/1.0
2. Host: example.com

# Looking at this as a Web Page

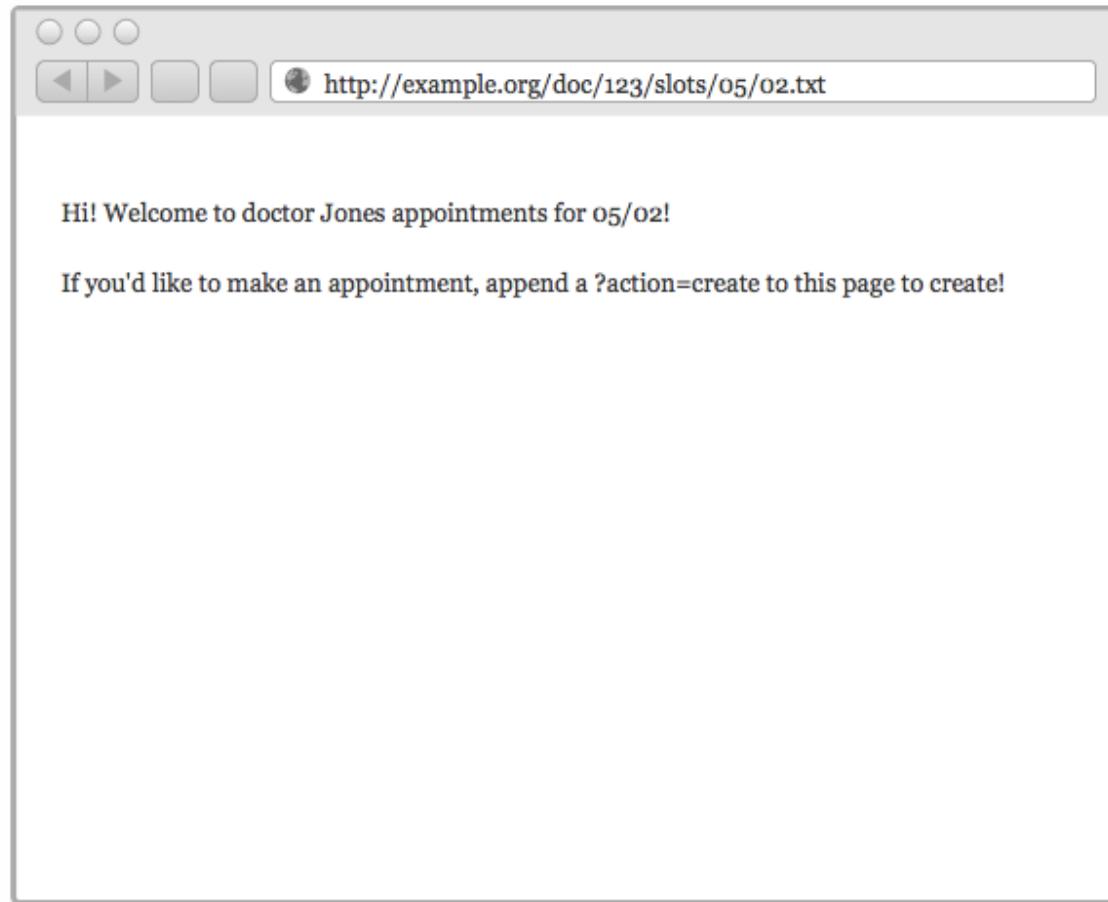


<http://blog.sgo.to/2014/04/what-your-api-would-look-like-as-webpage.html>

