

# A Practical Guide to Hypermedia APIs

Eric Oestrich  
@ericoestrich  
@smartlogicsolutions





woofound.com



**Steve Klabnik**

@steveklabnik



 Follow



I just saw a diff where someone converted their  
app + client (same repo) to a hypermedia api:  
+900, -5000 LOC

 Reply  Retweet  Favorited

**13**  
RETWEETS

**4**  
FAVORITES



5:53 PM - 13 Nov 12 · Embed this Tweet

# **That was this project**

Prepare yourself

**But first**

Hypermedia Basics

**Resources  $\neq$  Models**

# **HTTP is your friend**

GET, POST, PUT, DELETE

**Client knows only one  
URL**

**"/"**



# Media types

HAL+JSON

# Sample HAL Response

```
{
  "status": "Processing",
  "_embedded": {
    "items": [
      {
        "name": "Book",
        "_links": { "self": { "href" => "http://example.com/items/1" } }
      }
    ]
  },
  "_links": {
    "self" => { "href" => "http://example.com/orders/1" }
  }
}
```

# Server

# Serializers

- Disable root
- Embedded resources
- Links
- Changing serializer representations

# Controllers

- Use `respond_with` and `respond_to`

# Models

- Doesn't have to be a subclass of ActiveRecord::Base

# Client

# Resources

- Data only
- Convenience methods



# Loaders

- Pull data down from the API
- Only GETs

# Services

- Very similar to Loaders
- Push data back up

# Resources

- Creating a Hypermedia Client
  - <http://blog.oestrich.org/2012/12/creating-a-hypermedia-client/>
- ActiveSupport::Serializers
  - [https://github.com/rails-api/active\\_model\\_serializers](https://github.com/rails-api/active_model_serializers)
- Abusing ActiveSupport::Serializers for HAL
  - <http://blog.oestrich.org/2012/10/abusing-active-model-serializers-for-hal/>