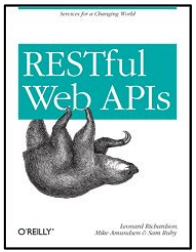
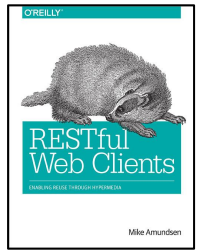
Designing

Mike Amundsen
@mamund

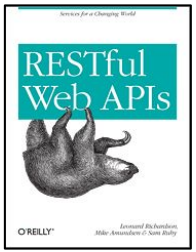
Designing and Building Great APIs





Designing Great APIs

- The Power of Design
- Typical Process
- API Design Method

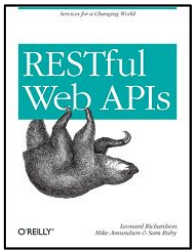


The Power of Design

#mcaTravels

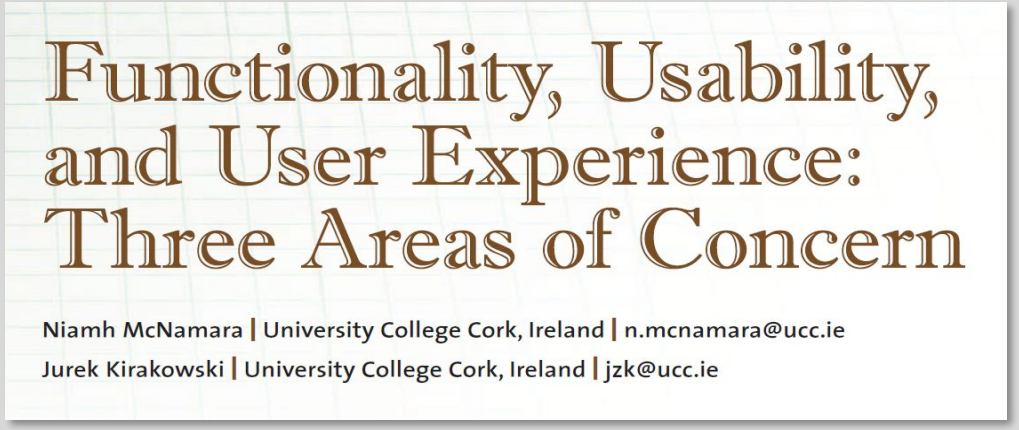
@mamund

#perth2018



“Few people think about it or are aware of it. But there is nothing made by human beings that does not involve a design decision somewhere.”

