REST

Daniel Jackson
browser as editor

Screenshot of Tim Berners-Lee's HyperMedia Browser/Editor removed due to copyright restrictions.
a web for shared document editing

Tim Berners Lee’s web
› distributed document editing
› like one big Wiki

HTTP methods
› GET: get a document from a server
› POST: send a document to a server
› PUT: update a document at a server
› DELETE: delete a document from a server

request components
› query string: for searching
› body: for new version of page
mosaic (1993)
what happened?

early browsers
› just displayed pages
› only needed GET

Mosaic
› added forms

to pass form data
› GET: in query string
› POST: in body
URLs in the late 1990s


script

arguments
URLs in the late 1990s

```
```

so now rather than

› DELETE pages/3

we see

› POST /deletePage
› GET /deletePage?id=3

or

› GET /foo.asp?action=delete&page=3
### 4 functions of persistent storage

<table>
<thead>
<tr>
<th><strong>CRUD</strong></th>
<th><strong>SQL</strong></th>
<th><strong>HTTP</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>create</td>
<td>insert</td>
<td>POST</td>
</tr>
<tr>
<td>read</td>
<td>select</td>
<td>GET</td>
</tr>
<tr>
<td>update</td>
<td>update</td>
<td>PUT</td>
</tr>
<tr>
<td>delete</td>
<td>delete</td>
<td>DELETE</td>
</tr>
</tbody>
</table>
RESTful calls

GET http://tea.com/teas/123/reviews/4

HTTP verb gives the action
path names a resource
kinds of path

collection paths
› http://tea.com/teas
› http://tea.com/teas/123/reviews
› http://tea.com/teas/green

instance paths
› http://tea.com/teas/123
› http://tea.com/teas/123/reviews/4
actions

show reviews of tea #123
› GET http://tea.com/teas/123/reviews

create a review of tea #123
› POST http://tea.com/teas/123/reviews

update review #4 of tea #123
› PUT http://tea.com/teas/123/reviews/4

delete review #4 of tea #123
› DELETE http://tea.com/teas/123/reviews/4
generating forms

creating a review
› step 1: display form to user
› step 2: user submits form

how this is done
› GET http://tea.com/teas/123/reviews/new
› POST http://tea.com/teas/123/reviews

and for update
› GET http://tea.com/teas/123/reviews/4/edit
› PUT http://tea.com/teas/123/reviews/4
non-CRUD actions

general strategy
› instead of calling an action, create a resource
› instead of “login”, create a “session”
› instead of “closing account”, create a “closing”

how to rate a review?
› POST http://tea.com/teas/123/reviews/4/ratings
## logging in and out

<table>
<thead>
<tr>
<th>option A</th>
<th>option B</th>
<th>option C</th>
</tr>
</thead>
<tbody>
<tr>
<td>/login</td>
<td>POST /sessions</td>
<td>POST /session</td>
</tr>
<tr>
<td>/logout</td>
<td>DELETE /sessions/12</td>
<td>DELETE /session</td>
</tr>
</tbody>
</table>

### what I’d do

- whatever your login/authentication plugin does
- more important to have session in model than to reflect in URIs