ERRest: The Basics

Pascal Robert
MacTI.ca

Our agenda

- What is REST ?
- Basic ERRest concepts
- Our Model of the day
- Routes for basic operations

REST ?

- REpresentational State Transfer
- Architecture style, not a protocol or format

REST and URLs

- Goal: having consistent and logical URLs that have a meaning
  - GitHub is good example:
    - https://github.com/**projectwonder**/**wonder**
    - https://github.com/projectwonder/wonder/**admin**

REST and HTTP protocol

- Goal: to use as much as possible the HTTP protocol
  - Use HTTP verbs
  - Use headers
  - Use redirection
  - Read the spec (RFC 2616)!

REST and stateless

- Goal: to be stateless
  - Avoid storing stuff in sessions
  - Back button friendly :-)
  - More scalable (Ramsey will disagree)

REST and representations

- Goal: to deliver representations (JSON, XML, etc.) with the same URL
  - **HTML view**: https://github.com/projectwonder/wonder/commit/386d19e3080695d309e676d66db99e271c90dbab
  - **Git patch view**: https://github.com/projectwonder/wonder/commit/386d19e3080695d309e676d66db99e271c90dbab.**patch**

HTTP verbs

- Goal: to use the many HTTP verbs
  - Use GET to get representation of data
  - Use POST on a collection to create new data
  - Use PUT to update data
  - Use DELETE to delete data
- Can use the other verbs (OPTIONS, etc.) too

HATEOAS

- HATEOAS (**H**ypermedia **A**s **T**he **E**ngine of **A**pplication **S**tate) is the "full" REST
  - Goal: doing the same as a browser (start page, follow links)
  - One endpoint, and links to other resources
  - Examples: Atom Publishing, CMIS, Google Docs APIs
  - **Not currently supported by ERRest**

ERRest concepts

Key concepts

- Routes : the paths (URLs) to get/create/update/delete data
- Controllers : based on a DirectAction class
- Formats : the representations (JSON, XML, etc.)
- Filters : which attributes are in and out

Routes

- You register them in the REST request handler (/ra)
  - Handler class is: ERXRouteRequestHandler
- Can have bindings in them
  - /ra/posts/{title:String}
  - /ra/posts/{post:Post}
- Can register default (CRUD) routes in one step

Controllers

- A controller is like a DirectAction class
- A route is mapped to one method in a controller
- Base classes: ERXRouteController and ERXDefaultRouteController

Formats

- Format is the representation
- Many formats are supported (JSON, XML, plist, etc.)
- You can add your own
- Base class: ERXFormat

Filters

- Filtering is to specify what's in the object graph
- Can have different filters per route and in/out
  - Useful to hide sensible information
- You use ERXKeyFilter for filtering

