

# Spring Data REST: Data Meets Hypermedia

Unless otherwise indicated, these slides are © 2013-2014 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>

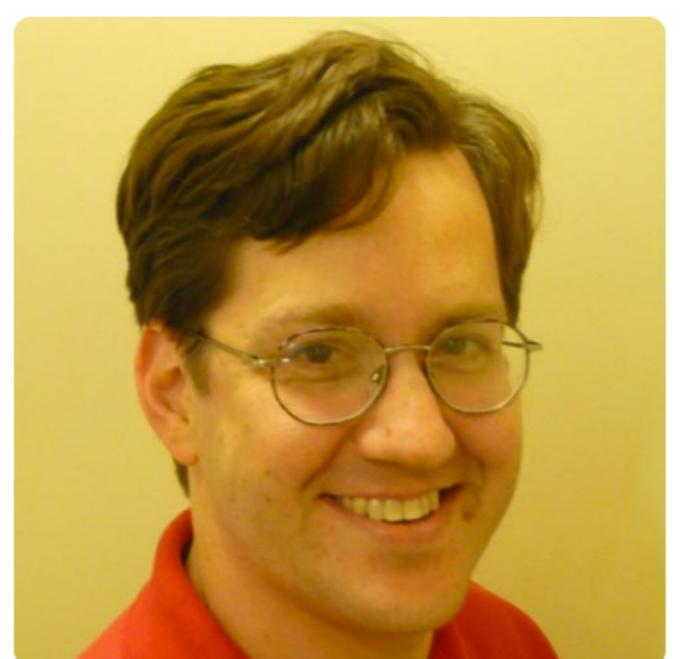
# pivotal

TM

DEVNEXUS™

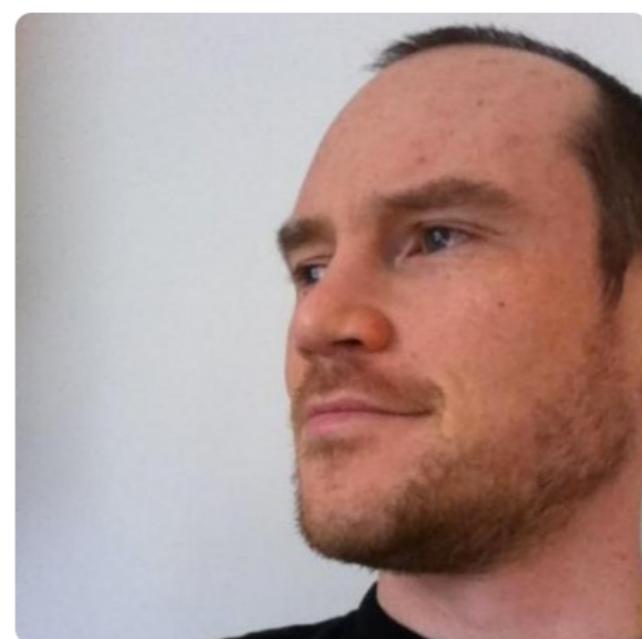
 spring  
by Pivotal

**Greg Turnquist**  
@gregturn  
gturnquist@pivotal.io  
github.com/gregturn



**Greg Turnquist**  
gregturn

**Roy Clarkson**  
@royclarkson  
rclarkson@pivotal.io  
github.com/royclarkson



**Roy Clarkson**  
royclarkson

# Recognize this?

The screenshot shows an Amazon product page for the book "Learning Spring Boot" by Greg L. Turnquist. The page includes the book cover, a "Look inside" button, a "Flip to back" button, and a "See all 2 images" link. The book is categorized under "Programming Books" and "C | Java | PHP | Python". A "CODE COMPLETE 2" badge is visible. The page also features a "Tax Central" advertisement for TurboTax.

Search

Tax Central Sponsored by TurboTax

Browse Programming Books C | Java | PHP | Python

CODE COMPLETE 2 Learn more

and over one million other books are available for **Amazon Kindle**.

