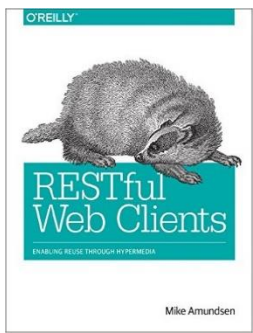
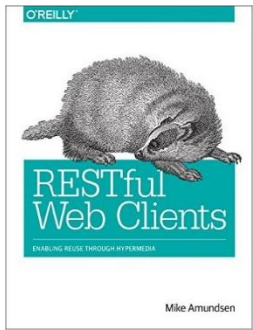
Processing Representations

Mike Amundsen
API Academy
@mamund

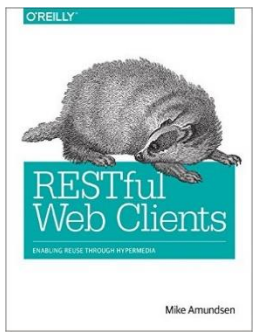


Highlights

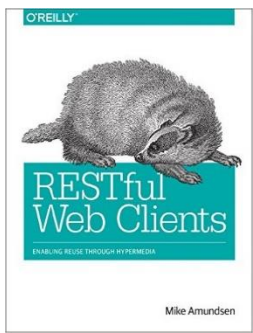
- Media Types, Not Object Types
- Collection+JSON
- Client-Side Representors



Media Types, Not Object Types



Collection+JSON Media Type

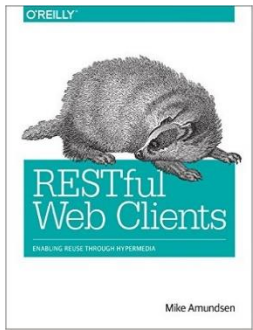


Collection+JSON

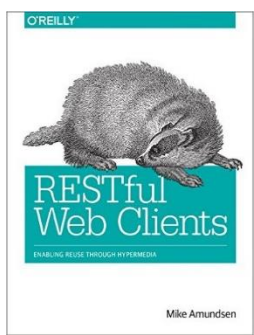
- **Collection+JSON** media type
- First designs in early 2011, registered w/ IANA mid 2011
- *“…a JSON-based read/write hypermedia-type designed to support management and querying of simple collections.”*
- It’s Atom w/ LT + templated writes

Contents

NOTE:
The key words "MUST", "MUST NOT", "
"RECOMMENDED", "MAY", and "OPTI



```
{
  - collection: {
        version: "1.0",
        href: "//rwcbook12.herokuapp.com/home/",
        title: "TPS - Task Processing System",
        content: "<div class="ui segment"><h3>Wel
      + links: […],
        items: [ ],
        queries: [ ],
      - template: {
            data: [ ]
        }
    }
}
```



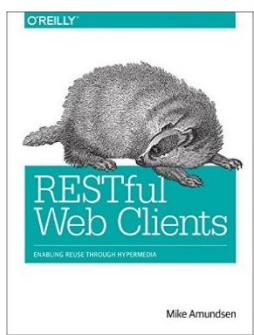
```
- links: [
    - {
          href: "http://rwcbook12.herokuapp.com/home/",
          rel: "self home collection",
          prompt: "Home"
      },
    - {
          href: "http://rwcbook12.herokuapp.com/task/",
          rel: "task collection",
          prompt: "Tasks"
      },
    - {
          href: "http://rwcbook12.herokuapp.com/user/",
          rel: "user collection",
          prompt: "Users"
      }
  ],
```

```
- items: [
    - {
          rel: "item",
          href: "//rwcbook12.herokuapp.com/task/1l9fz7bhaho",
        - data: [
            - {
                  name: "id",
                  value: "1l9fz7bhaho",
                  prompt: "ID",
                  display: "true"
              },
            - {
                  name: "title",
                  value: "extensions",
                  prompt: "Title",
                  display: "true"
              },
            - {
                  name: "tags",
                  value: "forms testing",
                  prompt: "Tags",
                  display: "true"
              },
            - {
                  name: "completeFlag",
                  value: "true",
                  prompt: "Complete Flag",
                  display: "true"
```

