adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 28-30 SEPTEMBER 2015

Hypermedia API Tools for Sling (HApi)

Andrei Dulvac, Adobe

ToC

- HatEoAS, Hypermedia formats, and semantic data
- Hypermedia API tools (HApi) overview
- Publick example
- Demo
- Q & (hopefully) A

HatEoAS, Hypermedia formats, and semantic data - **quick** overview

HatEoAS

- Hypermedia as the Engine of Application State
  - Constraint of REST
  - Use hypermedia controls to drive the app

Hypermedia API

- API
  - machine-parsable
  - formal description.
- Hypermedia
  - Drive the application with hypermedia controls.
- HTML can work if we add semantic extensions

Semantic HTML

- HTML has semantics built-in; we need extensions
- Microformats
- RDFa
- **Microdata**
- Other
- …

HApi overview

What is HApi (1/2)

- Hypermedia API tools for Apache Sling
- *Annotates* HTML with domain-specific semantics
- Custom Types in the Sling repository
- Works with Sightly and JSP

What is HApi (2/2)

- Outputs semantic extensions into the markup
- Current implementation for attributes
  - Microdata
  - Can be swapped w\o changing the server-side code
- Type-inference for properties
- Validation of properties

What is it good for?

- Creating a REST API
- Make existing markup machine-parsable
- Easy tests for Sling web apps
  - minimum dependencies on the client-side
  - Resilient to code changes (prop validation at script compile)
  - Other advantages that formats like JSON have
- Avoiding code duplication
  - Don’t need a servlet for HTML and one for “everything else”

Components of HApi

- `HApiUtil` – OSGi service
  - Parameterized with a type from the repository
  - `getHelper`() – provides an `AttributeHelper`
  - Current implementation - Microdata
- `AttributeHelper` – output HTML attributes for
  - Elements describing an *object*
  - Elements describing *properties* of an *object*
- `HApiUse` – Sightly *Use* object

Example – Publick– app overview

Example – Publick– the semantic types

https://github.com/dulvac/publick-sling-blog adaptTo()

The HApi types (1/4)

- jcr_root/libs/publick/types/blog_post.json

```
{
  "jcr:primaryType": "nt:unstructured”,"description": "A blog post",
   "fqdn": "org.apache.sling.publick.blog_post",
   "name": "blog_post”, "parameters": [],
   "sling:resourceType": "sling/hapi/components/type",

 "title": {
          "jcr:primaryType": "nt:unstructured",
          "description": "The title of the blog post",
          "type": "org.apache.sling.hapi.common.Text",
          "multiple": false
     }, …
```

The HApi types (2/4)

```
…
 "label": {
        "jcr:primaryType": "nt:unstructured",
        "description": "A label or keyword for the blog post",
        "type": "org.apache.sling.hapi.common.Text",
        "multiple": true
 },
 "comment": {
        "jcr:primaryType": "nt:unstructured",
        "description": "A comment for this blog post",
        "type": "/libs/publick/types/comment",
        "multiple": true
 }
```

The HApi types (3/4)

- jcr_root/libs/publick/types/comment.json

```
"jcr:primaryType": "nt:unstructured”,"description": "A comment …",
  "fqdn": "org.apache.sling.publick.comment",
  "name": "comment”, "parameters": [],
  "sling:resourceType": "sling/hapi/components/type",

  "author": {…},
  “time_created”: {},
  “comment_text”: {},
```

The HApi types (4/4)

```
"reply": {
    "jcr:primaryType": "nt:unstructured",
    "description": "A comment that is a reply for this comment",
    "type": "/libs/publick/types/comment”,
    "multiple": true
}
```

Example – Publick– The markup

Annotating the markup (1/2)

- blogView.html

```
<div data-sly-use.hblog="${'org.apache.sling.hapi.HApiUse'
    @type='/libs/publick/types/blog_post'}”
    data-sly-attribute="${hblog.itemtype}”… >
  …
  <h1 data-sly-attribute="${hblog.itemprop.title}">${blog.title}</h1>
  <span data-sly-attribute="${hblog.itemprop.month_created}”> …</span>
```

Annotating the markup (2/2)

- Comments section

```
<ul data-sly-list.comment="${comments.comments}">

 <li data-sly-attribute="${hblog.itemprop.comment}”
   data-sly-use.hcomment= "${'o.a.s.h.HApiUse' @type=hblog.proptype.comment}”
   <span data-sly-attribute="${hcomment.itemprop.time_created}”>…

 <div data-sly-repeat.reply="${comment.replies}”
     data-sly-attribute="${hcomment.itemprop.reply}”
     data-sly-use.hreply=”${'o.a.s.h.HApiUse’@type=hcomment.proptype.reply}”>
   <span data-sly-attribute="${hcomment.itemprop.author}”>…</span>
    …
```

Rendered markup – HTML w/ microdata

- Comments section

```
<ul>
  <li itemprop="comment” itemscope
      itemtype=http://localhost:8080/libs/publick/types/comment.html>
    <span itemprop="time_created" itemscope itemtype=“[…]/text.html”>July 4</span>
    <div itemprop="reply" itemscope
        itemtype=http://localhost:8080/libs/publick/types/comment.html>
      <span itemprop="author" itemscope itemtype=“[…]/text.html”>Fido</span>
 …
```

Demo

Appendix

Hypermedia controls

- The comments <form>

```
<form data-rel="addcomment" method="POST"
action="/bin/publick/addcomment">
  <label>Name</label>
  <input type="text" name="author" class="form-control">
  <label>Comment</label>
  <textarea class="form-control" name="comment"></textarea>
  <button type="submit" class="btn btn-primary">Post</button>
</form>
```

Project location and code samples

- HApi tools
  - https://github.com/apache/sling/tree/trunk/contrib/extensions/hapi
- Publick blog engine fork
  - https://github.com/dulvac/publick-sling-blog
- Python html demo client for microdata
  - https://github.com/dulvac/htmlapi-client-python