**Learning Spring Boot** Paperback – November 27, 2014  
by (Author)

Read with our

Learn how to use Spring Boot to build apps faster than ever before

About This Book

- Create Spring-powered, production-grade applications and services with minimal fuss
- Support multiple environments with one artifact, and add production-grade support with features like custom metrics to track the number of messages published and consumed
- Each chapter introduces a different area that Spring Boot tackles and explains how to tweak your apps through different properties

Who This Book Is For

Also available on Kindle

Sell yours for a Gift Card  
We'll buy it for **\$21.24**

Have one to sell?

Accounting for Managers Using the Statement of ...  
Walker, Dale  
**\$3.99**

Flip to back

See all 2 images

# Is something missing?

# Recognize this?

The screenshot shows a product page for 'Learning Spring Boot' on Amazon. Numerous purple arrows point to specific UI elements:

- Top navigation bar: 'amazon Try Prime', 'Greg's Amazon.com', 'Today's Deals', 'Gift Cards', 'Sell', 'Help'.
- Search bar: 'Shop by Department', 'Search All' dropdown, 'learning spring boot' search term, 'Go' button.
- Category links: 'Books', 'Advanced Search', 'New Releases', 'Best Sellers', 'The New York Times® Best Sellers', 'Children's Books', 'Textbooks', 'Textbook Rentals', 'Sell Us Your Books', 'Best Books of the Month', 'Deals in Books'.
- Browsing section: 'Browse Programming Books' with categories 'C', 'Java', 'PHP', 'Python'.
- Book details: 'Look inside' thumbnail, book cover image, title 'Learning Spring Boot' (Paperback), author 'Greg L. Turnquist', publication date 'November 27, 2014', star rating '★★★★★', '1 customer review'.
- Purchase options: 'Buy New' button (\$40.49), 'Qty: 1' dropdown, 'List Price: \$44.99', 'Save: \$4.50 (10%)', 'FREE Shipping'.
- Stock status: 'In Stock'.
- Delivery info: 'Ships from and sold by Amazon.com', 'Gift-wrap available', 'Yes, I want FREE Two-Day Shipping with Amazon Prime' checkbox, 'Add to Cart' button.
- Sign-in options: 'Sign in to turn on 1-click ordering'.
- Shipping info: 'Want it Saturday, Feb. 28? Order within 1 hr 12 mins and choose Two-Day Shipping at checkout. Details'.
- Bottom links: 'About This Book', 'Who This Book Is For', 'Read more'.
- Footer links: 'Add to Wish List'.

## How about this?

# Is the answer...

TurboTax Tax Central Sponsored by TurboTax >

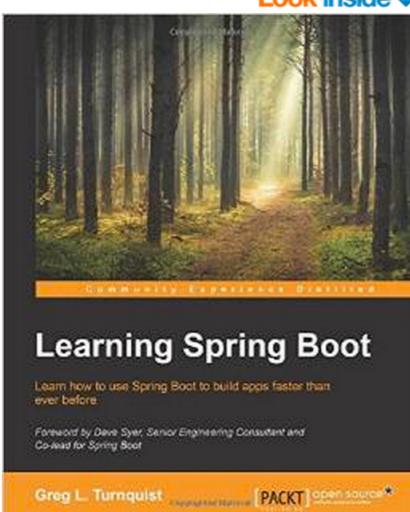
Search

Browse Programming Books C | Java | PHP | Python CODE COMPLETE 2 >Learn more

and over one million other books are available for Amazon Kindle.

**Learning Spring Boot** Paperback – November 27, 2014  
by (Author)

Look inside



Read with our

Learn how to use Spring Boot to build apps faster than ever before

About This Book

- Create Spring-powered, production-grade applications and services with minimal fuss
- Support multiple environments with one artifact, and add production-grade support with features like custom metrics to track the number of messages published and consumed
- Each chapter introduces a different area that Spring Boot tackles and explains how to tweak your apps through different properties

Who This Book Is For

Flip to back



See all 2 images

Sell yours for a Gift Card  
We'll buy it for \$21.24



Have one to sell?

