browser as editor
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
Upton Tea Imports
Vol.22 No.1  Holliston, Massachusetts  Winter, 2012

SHORTCUTS
- Home
- Catalog Index
- Order by Item Code
- Information
- Need Help?
- Just For You
- My Wish List
- Shopping Cart
- Your Account

CONTACT US
- Email: Cust. Service
- Phone: (800) 234-8327
- More contact info

SEARCH
To search this site, enter your keywords:

Search options

MORE LINKS
- Organic Teas

Browsing Green Teas

Quick-Search: current location

Teas > Green
Select the area below that you wish to browse:

Email me about new items added to this area.

Teas > Green Bestsellers
New to this Area

Go to: Ceylon, China, Formosa, India, Indonesia, Japan, Korea, Nepal

Browse Green Teas:

Green teas, the earliest form of tea, are processed without oxidation. The resulting leaves infuse to create a light and refreshing drink that should never be diluted with milk or sweetened with sugar or honey. Prized for centuries for their health benefits, green teas are an ideal choice for beverage connoisseurs.
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

- **CRUD**
  - create
  - read
  - update
  - delete

- **SQL**
  - insert
  - select
  - update
  - delete

- **HTTP**
  - POST
  - GET
  - PUT
  - DELETE
RESTful calls

HTTP verb gives the action

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

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
exercise

devise a RESTful structure for
› class registration at MIT

questions to think about?
› what are the resources?
› how should they be nested?
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
do pretty URIs matter?

one view
› URIs are just code points
› can be generated by the app
› user can ignore them

but
› users see URIs and type them
› web services try to be context-free