Pragmatic REST

IPC 2013 SE

Tobias Schlitt (@tobySen)

05.06.2013

Qafoo

passion for software quality

About me

- Tobias Schlitt (**Toby**)
- Degree in computer sience
- Proessional PHP since 2000
- Open source enthusiast
- Passion for
  - Software Design
  - Automated Testing

Co-founder of

Helping teams to create high quality web application development.

`http://qafoo.com`

- Expert consulting
- Individual training

A Story about Alex

"Hi, I'm Alex!"

- Alex is a developer
- Yet Another Webshop Nemesis (YAWN)
- Needs a web service API

Time Leap

"Go with REST, they said. Easy and awesome, they said."

But instead, he found REST to be . . .

- . . . bloated
- . . . overly complicated
- . . . unusable

Back to Start

Back to Start

Back to Start

“There is this REST thingy, I’ll look out for that.”

Network Architecture

LCoDC$SS

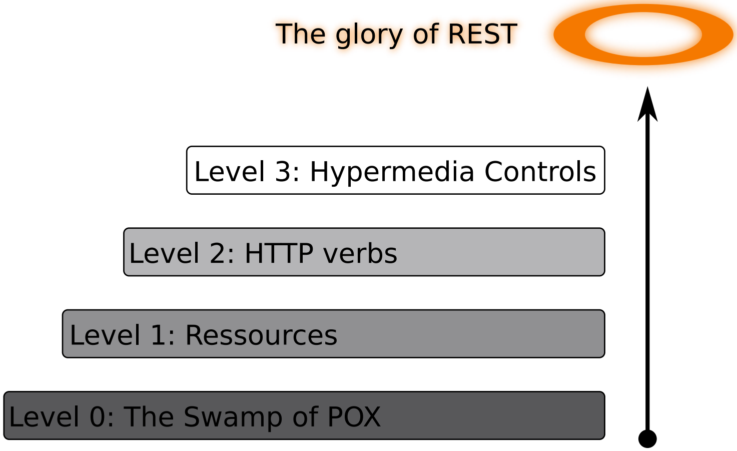
“*WT. . . ?*”

Network Architecture

Layered Code on Demand Client Cached Stateless Server

“*A, well, that’s HTTP based!*”

Richardson Maturity Model



Source: http://martinfowler.com/articles/richardsonMaturityModel.html

HATEOAS

Hypermedia As The Engine Of Application State

Resources

“*I should identify some resources first*”

- Product
- Category

Resources

- Identified by an URI
- Never duplicated

URIs

- http://example.com/products/23
- http://example.com/categories/geek_toys
    - http://.../categories/geek_toys/?index=20&limit=10
    - http://.../categories/geek_toys/?sort=sales&limit=10

Resource Representations

- Hypermedia format
- Semantical meaning
- Links
- Auto-discovery

Atom

- Hypermedia Format
- Publishing / aggregating content
- Has proper `<link>` element
- Re-used commonly in REST
- `https://tools.ietf.org/html/rfc4287`
- Alternative: X-Link

"I should use the `<link>` element from Atom!"

Qafoo
passion for software quality

IANA Link Relations

- IANA collects link relations
- Mostly from RFCs
- For example:
  - `nofollow`
  - `item`
  - `collection`
  - `first` / `last`
  - `self`
  - `edit`
  - `payment`
- `https://www.iana.org/assignments/link-relations/link-relations.xml`

Product Resource

```xml
<?xml version="1.0" encoding="UTF-8"?>
<product
  xmlns="urn:com.example.product"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <atom:link rel="self"
    type="application/vnd.com.example.product+xml"
    href="http://example.com/products/23" />
  <name>Glow Stone Driveway</name>
  <description>Awesome ...</description>
  <atom:link rel="collection"
    type="application/vnd.com.example.category+xml"
    href="http://example.com/categories/geek_toys" />
  <!-- More links ... -->
</product>
```

passion for software quality

Category Resource

```xml
<?xml version="1.0" encoding="UTF-8"?>
<category
  xmlns="urn:com.example.category"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <atom:link rel="self"
    type="application/vnd.com.example.category+xml"
    href="http://example.com/categories/geek_toys" />
  <name>Geek Toys</name>
  <products>
    <atom:link rel="item"
      type="application/vnd.com.example.product+xml"
      href="http://example.com/products/23" />
    <!-- ... -->
  </products>
  <!-- Links: overview, paging, sorting, ... -->
</category>
```

Proud Alex

“*This is an awesome start for my REST API!*”

Flashback to Reality

- Will people really auto-discover?
  - Hard coded URLs
  - No support for redirects
- Will they use URIs for identification?
  - Parsing URLs with Regex

Move on . . .