Bill Moggridge
Interaction Design Pioneer



design

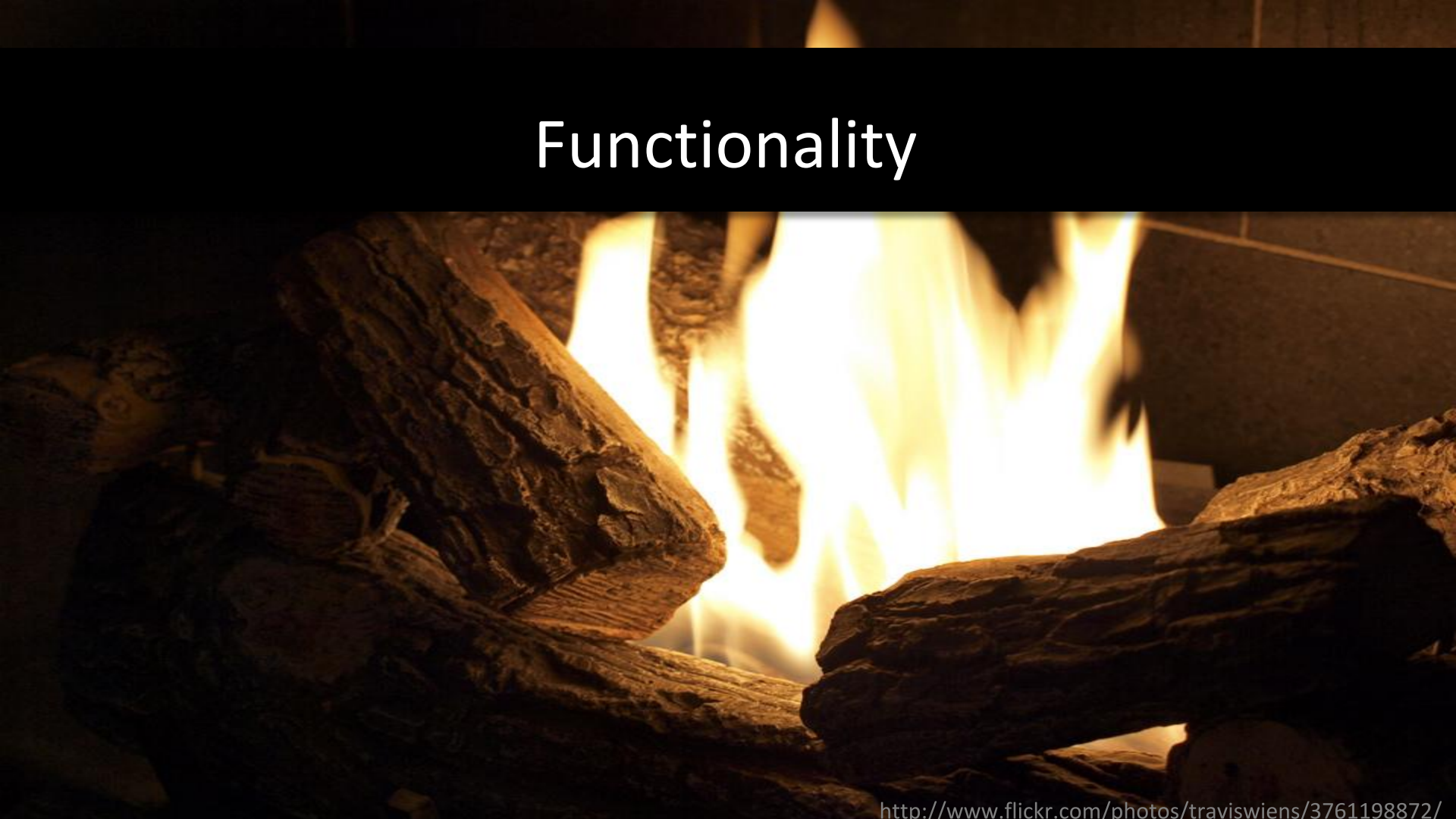
Functionality

Usability

Experience

http://www.flickr.com/photos/monicamuller/3171329060

Functionality

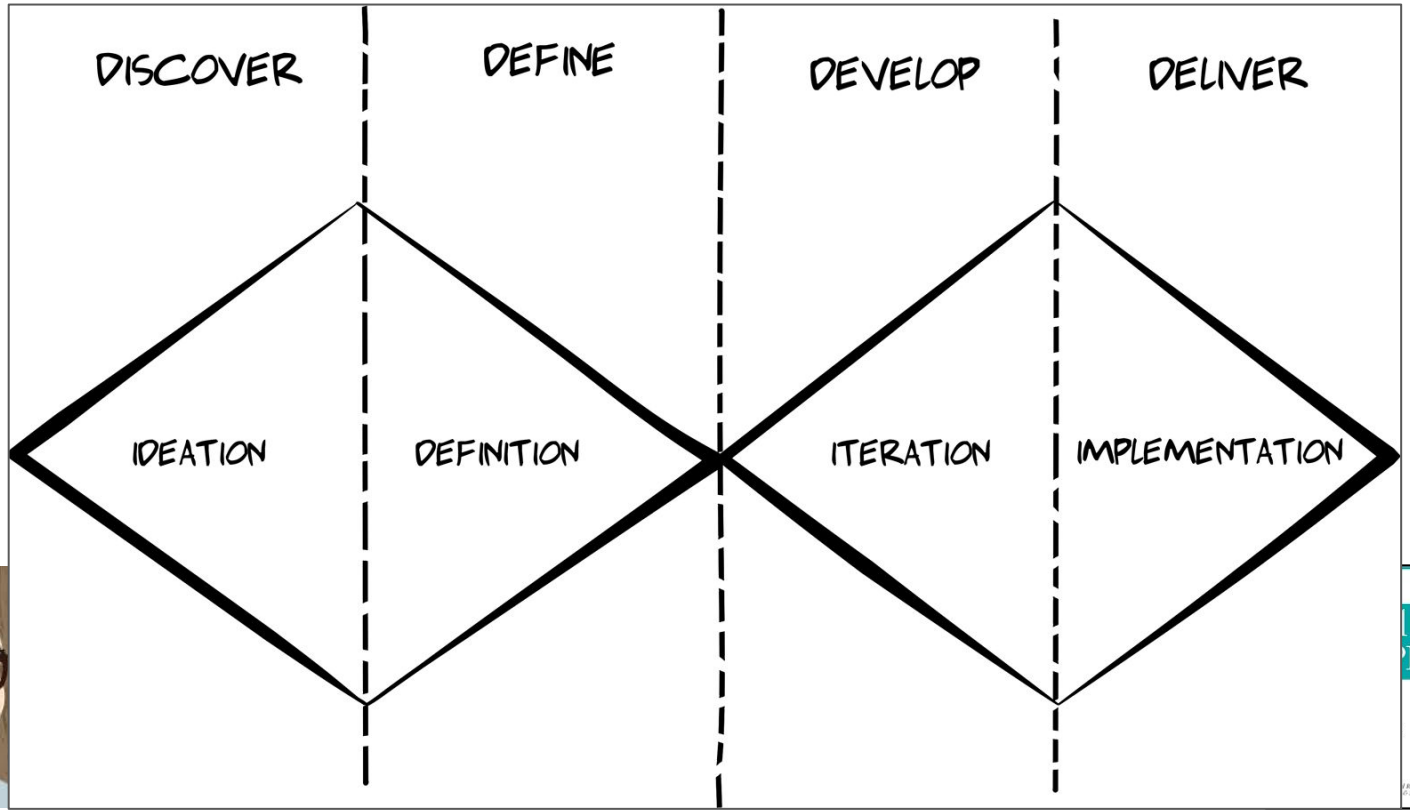


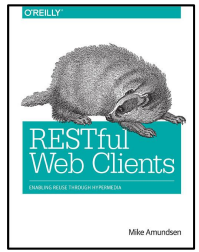
http://www.flickr.com/photos/traviswiens/3761198872/

