

# Hypermedia Hacking Workshop

Mike Amundsen  
API Academy at CA Technologies  
@mamund

# The JSON Profile Client

# What are Display Profiles?

- What is the metadata for *displaying* an object?
  - Value
  - Prompt
  - Name
  - Id
  - Rendering hints (display type, hidden, etc.)
- IOW, the programming we do to decide what to display

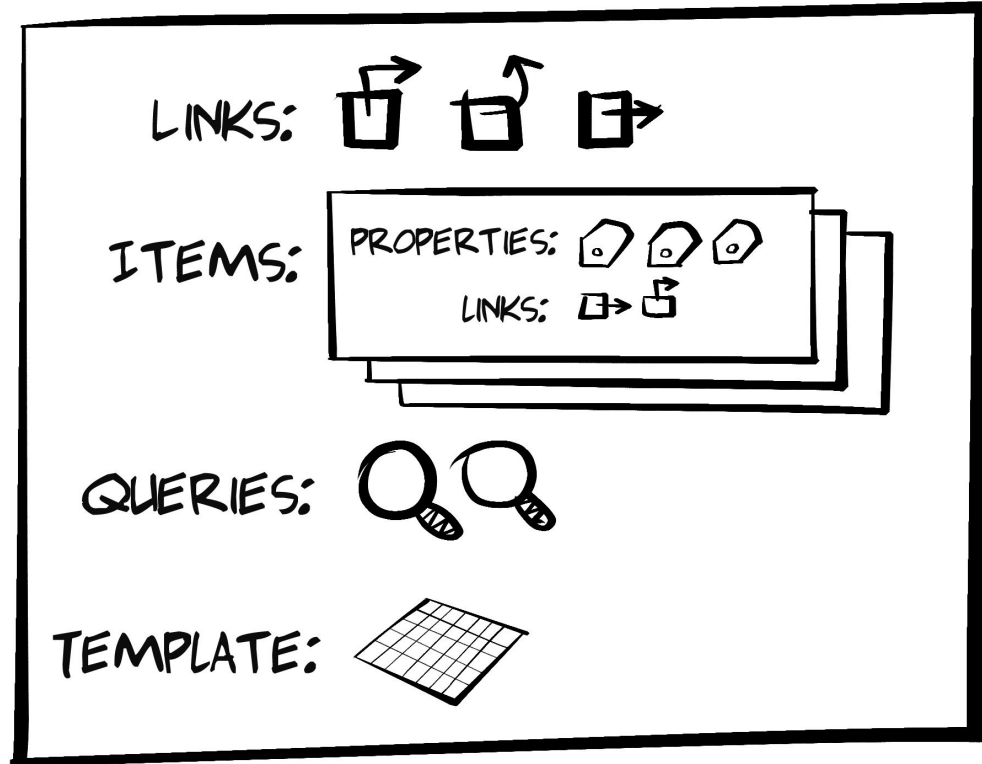
# The Collection+JSON Media Type

Mike Amundsen, 2011

*"[Cj] is similar to the Atom Syndication Format (RFC4287) and the Atom Publishing Protocol (RFC5023). However, Collection+JSON defines both the format and the protocol semantics in a single media type. [Cj] also includes support for Query Templates and expanded write support through the use of a Write Template."*

<http://amundsen.com/media-types/collection>

# The Collection+JSON Media Type



```

{
  "collection": {
    "version": "1.0",
    "href": "http://rwcbook12.herokuapp.com",
    "title": "TPS - Task Processing System",
    "links": [
      {
        "href": "http://rwcbook12.herokuapp.com/",
        "rel": "collection",
        "prompt": "All task"
      }
    ],
    "items": [
      {
        "rel": "item",
        "href": "http://rwcbook12.herokuapp.com/1sv697h2yij",
        "data": [
          { "name": "id", "value": "1sv697h2yij", "prompt": "id" },
          { "name": "title", "value": "Marina", "prompt": "title" },
          { "name": "completed", "value": "false", "prompt": "completed" }
        ]
      },
      {
        "rel": "item",
        "href": "http://rwcbook12.herokuapp.com/25ogsjhqtk7",
        "data": [
          { "name": "id", "value": "25ogsjhqtk7", "prompt": "id" },
          { "name": "title", "value": "new stuff", "prompt": "title" },
          { "name": "completed", "value": "true", "prompt": "completed" }
        ]
      }
    ],
    "queries": [
      {
        "rel": "search",
        "href": "http://rwcbook12.herokuapp.com/",
        "prompt": "Search tasks",
        "data": [
          { "name": "title", "value": "", "prompt": "Title" }
        ]
      }
    ],
    "template": {
      "prompt": "Add task",
      "rel": "create-form",
      "data": [
        { "name": "title", "value": "", "prompt": "Title" },
        { "name": "completed", "value": "false", "prompt": "Complete" }
      ]
    }
  }
}

```

OK, so let's design our own...

# The POD Spec

## **Profile for Object Description (POD)**

Mike Amundsen 2016

*"This specification is designed to describe the rules for displaying domain objects and their properties in a human-centric user interface."*

<http://rwcbook.github.io/pod-spec/>



# The POD Spec

## A Valid POD Document

```
{
  "user" : {
    "id" : {"prompt" : "ID", "render" : "none"},
    "nick" : {"prompt" : "Nickname", "render" : "text"},
    "email" : {"prompt" : "Email", "render" : "text"},
    "name" : {"prompt" : "Full Name", "render" : "text"},
    "password" : {"prompt" : "Password", "render" : "text"},
    "dateCreated" : {"prompt" : "Date Created", "render" : "none"},
    "dateUpdated" : {"prompt" : "Date Updated", "render" : "none"}
  }
}
```

# **Exercise:**

## Adding POD Support

# Produce POD info

- Update representor
- Emit item w/ POD
  - Name
  - Prompt
  - Render

# Consume POD info

- Update client
- Drop local array
- Replace w/ message data

**Discussion:**  
Can we improve POD?