Hypermedia Made Easy

Mike Amundsen
@mamund
youtube.com/mamund
g.mamund.com/GreatWebAPIs

"From design to code to test to deployment, unlock hidden business value and release stable and scalable web APIs that meet customer needs and solve important business problems in a consistent and reliable manner."

-- Pragmatic Publishers (2020)
With this practical guide, you’ll learn how to move from one-off implementations to general-purpose client apps that are stable, flexible, and reusable.

-- O'Reilly Media (2017)
Hypermedia Made Easy : The Big Picture

● Goals
● Consumer View
● Provider View
Hypermedia Made Easy: Goals

- Improve Reach
- Increase Usability
- Reduce Code
Hypermedia Made Easy : Goals

- Improve Reach
  - HTML/HTTP has massive reach; why?
  - Low barrier of entry, high utility
- Increase Usability
- Reduce Code
Hypermedia Made Easy: Goals

- Improve Reach
- Increase Usability
  - HTML is about documents, not content.
  - Good hypermedia design is about communication
- Reduce Code
Hypermedia Made Easy : Goals

- Improve Reach
- Increase Usability
- Reduce Code
  - What is not in the message must be in the code
  - Messages are more easily shared than code
Consumer View
Hypermedia Made Easy: Consumer View

- Quick Tour of Cj
- The Cj SPA client
- Check out the code
TPS - Tasks

Manage your TPS Tasks here.

You can do the following:
- Add, Edit and Delete tasks
- Mark tasks "complete", assign tasks to a user
- Filter the list by Title, Assigned User, and Completed Status

**Add Task**

<table>
<thead>
<tr>
<th>Title</th>
<th>Tags</th>
<th>Complete</th>
<th>Assigned User</th>
</tr>
</thead>
<tbody>
<tr>
<td>extensions</td>
<td>forms testing</td>
<td>No</td>
<td>mamund</td>
</tr>
</tbody>
</table>

**Completed Tasks**

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Tags</th>
<th>Complete</th>
<th>Assigned User</th>
</tr>
</thead>
<tbody>
<tr>
<td>1m80x2xpuv5</td>
<td>Run client-side tests</td>
<td>test</td>
<td>true</td>
<td>alice</td>
</tr>
</tbody>
</table>

**Queries**

copyright © 2020 by amundsen.com, inc. -- all rights reserved
Hypermedia Made Easy: Cj Tour

```
{ 
  - collection: {
    version: "1.0",
    title: "TPS - Tasks",
    content: "<h3>Manage your TPS Tasks here. </h3><p>You can do the following:</p><ul><li>Add, Edit and Delete tasks</li><li>Mark tasks "complete", assign tasks to a user</li><li>Filter the list by Title, Assigned User, and Completed Status</li></ul>"
  ,
  - links: [
    - {
      rel: "home collection",
      prompt: "Home"
    },
    - {
      rel: "self task collection",
      prompt: "Tasks"
    }
  ]
}
```
Hypermedia Made Easy: Cj Tour

```json
{
  "collection": {
    "version": "1.0",
    "href": "//rwcbbook14.herokuapp.com/task/",
    "title": "TPS - Tasks",
    "content": "<h3>Manage your TPS Tasks here.</h3><p>You can do the following:</p><ul>
  <li>Add, Edit and Delete tasks</li>
  <li>Mark tasks "complete" action tasks to a week</li>
  <li>Filter the Status</li>
</ul>"
  
  "links": [
    {
      "href": "",
      "rel": "home",
      "prompt": "Home"
    },
    {
      "href": "",
      "rel": "search",
      "prompt": "Search"
    },
    {
      "href": "",
      "rel": "settings",
      "prompt": "Settings"
    },
    {
      "href": "",
      "rel": "tasks",
      "prompt": "Tasks"
    },
    {
      "href": "",
      "rel": "profile",
      "prompt": "Profile"
    }
  ],
  "data": [
    {
      "name": "id",
      "value": "19f7bhaoh",
      "prompt": "ID",
      "display": "true"
    },
    {
      "name": "title",
      "value": "extensions",
      "prompt": "Title",
      "display": "true"
    },
    {
      "name": "tags",
      "value": "forms testing",
      "prompt": "Tags",
      "display": "true"
    }
  ]
}
```
Hypermedia Made Easy: Cj Tour
Hypermedia Made Easy: Cj Tour
Hypermedia Made Easy: SPA Markup
function cj() {
    var d = domHelp();
    var g = {};

    g.url = '';
    g.cj = null;
    g.cetype = "application/vnd.collection+json";

    // init library and start
    function init(url, title) {
        if(!url || url==="") {
            alert("*** ERROR:\n\nMUST pass starting URL to the Cj library");
        }
        else {
            g.url = url;
            g.title = title||"Cj Client";
            req(g.url,"get");
        }
    }

    // primary loop
    function parseCj() {
        dump();
        title();
        content();
        links();
        items();
        queries();
        template();
        error();
        cjClearEdit();
    }
}
Hypermedia Made Easy: Consumer View