Usability

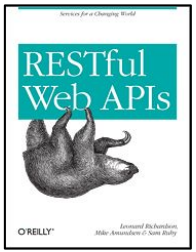
Experience

Double Diamond of Design





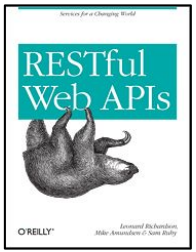
Typical Process



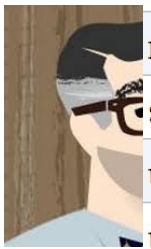
Design Guidelines

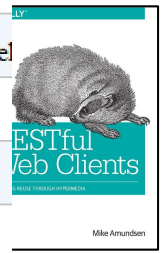
- Craft [good/pretty/usable/stable] URIs

#mcaTravels **@mamund** #perth2018



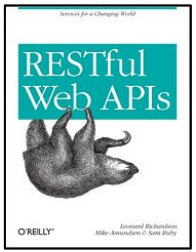
Term	Description
Authority	A URI component that identifies the party with jurisdiction over the namespace defined by the remainder of the URI.
Collection	A resource archetype used to model a server-managed *directory* of resources.
Controller	A resource archetype used to model a procedural concept.
CRUD	An acronym that stands for the four classic storage-oriented functions: create, retrieve, update, and delete.
Developer portal	A Web-based graphical user interface that helps a REST API acquire new clients.
Docroot	A resource that is the hierarchical ancestor of all other resources within a REST API's model. This resource's URI should be the l advertised entry point.
Document	A resource archetype used to model a singular concept.
Forward slash separator (/)	Used within the URI path component to separate hierarchically related resources.
Opacity of URIs	An axiom, originally described by Tim Berners-Lee, that governs the visibility of a resource identifier's composition.
Parent resource	The document, collection, or store that governs a given subordinate concept by preceding it within a URI's hierarchical path.
Query	A URI component that comes after the path and before the optional fragment.
Resource archetypes	A set of four intrinsic concepts (document, collection, store, and controller) that may be used to help describe a REST API's model
Store	A resource archetype used to model a client-managed resource repository.
URI path segment	Part of a resource identifier that represents a single node within a larger, hierarchical resource model.
URI template	A resource identifier syntax that includes variables that must be substituted before resolution.





Design Guidelines

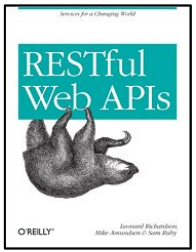
- Craft [good/pretty/usable/stable] URIs
- Map domain actions to HTTP methods (CRUD)



Term	Description
`DELETE`	HTTP request method used to remove its parent.
`GET`	HTTP request method used to retrieve a representation of a resource's state.
`HEAD`	HTTP request method used to retrieve the metadata associated with the resource's state.
`OPTIONS`	HTTP request method used to retrieve metadata that describes a resource's available interactions.
`POST`	HTTP request method used to create a new resource within a collection or execute a controller.
`PUT`	HTTP request method used to insert a new resource into a store or update a mutable resource.
`Request-Line`	RFC 2616 defines its syntax as `Method SP Request-URI SP HTTP-Version CRLF`
Request method	Indicates the desired action to be performed on the request message's identified resource.
Response status code	A three-digit numeric value that is communicated by a server to indicate the result of a client's request.
`Status-Line`	RFC 2616 defines its syntax as: `HTTP-Version SP Status-Code SP Reason-Phrase CRLF`
Tunneling	An abuse of HTTP that masks or misrepresents a message's intent and undermines the protocol's transparency.

Design Guidelines

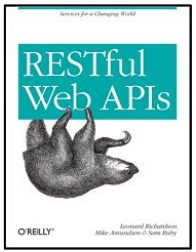
- Craft [good/pretty/usable/stable] URIs
- Map domain actions to HTTP methods (CRUD)
- Use the proper HTTP Status Codes



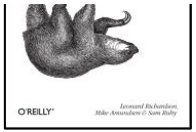
Code	Name	Meaning
400	Bad Request	Indicates a nonspecific client error
401	Unauthorized	Sent when the client either provided invalid credentials or forgot to send them
402	Forbidden	Sent to deny access to a protected resource
404	Not Found	Sent when the client tried to interact with a URI that the REST API could not map to a resource
405	Method Not Allowed	Sent when the client tried to interact using an unsupported HTTP method
406	Not Acceptable	Sent when the client tried to request data in an unsupported media type format
409	Conflict	Indicates that the client attempted to violate resource state
412	Precondition Failed	Tells the client that one of its preconditions was not met
415	Unsupported Media Type	Sent when the client submitted data in an unsupported media type format
500	Internal Server Error	Tells the client that the API is having problems of its own

