

REST API Basics

Agenda

- REST API Design
- Dropwizard Framework
- Postman and other tools

Who am I?

- Senior Application Developer,
Architecture Team at Clear Capital
- Developing web applications since 2000
(thank you dot com bust)
- ColdFusion, ASP.NET, C#, Java, Oracle

REST Defined

- “Representational State Transfer”
- An architectural style, not a specification
- Leverages HTTP specification

REST API Design



<https://apigee.com>

Architectural Points

- Nouns as entities
- HTTP verbs
- HTTP status codes
- Collections of representations
- Stateless

REST API Design

- Nouns
- Verbs
- Associations
- HTTP Status Codes
- Content Types
- Versioning
- Authentication

REST API Design

HTTP Method and URI	POST create	GET read	PUT update	DELETE delete
/plural-noun	Create/return a new noun. 201 Created or 202 Accepted Location header	Get list of nouns. 200 OK 404 Not Found	N/A 405 Method Not Allowed	Delete all nouns. 204 No Content 404 Not Found 405 Method Not Allowed
/plural-noun/id	N/A 405 Method Not Allowed	Get noun ID 200 OK 404 Not Found	Update by ID 204 No Content 404 Not Found	Delete by ID 204 No Content 404 Not Found

Domain Model

Publisher

ID int

CompanyName string

ContactName string

City string

Book

ID int

Title string

Genre string

Publisher Publisher

Author

ID int

Name string

Books Book[]

Age int

REST API Design

- Action verbs
- Multiple format support
- Searching and filtering
- Paging / field selection
- Response nice-to-haves

REST API Design

Action Verbs

- Responses that don't involve resources
- Use sparingly
- Document

`GET /convert?from=USD&to=EUR&amount=1`

REST API Design

Multiple format support

- Allow users specify the return format
- Headers, query string, “extension”
- Default to json

GET /authors.json

GET /books.xml

GET /books.html

REST API Design

Searching and Filtering

- Global search
- Scoped search

GET /search?q=dave+jones

GET /authors?q=dave+jones

REST API Design

Paging / field selection

- Let's user specify what they want
- Saves bandwidth and processing

GET /authors/123?fields=books

GET /books?limit=15&offset=30

REST API Design

Response Nice-to-Haves

- Previous / next links
- Object count

```
GET /books?limit=15&offset=30
```

```
{  
  "books"      : [{books here}],  
  "count"      : 15,  
  "next"       : "/books?limit=15&offset=45",  
  "previous"   : "/books?limit=15&offset=15"  
}
```

Resources

- ApiGee – apigee.com
- Dropwizard – www.dropwizard.io
- Postman – chrome.google.com/webstore
- Apiary – apiary.io
- Swagger – swagger.io