Also available on Kindle



Accounting for Managers Using the Statement of ...  
Walker, Dale  
\$3.99

# ...this?

## Amazon

These are all the links for form controls from an Amazon web page

### default

Show/Hide | List Operations | Expand Operations

GET	/
POST	/gp/product/B00DBYBNEE
GET	/gp/yourstore/home
POST	/gp/goldbox
GET	/gift-cards/b
POST	/gp/seller-account/mm-product-page.html
GET	/gp/help/customer/display.html
GET	/gp/site-directory
GET	/gp/css/homepage.html
GET	/gp/registry/wishlist

**“I am getting frustrated by the number of people calling any HTTP-based interface a REST API. Today’s example is the SocialSite REST API. That is RPC. It screams RPC...”**

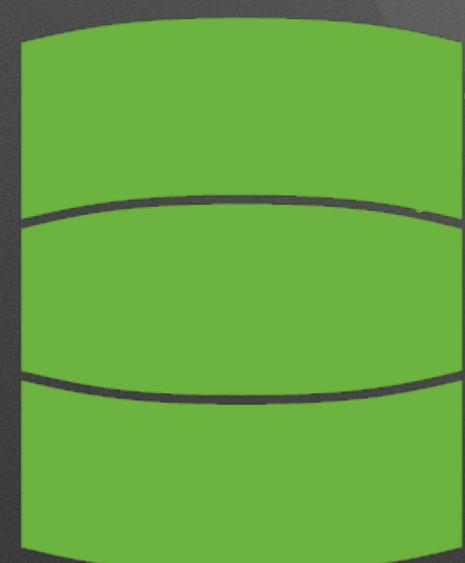
*–Roy Fielding*

*<http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>*

**“...What needs to be done to make the REST architectural style clear on the notion that hypertext is a constraint? In other words, if the engine of application state (and hence the API) is not being driven by hypertext, then it cannot be RESTful and cannot be a REST API. Period. Is there some broken manual somewhere that needs to be fixed?”**

*-Roy Fielding*

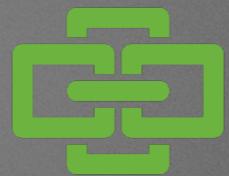
*<http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>*



