Embracing HTTP in the era of Web API’s

Hadi Hariri
With HTTP, your API is your Application
Two Interfaces

User Interface

Web Application

API
The Advantages

- It’s easy
- No real “other” way
The Disadvantages

- Two Systems to maintain
- API is usually an afterthought
- Limitations of API
One Interface

Web Application
One Interface?

NEWMSG("joe@gmail.com")
One Interface

API

Web Application
The Advantages

• One Interface
• Single System
• The API is designed from the get-go
The Disadvantages

- The API is designed from the get-go
Is this possible?
Why is this possible? Why now?

• Frameworks Evolve – More push to Client Side
• JavaScript has become a viable language
• HTTP has been undervalued
• REST is now a buzzword
HTTP in the OSI Layer

Application
Presentation
Session
Transport
Network
Data Link
Physical
HTTP 101
Demo

HTTP Headers, Queries and Posts
Let’s talk API
POST /domain.com/appServices

<?xml version="1.0" encoding="ISO-8859-1"?>

<methodCall>
    <methodName>
        calculateAge
    </methodName>
    <params>
        <param name="year">
            1973
        </param>
    </params>
</methodCall>

<methodResponse>
    <param name="age">
        41
    </param>
</methodResponse>

200 OK
We can do better
Multiple Endpoints
POST /domain.com/calculateAge

```
<params>
  <year>1973</year>
</params>
```

200 OK

```
<methodResponse>
  <param name="age">
    41
  </param>
</methodResponse>
```

POST /domain.com/getCustomers

200 OK

```
<customers>
  <customer>
    <id>1</id>
    <name>Joe Smith</name>
  </customer>
  <customer>
    <id>2</id>
    <name>Jane Jones</name>
  </customer>
</customers>
```
Typical CRUD Apps

• Create
• Read
• Update
• Delete
Typical CRUD Apps

- Create – POST
- Read – GET
- Update – PUT
- Delete – DELETE

* Don’t take this literally...
GET /domain.com/customer/1 200 OK

```
<customers>
  <customer>
    <id>1</id>
    <name>Joe Smith</name>
  </customer>
</customers>
```

POST /domain.com/customer 200 OK

```
<customers>
  <customer>
    <name>Jane Jones</name>
  </customer>
</customers>
```

```
<customers>
  <customer>
    <id>2</id>
  </customer>
</customers>
```
Demo
Response with Json
Verbs

- Client-Side Support
- Hidden Field with POST
- Query Parameter with GET/POST
- X-Http-Method-Override Header
Allowed Verbs

- Not every resource allows every method
- Allows you to restrict what can be done
Demo

Auto Options
Creating Resource and Location Headers

POST /domain.com/customer 200 OK

```
<customers>
  <customer>
    <name>Jane Jones</name>
  </customer>
</customers>
```

Cache-Control: private
Content-Type: application/xml; charset=UTF-8
Location: http://domain.com/customer/2
Errors

POST /domain.com/customer

```xml
<customers>
  <customer>
    <name>Jane Jones</name>
  </customer>
</customers>
```

500 Internal Server Error

```xml
<errors>
  <message>
    Could not create customer
  </message>
</errors>
```
<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>OK</td>
</tr>
<tr>
<td>201</td>
<td>Created. New resource</td>
</tr>
<tr>
<td>202</td>
<td>Accepted. Async Ops</td>
</tr>
<tr>
<td>301</td>
<td>Moved Permanently.</td>
</tr>
<tr>
<td>302</td>
<td>Found. (Used for Redirect)</td>
</tr>
<tr>
<td>304</td>
<td>Not Modified. Conditional GET</td>
</tr>
<tr>
<td>400</td>
<td>Bad Request</td>
</tr>
<tr>
<td>401</td>
<td>Unauthorized</td>
</tr>
<tr>
<td>403</td>
<td>Forbidden</td>
</tr>
<tr>
<td>404</td>
<td>Not Found</td>
</tr>
<tr>
<td>405</td>
<td>Method Not allowed. Control Flow</td>
</tr>
<tr>
<td>409</td>
<td>Conflict</td>
</tr>
<tr>
<td>500</td>
<td>Internal Server Error</td>
</tr>
<tr>
<td>501</td>
<td>Not Implemented</td>
</tr>
<tr>
<td>503</td>
<td>Service unavailable</td>
</tr>
<tr>
<td>....</td>
<td>....</td>
</tr>
</tbody>
</table>
Let’s talk Interfaces
GET /checkvist.com/checklists/440751