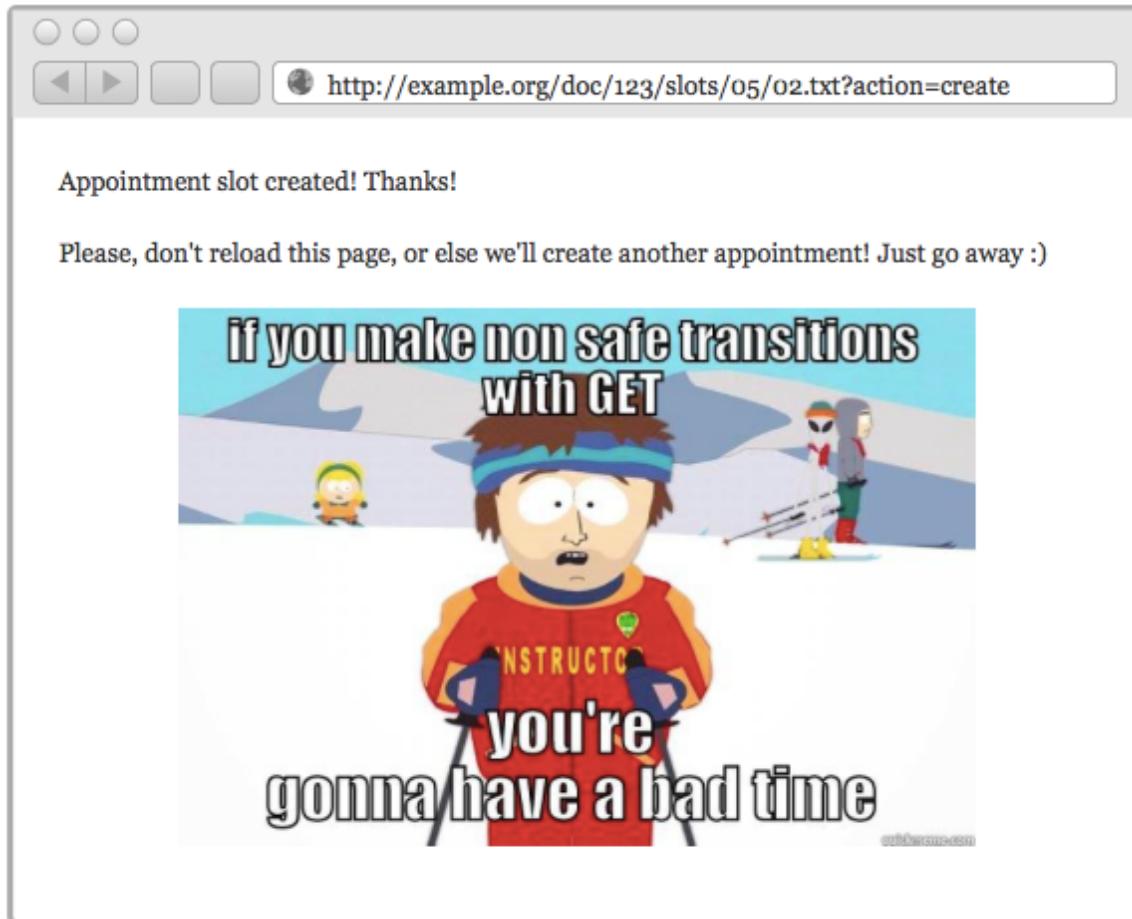
# Looking at this as a Web Page



# Looking at this as a Web Page



# Looking at this as a Web Page