# Spring Data REST DEMO



spring  
by Pivotal



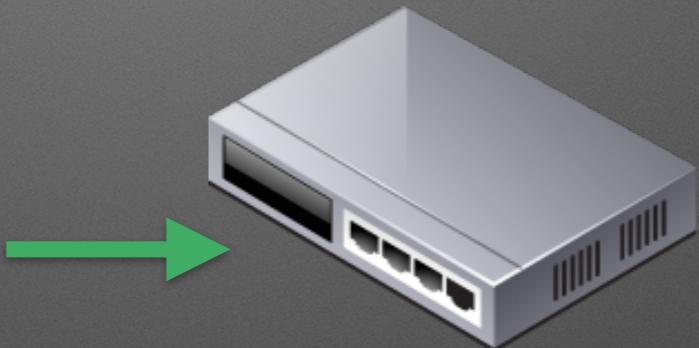
Spring HATEOAS



Spring Security

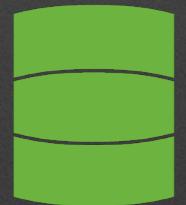


GET /api



Spring Framework

SQL



Spring Data

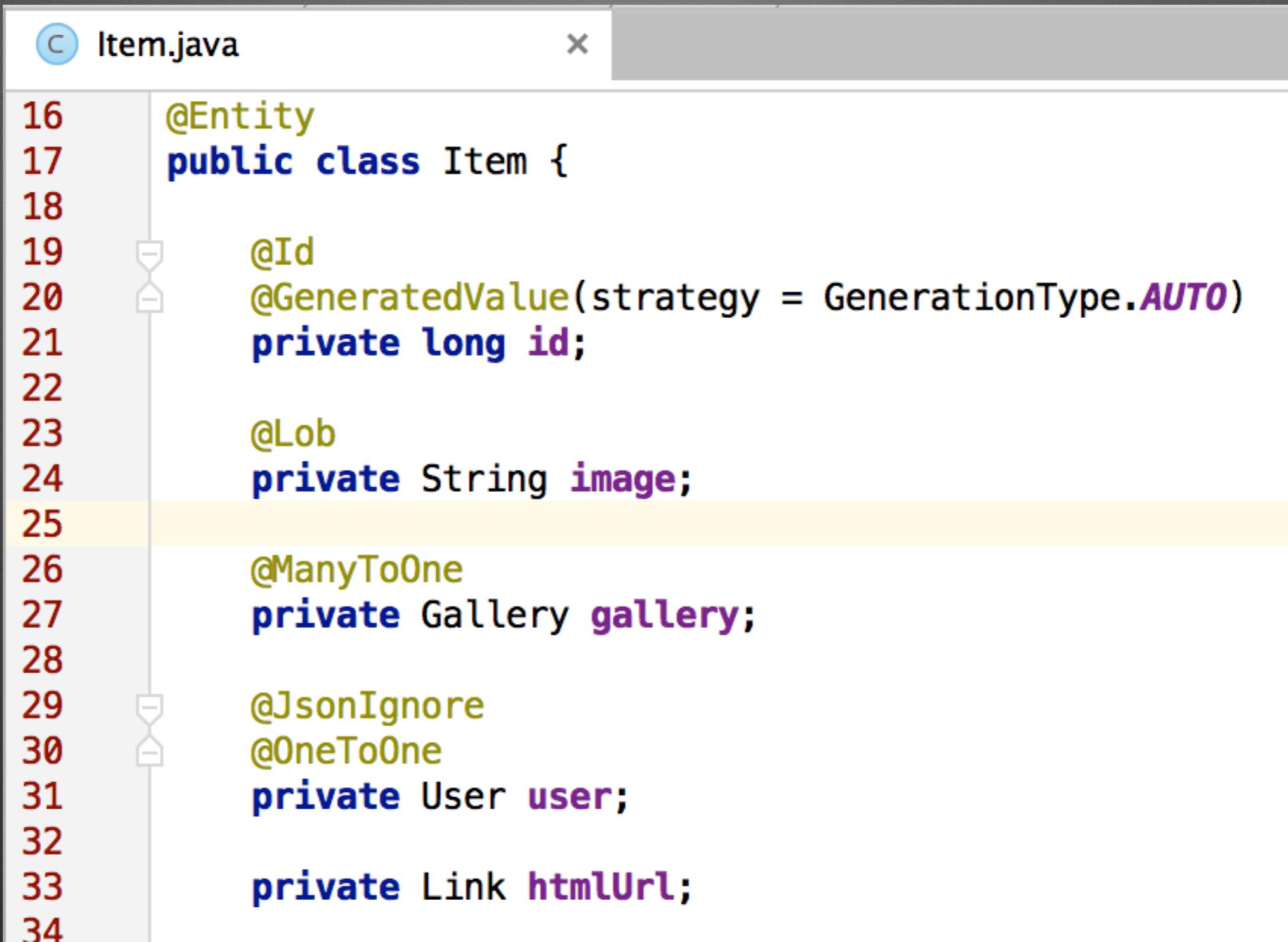
# What is Spring Data REST?

- Leverages ***HYPERMEDIA*** & Internet standards
  - HAL (draft) 
  - ALPS (draft) 
  - URI Templates (RFC 6570)
  - text/uri-list mediatype (RFC 2483)
  - profile link relation (RFC 6906)

# Item resource

```
gturnquist ~ % curl -X GET http://localhost:8080/api/items/1
{
  "image" : "... image data ...",
  "htmlUrl" : {
    "href" : "http://localhost:8080/image/1"
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/items/1{?projection}",
      "templated" : true
    },
    "gallery" : {
      "href" : "http://localhost:8080/api/items/1/gallery"
    }
  }
}retina:~ gturnquist$
```

# Item defined in Java



The screenshot shows a Java code editor window titled "Item.java". The code defines a class named "Item" with various annotations and fields. The annotations include @Entity, @Id, @GeneratedValue, @Lob, @ManyToOne, @JsonIgnore, @OneToOne, and private variables for id, image, gallery, user, and htmlUrl.

```
16  @Entity
17  public class Item {
18
19      @Id
20      @GeneratedValue(strategy = GenerationType.AUTO)
21      private long id;
22
23      @Lob
24      private String image;
25
26      @ManyToOne
27      private Gallery gallery;
28
29      @JsonIgnore
30      @OneToOne
31      private User user;
32
33      private Link htmlUrl;
34  }
```