CRUD

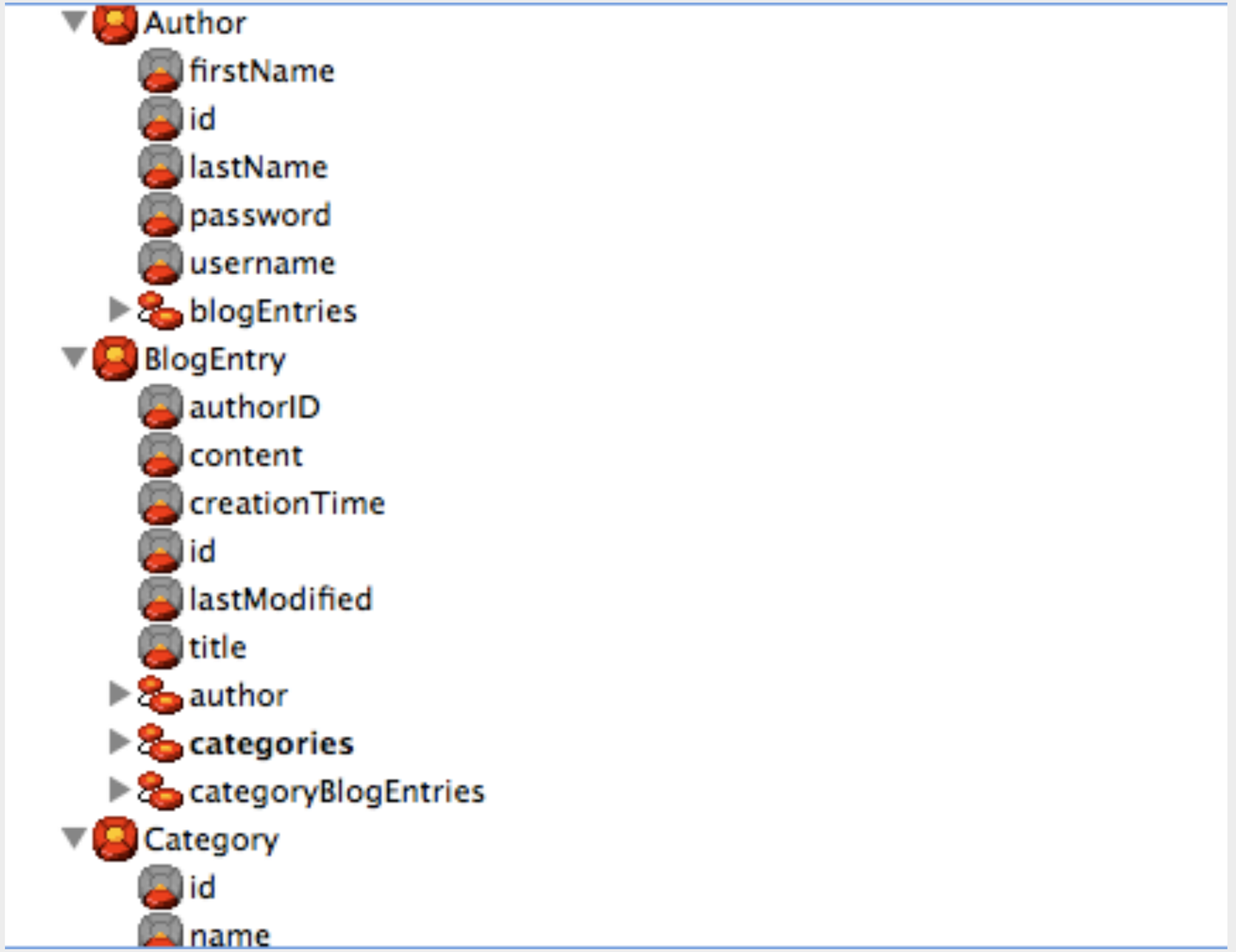
- Extends from controller from ERXDefaultRouteController
- Register your routes with restRequestHandler.addDefaultRoutes(Entity.ENTITY_NAME)

Let's create a REST service

The Model

- Example will be a blog system
- Entities:
  - BlogEntry
  - Author
  - Category

The Model


Author
firstName
id
lastName
password
username
blogEntries
BlogEntry
authorID
content
creationTime
id
lastModified
title
author
categories
categoryBlogEntries
Category
id
name

Project Setup

Adding basic routes

Basic routes

- Adding an author
- Adding a category
- Adding a blog post
- Fetching a blog post
- Updating a blog post
- Deleting a blog post

Basic routes setup and usage

Query arguments

ERXRestFetchSpecification

- ERXRestFetchSpecification gives you automatic query arguments management
- Built-in arguments:
  - sort=lastModified|desc,title|asc
  - batchSize=25&batch=1
  - qualifier=title%3D'WOWODC'
  - Range=items%3D10-19

Manual query arguments

- Same as direct actions: request().stringFormValueForKey
- A "login" action is such an example

Query arguments demo

Tomorrow

- HTML routes
- Alias, non-EO and fake primary keys
- Handling status codes
- Handling redirections
- Handling headers

Q&A