Design Guidelines

- Craft [good/pretty/usable/stable] URIs
- Map domain actions to HTTP methods (CRUD)
- Use the proper HTTP Status Codes
- Document serialized objects as HTTP bodies

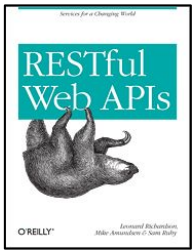


Term	Description
Field	A named slot with some associated information that is stored in its value.
Form	A structured representation that consists of the fields and links, which are defined by an associated schema.
Format	Describes a form's presentation apart from its schematic.
Link	An actionable reference to a resource.
Link formula	A boolean expression that may serve as HATEOAS calculator's input in order to determine the availability of state-sensitive hypermedia within a form.
Link relation	Describes a connection between two resources.
Schema	Describes a representational form's structure independent of its format.
State fact	A Boolean variable that communicates a condition that is relevant to some state-sensitive hypermedia.



Design Guidelines

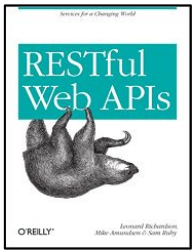
- Craft [good/pretty/usable/stable] URIs
- Map domain actions to HTTP methods (CRUD)
- Use the proper HTTP Status Codes
- Document serialized objects as HTTP bodies
- Use HTTP headers responsibly



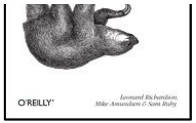
Code	Purpose
`Content-Type`	Identifies the entity body's media type
`Content-Length`	The size (in bytes) of the entity body
`Last-Modified`	The date-time of last resource representation's change
`ETag`	Indicates the version of the response message's entity
`Cache-Control`	A TTL-based caching value (in seconds)
`Location`	Provides the URI of a resource

Design Guidelines

- Craft [good/pretty/usable/stable] URIs
- Map domain actions to HTTP methods (CRUD)
- Use the proper HTTP Status Codes
- Document serialized objects as HTTP bodies
- Use HTTP headers responsibly
- Describe edge cases (async, errors, authN/Z)



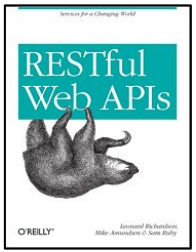
```
# Response
HTTP/1.1 202 Accepted ❶
Content-Type: application/xml;charset=UTF-8
Content-Location: http://www.example.org/images/task/1
Date: Sun, 13 Sep 2009 01:49:27 GMT

<status xmlns:atom="http://www.w3.org/2005/Atom">
  <state>pending</state>
  <atom:link href="http://www.example.org/images/task/1" rel="self"/>
  <message >Your request has been accepted for processing.</message>
  <ping-after>2009-09-13T01:59:27Z</ping-after> ❷
</status>
```



```
# Response
HTTP/1.1 409 Conflict
Content-Type: application/xml;charset=UTF-8
Content-Language: en
Date: Wed, 14 Oct 2009 10:16:54 GMT
Link: <http://www.example.org/errors/limits.html>;rel="help"

<error xmlns:atom="http://www.w3.org/2005/Atom">
  <message>Account limit exceeded. We cannot complete the transfer due to
  insufficient funds in your accounts</message>
  <error-id>321-553-495</error-id>
  <account-from>urn:example:account:1234</account-from>
  <account-to>urn:example:account:5678</account-to>
  <atom:link href="http://example.org/account/1234"
             rel="http://example.org/rels/transfer/from"/>
  <atom:link href="http://example.org/account/5678"
             rel="http://example.org/rels/transfer/to"/>
</error>
```

```
# Request to obtain a request token
POST /request_token HTTP/1.1 ❶
Host: www.example.org
Authorization: OAuth realm="http://www.example.com/photos",
                     oauth_consumer_key=a1191fd420e0164c2f9aeac32ed35d23,
                     oauth_nonce=109843dea839120a,
                     oauth_signature=d8e19bb988110380a72f6ca33b2ba5903272fe1,
                     oauth_signature_method=HMAC-SHA1,
                     oauth_timestamp=1258308730,
                     oauth_version=1.0 ❷
Content-Length: 0

# Response containing a request token and a secret
HTTP/1.1 200 OK
Content-Type: application/x-www-form-urlencoded

oauth_token=0e713d524f290676de8aff4073b1bb52e37f065c
  &oauth_token_secret=394bc633d4c93f79aa0539fd554937760f05987c ❸
```

This is not design!