Checkvist

Filter list by #tag, ^due or keywords

import export copy shared (you + 1) notify actions

Spek

EAP Release

Blog Post

Get Maven Working
Resource Representation
Content Negotiation

URL Extension

http://domain.com/customer/25.json

http://domain.com/news/article1.es

URL Query Parameters

http://domain.com/customer/25?format=json

Accept Headers

Accept: text/html; q=0.7, application/xml; q=1.0
Accept-Language: en-gb; q=0.9, en-us
Demo

Single Entry. Single Representation
Demo

Manual Content Negotiation
Demo

Auto Content Negotiation
Performance and Reliability
Making things reliable

Safe and Idempotent Verbs

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>Idempotent</th>
<th>Safe</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPTIONS</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>GET</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>HEAD</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>PUT</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>POST</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>DELETE</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>PATCH</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Caching Built In

- Max-Age
- Expires
- Conditional GETs (ETags)
Demo

E-Tags
Let’s talk Program State
HATEOAS
Hypermedia
The All-New Kindle

Order from Amazon.co.uk: £109 Wi-Fi | £149 Free 3G+Wi-Fi

What Other Customers Are Looking At Right Now

Cabela's North American Adventures
Activision Publishing
Nintendo Wii

Inception
Leonardo DiCaprio, Ken Watanabe, ...
Blu-ray

Super Street Fighter IV
Capcom
PLAYSTATION 3
$39.99 $24.50

Duracell myGrid Cell Phone Starter Kit
$69.99 $69.99
Your Shopping Cart is empty.

Your Shopping Cart lives to serve. Give it purpose--fill it with books, CDs, videos, DVDs, toys, electronics, and more.

To put something in your Shopping Cart, start by searching or browsing through any of our stores. When an item interests you, click the Add to Shopping Cart button.

* Continue shopping on the Amazon.com home page.

**About the Shopping Cart**

- Items in your Shopping Cart always reflect the most recent price displayed on their product pages.
- [Learn more](#) about the Shopping Cart and how to buy items at Amazon.com

Do you have a gift card or promotional code? We'll ask you to enter your claim code when it's time to pay.
POST /domain.com/order

201 Created
Location: http://domain.com/order/20
Allow: GET, PUT

<order>
  <id>20</id>
  <description>MacBook Pro 13" Retina</description>
  <link rel="payment">/payment/order/20</link>
</order>
Media Types

• XML and Microformats

• Existing Formats: ATOM

• Custom Formats:

  Content-Type: application/vnd.company.doman+xml

• JSON:

  • HAL

  • JSON-DL
State Maintenance

- State is pushed to client
- Server cannot maintain state
- What about Cookies?
Added Benefit of Discoverability

• Known Entities – It’s all a resource
• Known Operations – Constraint on Verbs
• Known Representations – Same Resource
• Hypermedia Navigation – Know next steps
Let’s talk Versioning
Versioning

- URL Versioning
  
  GET /domain.com/api/v1/customer

- Custom Header
  
  X-Version: 2

- Accept Header
  
  Accept: application/vnd.mytype.v2+json
Let’s talk Security
Options

- HTTP
- HTTPS
- Digest
- OAuth
- Federated Security
Let’s talk ReST
Summary

• HTTP API == Application Interface

• You can have HTTP API and not be ReSTful

• ReSTFul systems over certain benefits when abiding by certain constraints
Recommended Reading
Thank you

@hhariri

mail@hadihariri.com