<http://blog.sgo.to/2014/04/what-your-api-would-look-like-as-webpage.html>

# Level 2 - HTTP Verbs

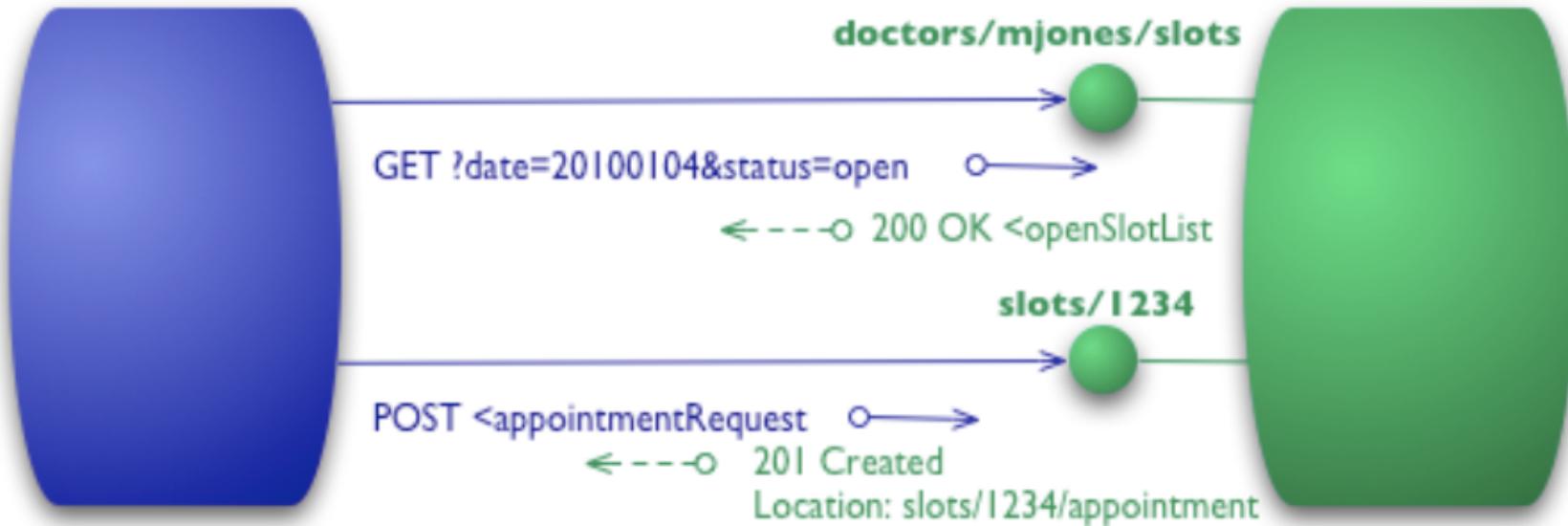
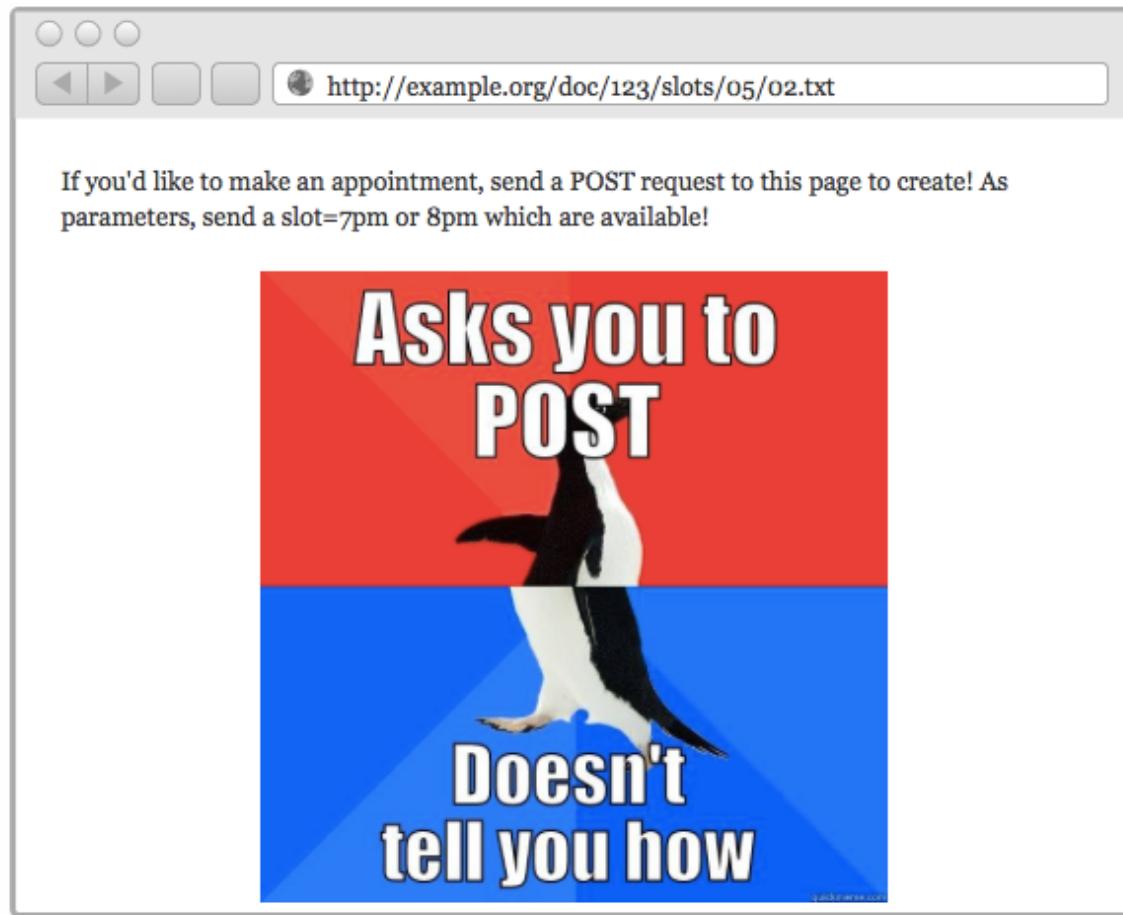