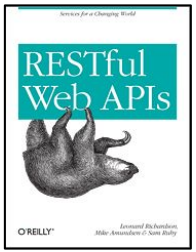


API Design Method

#mcaTravels

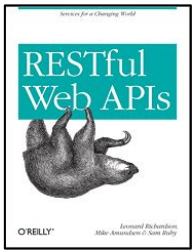
@mamund

#perth2018



API Design Method

- List Semantic Descriptors
- Draw a Diagram
- Reconcile Names
- Write a Profile

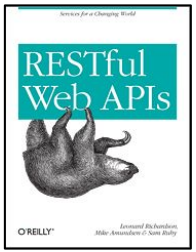


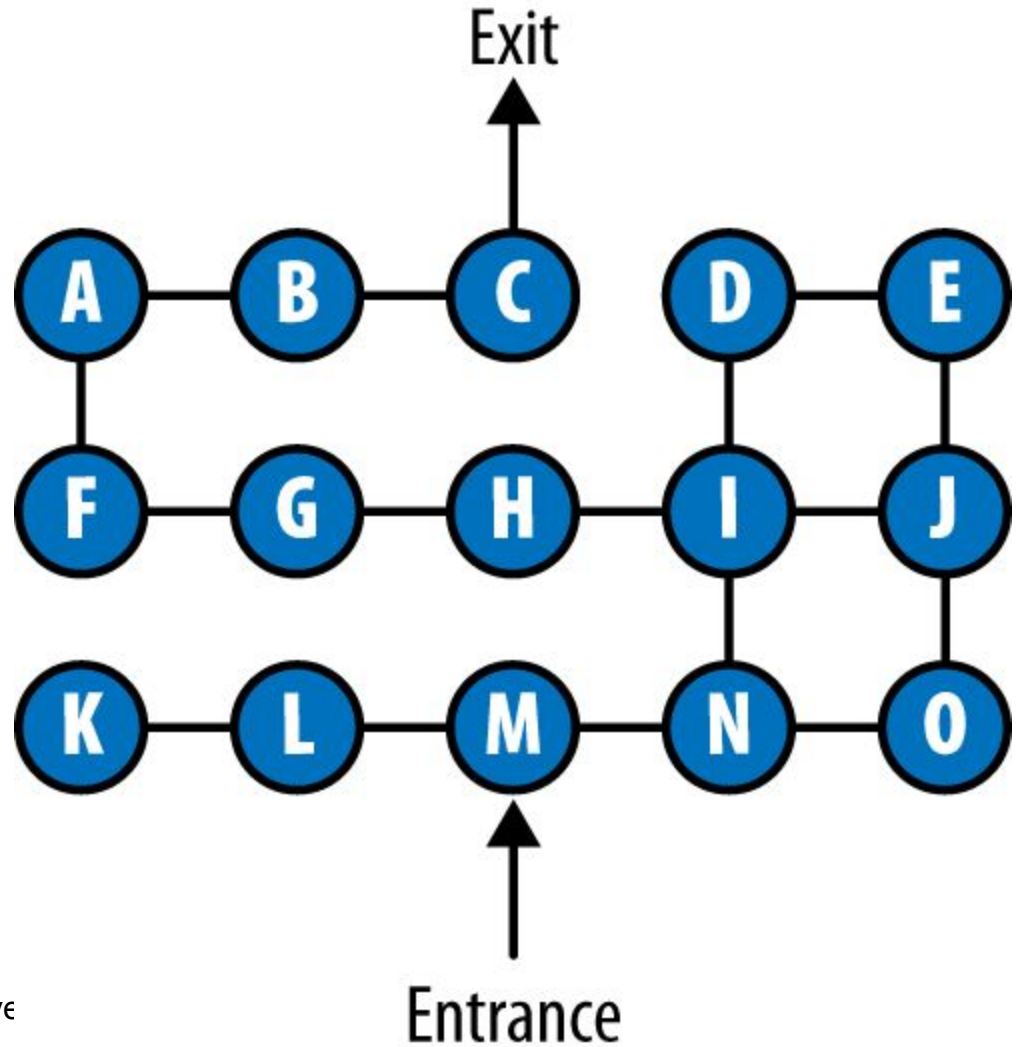
Let's design a Maze game API

#mcaTravels

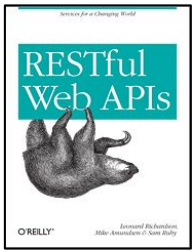
@mamund

#perth2018




Exit
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
Entrance

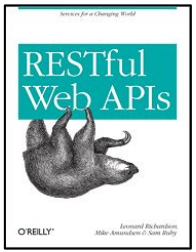
#mcaTrave


RESTful
Web APIs
O'REILLY


O'REILLY
RESTful
Web Clients
Mike Amundsen

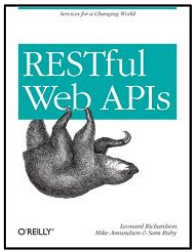
1. List the Semantic Descriptors

#mcaTravels **@mamund** #perth2018



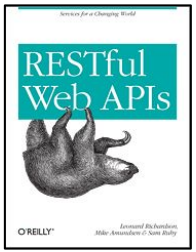
1. List the Semantic Descriptors

(the what?)



