RESTful Rails

Practical Benefits.

Joshua Miller : joshinharrisburg.com

What is REST?

- REpresentational State Transfer
- Defined in Roy Fielding's famous dissertation

http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf

Which means what?

via Wikipedia:

- Application state and functionality are abstracted into resources
- Every resource is uniquely addressable using a universal syntax for use in hypermedia links
- All resources share a uniform interface for the transfer of state between client and resource

More practically...

- Take advantage of features built into HTTP
  - verbs like GET / POST / PUT / DELETE
  - hyperlinking
  - universally-understood and implemented protocol

- Distribute different formats to different clients
  - HTML
  - JSON
  - XML
  - etc

How do I use it?

Model applications as resources (nouns)

Shopping Cart

The bad old days...

- shopping.com/addToCart?item=glass
- shopping.com/deleteFromCart?lineItem=2
- shopping.com/checkOut

Arbitrary collections of pages with ill-defined functionality

Shopping RESTfully

Modeled as resources:

- order
- item
- line_item
- etc.

Mapped to URIs:

- shopping.com/orders/103
- shopping.com/items/12

Shopping RESTfully

Use HTTP verbs to change application state:

- add an item to an order
  - POST shopping.com/orders/103/line_item/?item=12
- remove an item
  - DELETE shopping.com/orders/103/line_item/145

Using REST in Rails

- script/generate scaffold order customer_id:integer ...
  - That's it.

- This gives you...
  - db migration
  - model
  - controller with index, edit, create, etc.
  - routes
  - scaffolded views

What do I get from this?

In Rails:

- Rails wants you to do it this way
  - Makes it easy to do this way with link and form helpers.
  - Makes it painful to it other ways.
- Convention over configuration
- Domain model visualization, separation of concerns

What do I get from this?

In Ruby (or Java, or Python...)

- Easy-to-use REST clients for distributed functionality
  - in Ruby, gem install rest-client
- Non-browser applications
- Scripting, APIs

What do I get from this?

In Javascript:

- JSON representations, easy to parse
- Unobtrusive Javascript
- Compose complex UIs in the browser with toolkits like...
  - jQuery
  - Dojo
  - Sproutcore
  - etc.

Bottom Line:

FREE = GOOD.

Questions?