● Quick Tour of Cj
  ○ Highly structured message type
  ○ meta, links, items, queries, template, error

● The Cj SPA client
  ○ < 100 lines of semantic HTML
  ○ Easily styled via semantic css

● Check out the code
  ○ < 400 lines of JS
  ○ Easy to add support for extensions
Provider View
Hypermedia Made Easy: Provider View

- Quick Tour of NodeJS Server
- DARRT model for implementation
- Decoupled and extensible
// shared vars
var root = '';
var port = (process.env.PORT || 8181);
var htmlType = "text/html";
var jsonType = "application/json";
var halType = "application/vnd.hal+json";
var wstlType = "application/vnd.wstl+json";
var sirenType = "application/vnd.siren+json";
var cjType = "application/vnd.collection+json";
var csType = '';
var csAccept = '';

// routing rules
var reRoot = new RegExp('\\/$', 'i');
var reFile = new RegExp('\\/files\\/*', 'i');
var reHome = new RegExp('\\/home\\/*', 'i');
var reTask = new RegExp('\\/task\\/*', 'i');
var reUser = new RegExp('\\/user\\/*', 'i');
var reNote = new RegExp('\\/note\\/*', 'i');
var reWstl = new RegExp('\\/wstl\\/*', 'i');

// request handler
function handler(req, res) {
    var segments, i, x, parts, rtn, flg, doc, url;
```javascript
function main(object, mimeType, root) {
    var doc;

    // clueless? assume JSON
    if (!mimeType) {
        mimeType = defaultFormat;
    }

    // dispatch to requested representor
    switch (mimeType.toLowerCase()) {
        case "application/vnd.wstl+json":
            doc = wstljson(object, root);
            break;
        case "application/json":
            doc = json(object, root);
            break;
        case "application/vnd.wstl+hal+json":
            doc = haljson(object, root);
            break;
        case "application/vnd.siren+json":
            doc = siren(object, root);
            break;
        case "application/vnd.collection+json":
            doc = cj(object, root);
            break;
        default:
            doc = cj(object, root);
            break;
    }
    return JSON.stringify(doc, null, 2);
}
```
Hypermedia Made Easy: Extensibility

```javascript
function updateTask(elm, id, task, props) {
  var rtn, check, item;
  check = storage(elm, 'item', id);
  if(check !== null) {
    rtn = utils.exception("File Not Found", "No record on file", 404);
  } else {
    item = check;
    item.id = id;
    item.title = (task.title === undefined ? check.title : task.title);
    item.tags = (task.tags === undefined ? check.tags : task.tags);
    item.assignedUser = (task.assignedUser === undefined ? check.assignedUser : task.assignedUser);
    item.completeFlag = (task.completeFlag === undefined ? check.completeFlag : task.completeFlag);
    if(item.title === "") {
      rtn = utils.exception("Missing Title");
    } else {
      storage(elm, 'update', id, utils.setProps(item, props));
    }
  }
  return rtn;
}

function removeTask(elm, id) {
  var rtn, check;
  check = storage(elm, 'item', id);
  if(check !== null) {
    rtn = utils.exception("File Not Found", "No record on file", 404);
  } else {
    storage(elm, 'remove', id);
  }
}
```
Hypermedia Made Easy: Provider View

● Quick Tour of NodeJS Server
  ○ Relies on representer pattern for responses
  ○ Common internal model, negotiated external message formats

● DARRT model for implementation
  ○ Data, Action, Resource, Representation, Transitions
  ○ DA are internal RRT are external

● Decoupled and extensible
  ○ Internal and external models are loosely coupled
  ○ Easy to add support for new objects (internal) and messages (external)
And So ...
Hypermedia Made Easy

● Goals
● Consumer View
● Provider View
Hypermedia Made Easy

● Goals
  ○ Extend reach through structured messages
  ○ More in the message means less in the code
● Consumer View
● Provider View
Hypermedia Made Easy

- **Goals**

- **Consumer View**
  - Clients are tightly bound to messages, not content
  - Easy to extend as features are added to the message model

- **Provider View**
Hypermedia Made Easy

● Goals
● Consumer View
● Provider View
  ○ DARRT model for server implementation
  ○ Decouple internal models (DA) from external models (RRT)
Hypermedia Made Easy: Futures

We need more ...

- Highly structured message models (media types)
- Generic clients (no code, just messages)
- Negotiated vocabularies (ALPS)
- Runtime discovery (openDISCO)
That's all there is!
Resources

- "Design and Build Great Web APIs"
  [g.mamund.com/greatwebapis](g.mamund.com/greatwebapis)

- "RESTful Web Clients"
  [g.mamund.com/rwcbook](g.mamund.com/rwcbook)

- This talk (slides, links, etc.)
  [g.mamund.com/2020-05-oslcfest](g.mamund.com/2020-05-oslcfest)
Hypermedia Made Easy

Mike Amundsen
@mamund
youtube.com/mamund