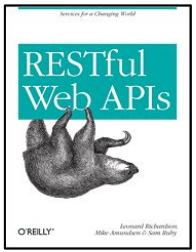
1. List the Semantic Descriptors
(the what?)

You know, the *stuff!*

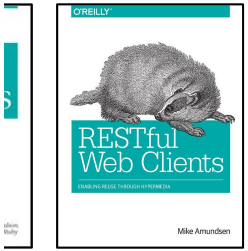


1. List the Semantic Descriptors

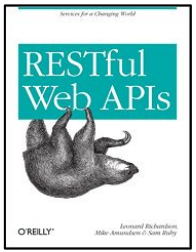
- A maze
- A maze cell
- A switch
- Switch position ("up" or "down")
- The title of a maze cell
- A doorway connecting to cells
- An exit from the maze
- A list of mazes

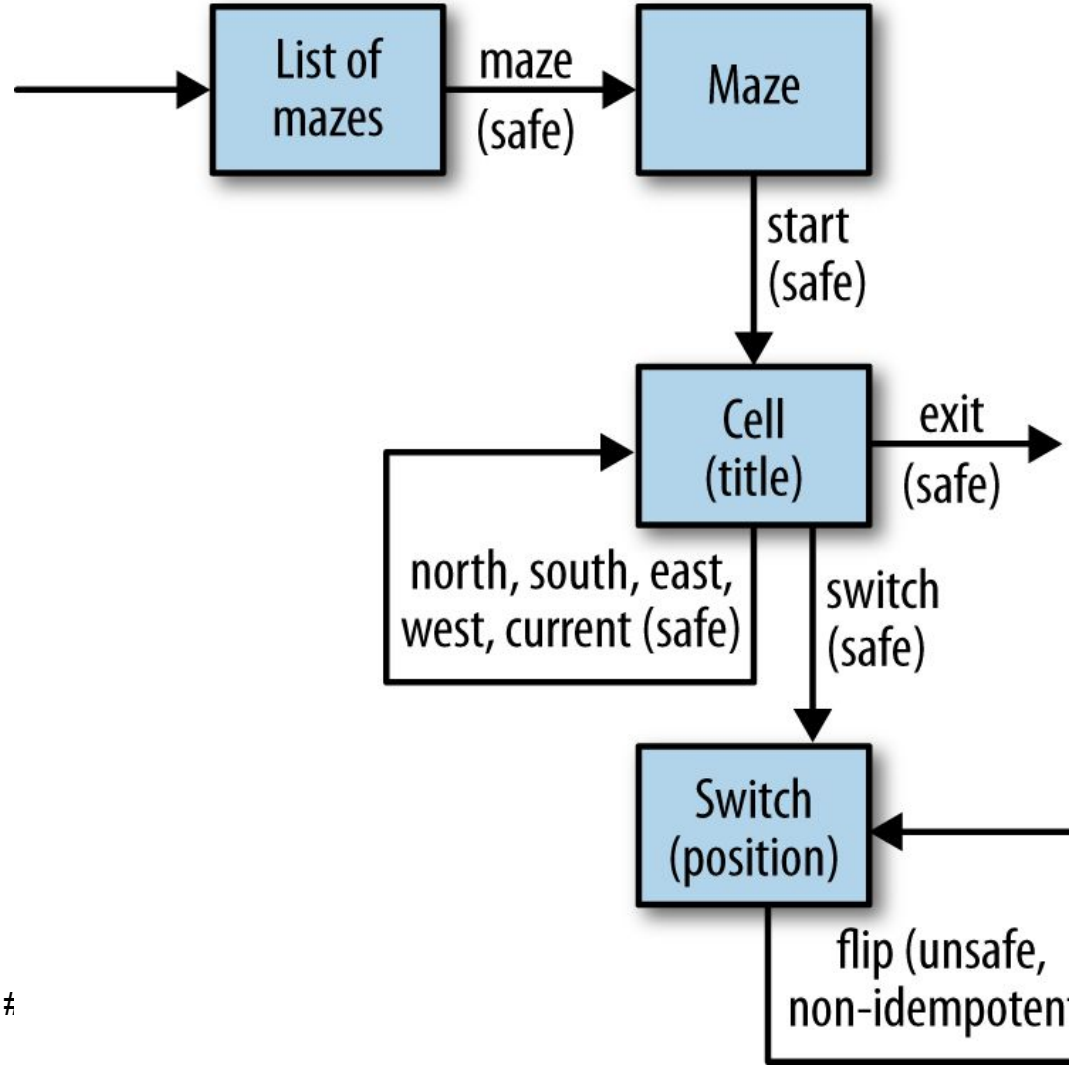


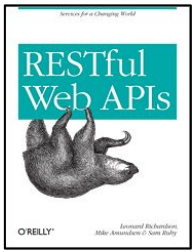
- A list of mazes
  - A maze
    - A maze cell
      - A title
      - A doorway to another cell
      - An exit from the current maze
      - A switch
        - A position ("up" or "down")