Figure 4: Level 2 adds HTTP verbs

# An example Level 2 API

```
1. GET / HTTP/1.0
2. Host: example.com
3.
4. HTTP/1.0 200 OK
5. Content-Type: text/json
6. {
7.   "doctors": [
8.     "name": "Dr. Jones",
9.     "id": "123"
10.   ]
11. }
```

# Looking at this as a Web Page



# Looking at this as a Web Page



<http://blog.sgo.to/2014/04/what-your-api-would-look-like-as-webpage.html>

# Level 3 – Hypermedia Controls

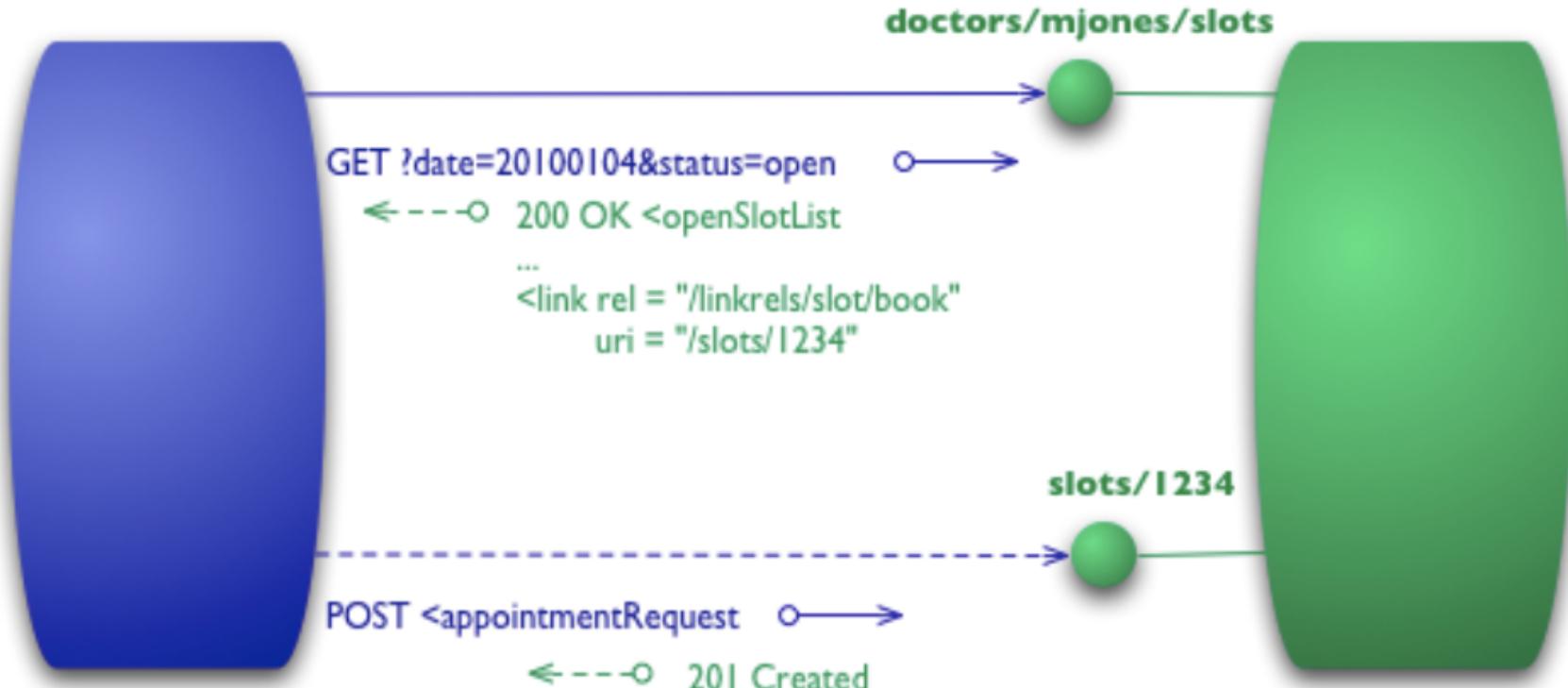
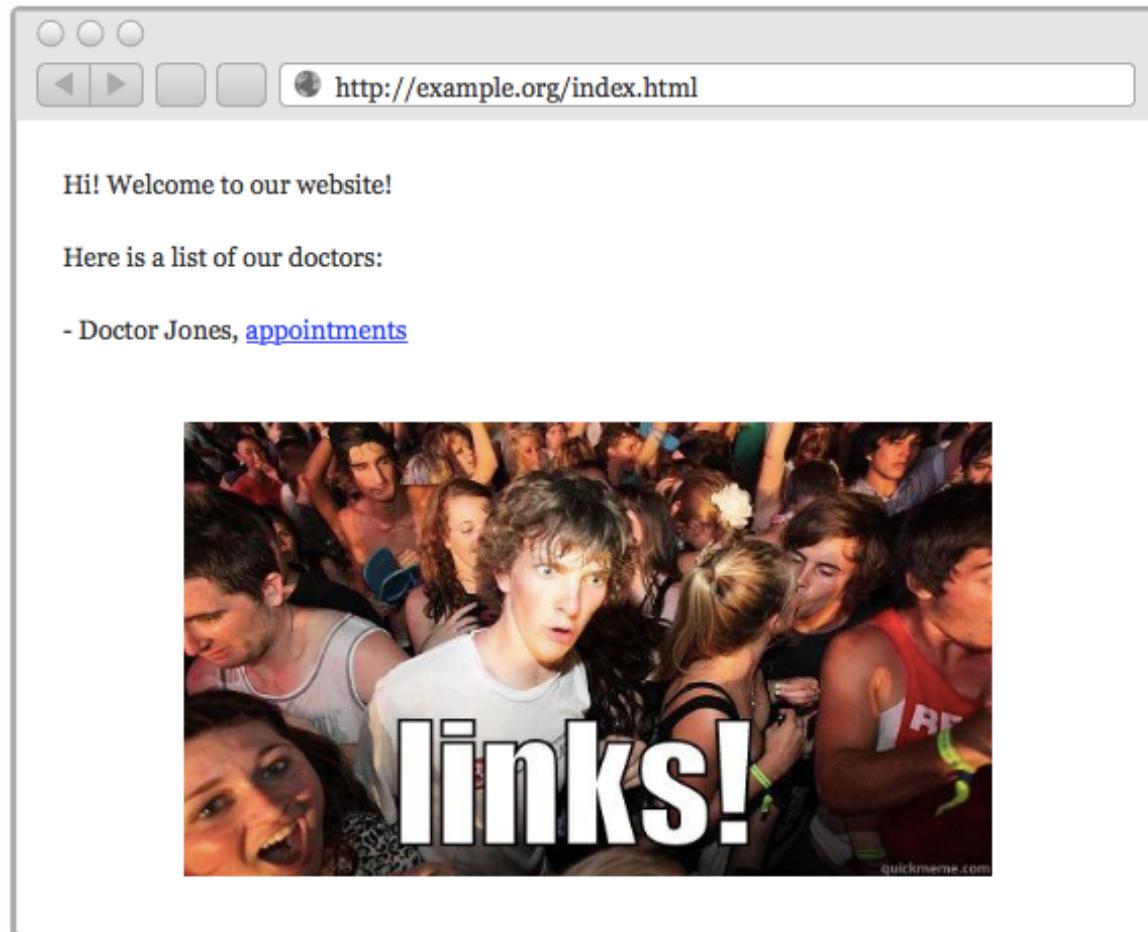