“*A standard use case is to fetch the top 10 products. People do that very often.*”

Resource Embedding

"Let's search the web . . ."

- Hypertext Application Language (HAL)
- Provides means of embedding resources
- Non-standardized
- `http://stateless.co/hal_specification.html`

HAL example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<resource href="/categories/geek_toys">
  <name>Geek Toys</name>
  <products>
    <resource rel="product" href="/products/23">
      <name>Glow Stone Driveway</name>
      <description>Awesome ...</description>
      <link rel="category" href="/categories/geek_toys" />
    </resource>
    <!-- ... -->
  </products>
</resource>
```

passion for software quality

HAL issues

"*Wait, hyper media . . . ?*"

- No semantical meaning
- Missing namespacing
- Non-standard links

Alternative Approach

```xml
<?xml version="1.0" encoding="UTF-8"?>
<category xmlns="..." xmlns:atom="..."
  xmlns:p="urn:com.example.product"><!-- ... -->
  <products>
    <atom:link rel="item" type="..." href="...">
      <p:product><!-- ... -->
        <p:name>Glow Stone Driveway</p:name>
        <p:description>Awesome ...</p:description>
        <atom:link rel="collection"
          type="application/vnd.com.example.category+xml"
          href="http://example.com/categories/geek_toys" />
        </p:product>
    </atom:link>
    <!-- ... -->
  </products>
</category>
```

passion for software quality

Alternative Approach

- Also not standardized . . .
- . . . but plays better with standards

Moving on . . .

“*What about caching?*”

Method Semantics

“Aha, methods can be safe and idempotent . . .”

- GET
- HEAD
- POST
- PUT
- DELETE
- OPTIONS
- . . .

passion for software quality

Cachability

- Basically GET / HEAD
- Invalidate by writing requests

Caching Headers

- Response
  - Last-Modified
  - Expires
  - ETag
  - . . .
- Request
  - If-Modified-Since
  - If-None-Match
  - . . .
- Both directions
  - Cache-Control

Concurrency Control

- If-Unmodified-Since
- If-Match

Oh my . . .

“*That caching stuff is soooo complicated . . .*”

Flashback to Reaility

Resource embedding makes caching even more complex

- Purging of representation
- State information retrieval

Moving on . . .

“*I need to send complete resources for update?*”

Partial Updates

"*But our products are huge, I need partial updates*"

- Custom HTTP method
- But there is PATCH
  - `https://tools.ietf.org/html/rfc5789`
- Requires new media types

Reality Flashback

- Can I really use custom methods?
  - Browsers
  - Webservers
  - Proxies
- What can I do?
  - Work around
  - e.g. X-Method-Overwrite header

Move on . . .

“*So, what about these media types?*”

Media Types

- Give semantics to entities
- Distinguish between representations
- Support validation
- Headers
    - Content-Type
    - Header

Media Type Examples

- application/com.example.product+xml
- application/com.example.product.v2+xml;
- application/com.example.product-update+json;
- application/xhtml+xml
- application/atom+xml

Flashback to Reality

- Do people really use media type?
  - Lucky if they send correct XML
- Shall I really validate on Accept?
  - Just accept . . .
  - Maybe, if Content-Type is available

Move on . . .

“*But I want to use the API from the browser!*”

XML vs. JSON

- Attributes
- Namespaces
- DOM
- Schema / validation

Flashback to Reality

- Will JS developers use my XML API?
  - No
  - Unless forced to . . .

XML & JSON

- Media types
  - application/com.example.product+xml
  - application/com.example.product+json
- 2 possible ways:
  - Strip XML down to JSON facilities
  - Attempt to emulate XML facilities in JSON

Flashback to Reality

- Should I emulate XML in JSON?
  - No
- Should I strip my XML down to JSON?
  - No, just use JSON

Move on . . .

“*Now, I need authentication . . .*”

Authentication

- Basic / Digest Auth
- OAuth
- No cookies!

Flashback to Reality

- What will I use from my own JavaScript?
  - OAuth
  - Cookies

Conclusion

- REST sucks
- REST is awesome
- REST is complex
- Check your use-case!
  - Service vs. application vs. project
  - Agile vs. long term vs. unknown
  - Users vs. implementors vs. business

passion for software quality

passion for software quality

THANK YOU

Rent a quality expert
qafoo.com