2. Draw a Diagram

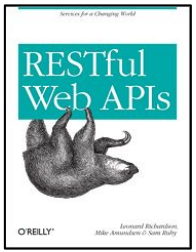



List of mazes
maze
(safe)
Maze
start
(safe)
Cell
(title)
exit
(safe)
north, south, east, west, current (safe)
switch
(safe)
Switch
(position)
flip (unsafe, non-idempotent)


RESTful
Web APIs

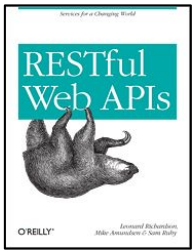

O'REILLY
RESTful
Web Clients
Mike Amundsen

3. Reconcile Names



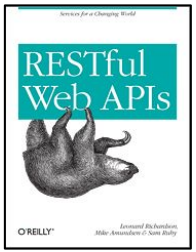
3. Reconcile Names

- IANA Link Relation Values
- schema.org
- microformats
- Dublin Core
- Activity Streams



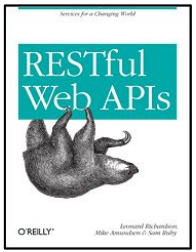
3. Reconcile Names

- maze
- start
- current
- exit
- north, south, east, west
- switch
- flip



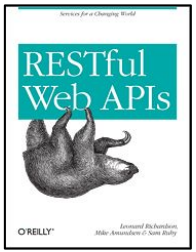
3. Reconcile Names

- maze
- start (IANA)
- current (IANA)
- exit
- north, south, east, west
- switch
- flip



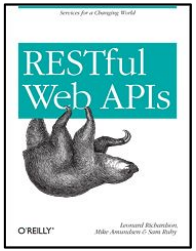
3. Reconcile Names

- maze
- start (IANA)
- current (IANA)
- exit (microformats)
- north, south, east, west (microformats)
- switch
- flip



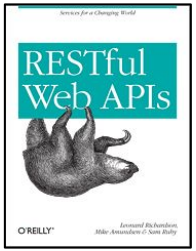
3. Reconcile Names

- maze
- start (IANA)
- current (IANA)
- exit (microformats)
- north, south, east, west (microformats)
- switch
- ~~flip~~ edit (IANA)



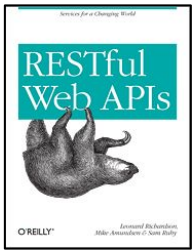
3. Reconcile Names

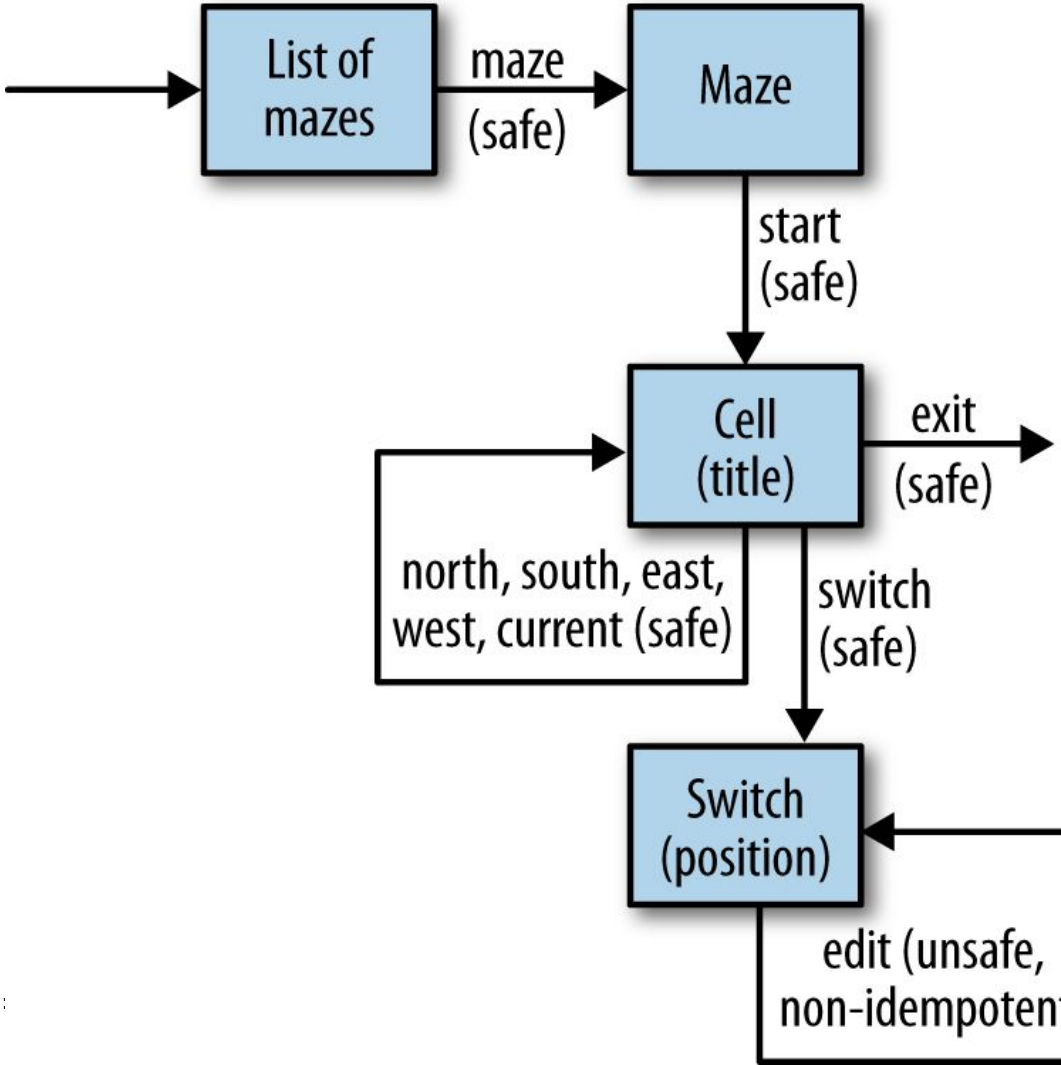
- http://mamund.com/rels/maze (RFC5988)
- start (IANA)
- current (IANA)
- exit (microformats)
- north, south, east, west (microformats)
- http://mamund.com.rels/switch (RFC5988)
- ~~flip~~ edit (IANA)