Figure 5: Level 3 adds hypermedia controls

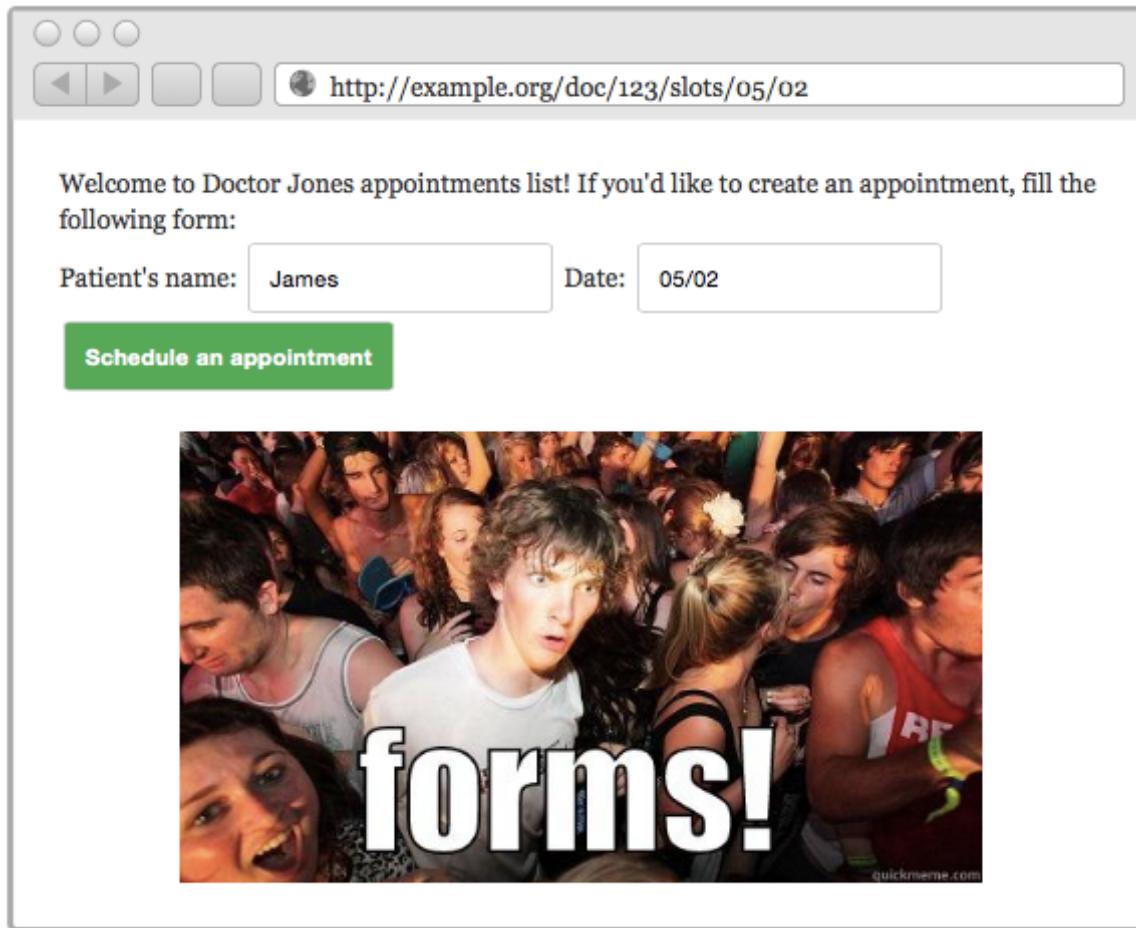
# An example Level 3 API

```
1. GET / HTTP/1.0
2. Host: example.com
3.
4. HTTP/1.0 200 OK
5. Content-Type: text/ld+json
6. {
7.   "@context": "schema.org",
8.   "@type": "Clinic",
9.   "@id": "http://example.com/",
10.  "doctors": [
11.    {"@type": "Doctor",
12.     "@id": "http://example.com/doctors/123",
13.     "name": "Dr. Jones",
14.     "appointments": {
15.       "@type": "AppointmentBook",
16.       "@id": "http://example.com/doctors/123/slots",
17.       "action": {
18.         "@id": "http://example.com/doctors/123/slots",
19.         "@type": "ScheduleAction"
20.       }
21.     }
22.   }]
23. }
```

# Looking at this as a Web Page



# Looking at this as a Web Page



<http://blog.sgo.to/2014/04/what-your-api-would-look-like-as-webpage.html>

# Using ALL of the Web

- **Level 1**
  - tackles the question of handling complexity by using divide and conquer, breaking a large service endpoint down into multiple resources.
- **Level 2**
  - introduces a standard set of verbs so that we handle similar situations in the same way, removing unnecessary variation.
- **Level 3**
  - introduces discoverability, providing a way of making a protocol more self-documenting.

# Points to Note

- **XML is not the only way a resource can be represented**
- **The choice of representation is left to the implementation**
- **But remember the difference between Hypermedia formats and static formats**
  - **HTML vs. TXT**

# In Summary

- **Following principles of the Web**
  - Scalable
  - Recoverable
  - Fault-Tolerable
  - Secure
  - Loosely Coupled

These are the same requirements we have  
for software systems

**Level 3 REST supports these requirements by using ALL of the Web**



# Thank You!

Ross Garrett

[rgarrett@axway.com](mailto:rgarrett@axway.com)

@gssor