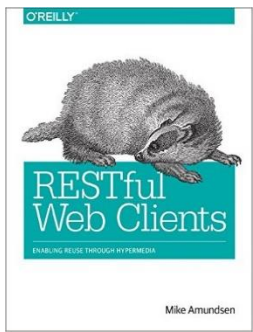
```
- queries: [
    - {
        rel: "completed search",
        href: "http://rwcbook12.herokuapp.com/task/",
        prompt: "Completed Tasks",
      - data: [
          - {
              name: "completeFlag",
              value: "true",
              prompt: "Complete",
              required: false,
              readOnly: true,
              patttern: ""
            }
        ]
    },
    - {
        rel: "active search",
        href: "http://rwcbook12.herokuapp.com/task/",
        prompt: "Active Tasks",
      - data: [
          - {
              name: "completeFlag",
              value: "false",
              prompt: "Complete",
              required: false,
              readOnly: true,
              patttern: ""
```

```
- template: {
      prompt: "Add Task",
      rel: "create-form //rwcbook12.herokuapp.com/rels/taskAdd",
    - data: [
        - {
            name: "title",
            value: "",
            prompt: "Title",
            required: true,
            readOnly: false,
            patttern: ""
          },
        - {
            name: "tags",
            value: "",
            prompt: "Tags",
            required: false,
            readOnly: false,
            patttern: ""
          },
        - {
            name: "completeFlag",
            value: "false",
            prompt: "Complete",
            required: false,
            readOnly: false,
            patttern: "true|false"
          }
```

```
{
  - collection: {
        version: "1.0",
        href: "//rwcbook12.herokuapp.com/home/",
        title: "TPS - Task Processing System",
        content: "<div class="ui segment"><h3>Wel
      + links: […],
        items: [ ],
        queries: [ ],
      - template: {
            data: [ ]
        }
    }
}
```

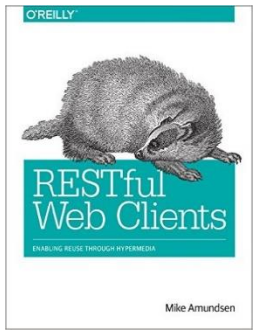


Client-Side Representor



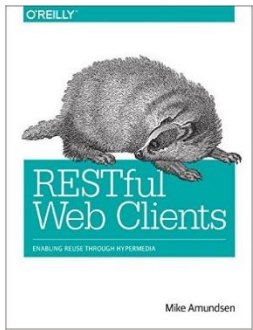
Client-Side Representor

- Request Resource (`accept:application/vnd.collection+json`)
- Parse Response (`content-type:application/vnd.collection+json`)
- Render Results (Convert Cj -> HTML-DOM)



Request Resource

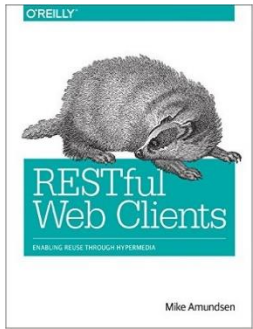
```
435  // low-level HTTP stuff
436  function req(url, method, body) {
437    var ajax = new XMLHttpRequest();
438    ajax.onreadystatechange = function(){rsp(ajax)};
439    ajax.open(method, url);
440    ajax.setRequestHeader("accept",g.ctype);
441    if(body && body!==null) {
442      ajax.setRequestHeader("content-type", g.ctype);
443    }
444    ajax.send(body);
445  }
446  function rsp(ajax) {
447    if(ajax.readyState===4) {
448      g.cj = JSON.parse(ajax.responseText);
449      parseCj();
450    }
451  }
452
```



Parse Response

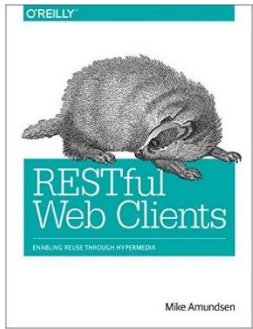
```
// primary loop
function parseCj() {
  dump();
  title();
  links();
  items();
  queries();
  template();
  error();
  cjClearEdit();
}

// handle response dump
function dump() {
  var elm = d.find("dump");
  elm.innerText = JSON.stringify(g.cj, null, 2);
}
```



Render HTML-DOM

```
coll = g.cj.collection.queries;
for(var query of coll) {
  li = d.node("li");
  form = d.node("form");
  form.action = query.href;
  form.className = query.rel;
  form.method = "get";
  form.onsubmit = httpQuery;
  fs = d.node("fieldset");
  lg = d.node("legend");
  lg.innerHTML = query.prompt + " ";
  d.push(lg,fs);
  for(var data of query.data) {
    p = d.input({prompt:data.prompt,name:data.name,
    d.push(p,fs);
  }
  p = d.node("p");
  inp = d.node("input");
  inp.type = "submit";
```



ORM Hyper-Tasks

All task

Item Edit Delete
Title : Danny Boyz
Completed : true

Item Edit Delete
Title : new stuff
Completed : true

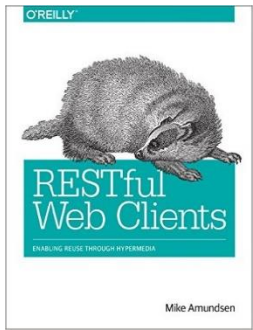
Item Edit Delete
Title : hockey
Completed : false

Item Edit Delete
Title : Marina
Completed : false

Item Edit Delete
Title : new stuff
Completed : true

Summary

- Hypermedia clients process media types
- Collection+JSON is a hypermedia type
- Our client-side representor converts Cj into HTML DOM



Processing Representations

Mike Amundsen
API Academy
@mamund