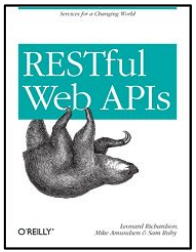


3. Reconcile Names

- IANA
  - edit
  - start
  - current
- microformats
  - exit
  - north, south, east, west
- RFC5988
  - http://mamund.com/rels/switch
  - http://mamund.com/rels/maze




List of mazes
maze (safe)
Maze
start (safe)
Cell (title)
exit (safe)
north, south, east, west, current (safe)
switch (safe)
Switch (position)
edit (unsafe, non-idempotent)


RESTful Web APIs
O'REILLY


O'REILLY
RESTful Web Clients
Mike Amundsen

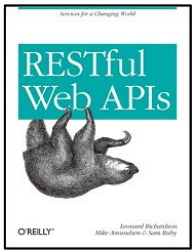
018

4. Write a Profile

#mcaTravels

@mamund

#perth2018



INFORMATIONAL

Independent Submission
Request for Comments: 6906
Category: Informational
ISSN: 2070-1721

E. Wilde
EMC Corporation

March 2013

The 'profile' Link Relation Type

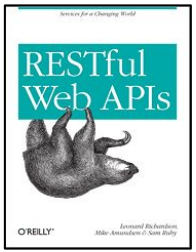
Abstract

This specification defines the 'profile' link relation type that allows resource representations to indicate that they are following one or more profiles. A profile is defined not to alter the semantics of the resource representation itself, but to allow clients to learn about additional semantics (constraints, conventions, extensions) that are associated with the resource representation, in addition to those defined by the media type and possibly other mechanisms.

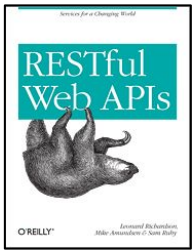
```
<alps version="1.0">
  <link href="http://amundsen.com/media-types/maze/" rel="help" />

  <!-- semantic descriptors -->
  <descriptor id="maze" type="safe" def="RFC5988"/>
  <descriptor id="switch" type="safe" def="RFC5988" />

  <descriptor id="edit" type="safe" def="http://www.iana.org/assignments/link-relations/"/>
  <descriptor id="start" type="safe" def="http://www.iana.org/assignments/link-relations/"/>
  <descriptor id="current" type="safe" def="http://www.iana.org/assignments/link-relations/"/>

  <descriptor id="exit" type="safe" def="http://microformats.org/wiki/existing-rel-values" />
  <descriptor id="north" type="safe" def="http://microformats.org/wiki/existing-rel-values" />
  <descriptor id="south" type="safe" def="http://microformats.org/wiki/existing-rel-values" />
  <descriptor id="west" type="safe" def="http://microformats.org/wiki/existing-rel-values" />
  <descriptor id="east" type="safe" def="http://microformats.org/wiki/existing-rel-values" />
</alps>
```

API Design Method

- List Semantic Descriptors
- Draw a Diagram
- Reconcile Names
- Write a Profile



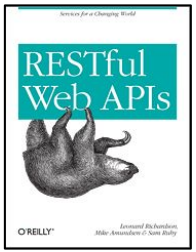
So...


RESTful
Web APIs
O'REILLY

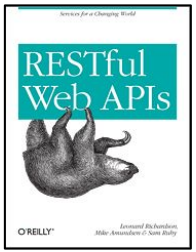

O'REILLY
RESTful
Web Clients
Mike Amundsen

Designing Great APIs

- The Power of Design
- Typical Process
- API Design Method



Exercise


RESTful
Web APIs
O'REILLY


O'REILLY
RESTful
Web Clients
Mike Amundsen

Designing

Mike Amundsen
@mamund

Designing and Building Great APIs