

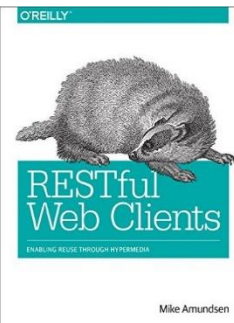
Supporting Workflow with FORMS

Mike Amundsen
API Academy
@mamund



Highlights

- Domain actions vs. Network actions
- Network actions through FORMs
- Cj Queries and Templates



Domain Actions vs. Network Actions



Domain Actions

- The domain is what your service is “about”
 - Accounting
 - Shopping
 - Task Management
- Every domain has it's own set of actions...
 - Search
 - Approve
 - Promote

BTW - POST is not a domain action!

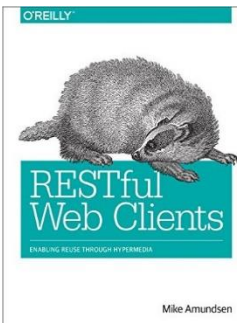


Network Actions

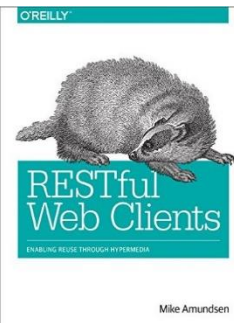
- The network is what your service uses to commit domain actions
 - HTTP
 - WebSockets
 - FTP
- Every NETWORK has it's own set of actions...
 - GET
 - PATCH
 - PUT
 - DELETE
 - OPTIONS
 - POST



Implementing an API means matching domain actions to network actions

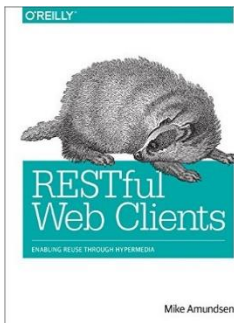


Network Actions through FORMs



FORMs Describe Network Actions

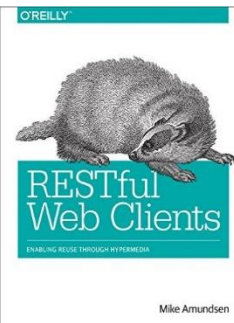
- Domain Identifier (“editTask”)
- Network identifier (<http://example.org/task/q1w2e3>)
- Network action (PUT)
- Domain Arguments
 - ID
 - Title
 - Completed
- Network Format
 - `application/x-www-form-urlencoded`



Example FORM

```
form: {  
  method: "post"  
  href: "http://example.org/task/"  
  encoding: "application/x-www-form-urlencoded"  
  prompt: "Add task",  
  rel: "create",  
  data: [  
    {name: "title", value: "", prompt: "Title"},  
    {name: "completed", value: "false", prompt: "Complete"}  
  ]  
}
```

Cj Queries and Templates



Cj Queries

- Query FORMS are for read-type domain actions
 - Search, Sort, List, Read, etc.
- Query FORMS are always network safe
 - HTTP GET

```
{
  "rel": "search",
  "href": "http://orm-hyper-todo.herokuapp.com/",
  "prompt": "Search tasks",
  "data": [
    {
      "name": "title",
      "value": "",
      "prompt": "Title"
    }
  ]
}
```

Cj Templates

- Template FORMS are for write-type domain actions
 - Add, Edit, Update, Append,
- Template FORMS are never network safe
 - HTTP POST or PUT

```
"template": {  
  "prompt": "Add task",  
  "rel": "create-form",  
  "data": [  
    {"name": "title", "value": "", "prompt": "Title"},  
    {"name": "completed", "value": "false", "prompt": "Complete"}  
  ]  
}
```

Summary

- API Design identifies DOMAIN actions
- Protocols (HTTP) identify NETWORK actions
- Media Types (HTML, Cj, etc.) map domain->network actions
- Cj uses Queries and Templates for Actions



Supporting Workflow with FORMS

Mike Amundsen
API Academy
@mamund