# Gallery resource

gturnquist — bash — 68x11

```
{  
  "description" : "Collection of cats",  
  "_links" : {  
    "self" : {  
      "href" : "http://localhost:8080/api/galleries/1"  
    },  
    "items" : {  
      "href" : "http://localhost:8080/api/galleries/1/items"  
    }  
  }  
}retina:~ gturnquist$
```

# Gallery defined in Java

```
c Gallery.java x
10  @Entity
11  public class Gallery {
12
13      @Id
14      @GeneratedValue(strategy = GenerationType.AUTO)
15      private long id;
16
17      private String description;
18
19      @OneToMany(mappedBy = "gallery")
20      private List<Item> items;
21
```

# How to get

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-rest</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

**“Do not implement security on your own.”**

*-Rob Winch, [MTuSpringtSecurity](#)*

# Use HTTPS

# Authentication

# Authorization

# Attack Vectors

- XSS - Cross-Site Scripting
- CSRF - Cross Site Request Forgery
- clickjacking - User Interface redress attack
- HSTS - HTTP Strict Transport Security

# Spring Security

# Security for images

```
I ItemRepository.java X
9  @PreAuthorize("hasRole('ROLE_USER')")
10 public interface ItemRepository extends PagingAndSortingRepository<Item, Long> {
11
12     List<Item> findByGalleryIsNull();
13
14     @Override
15     @PreAuthorize("#item?.user == null or #item?.user?.name == authentication?.name")
16     Item save(@Param("item") Item item);
17
18     @Override
19     @PreAuthorize("#item?.user?.name == authentication?.name")
20     void delete(@Param("item") Item item);
21
22     @Override
23     @PreAuthorize("@itemRepository.findOne(#id)?.user?.name == authentication?.name")
24     void delete(@Param("id") Long id);
25 }
```

# Security for users

C User.java

```
11  @Entity
12  public class User {
13
14      public static final PasswordEncoder PASSWORD_ENCODER = new BCryptPasswordEncoder();
15
16      @Id @GeneratedValue
17      private Long id;
18
19      private String name;
20
21      // This field MUST be protected against any form of
22      // serialization to avoid security leakage
23      @JsonIgnore
24      private String password;
25
26      private String[] roles;
27
```

I UserRepository.java

```
6      @RepositoryRestResource(exported = false)
7      public interface UserRepository extends CrudRepository<User, Long> {
8
```

# Tailoring data with projections

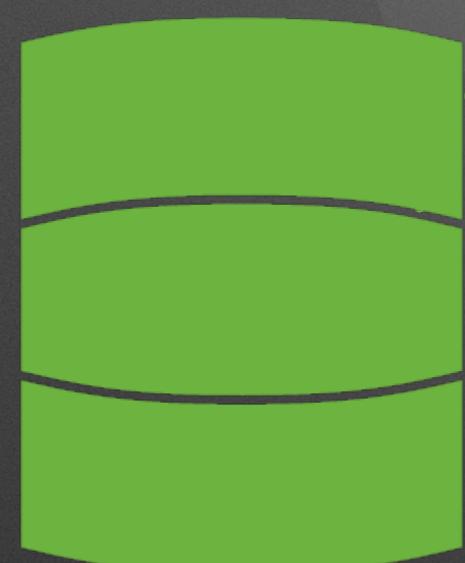
```
I Owner.java x
6 @Projection(name = "owner", types = Item.class)
7 public interface Owner {
8
9     public User getUser();
10
11    public String getImage();
12
13    public Link getHtmlUrl();
14
15 }
16
```

# Tailoring data with projections

```
retina:~ gturnquist$ curl -u greg:turnquist localhost:8080/api/items/1?projection=owner
{
  "user" : {
    "name" : "roy",
    "roles" : [ "ROLE_USER" ]
  },
  "_links" : {
    "self" : {
      "href" : "http://localhost:8080/api/items/1{?projection}",
      "templated" : true
    },
    "gallery" : {
      "href" : "http://localhost:8080/api/items/1/gallery"
    }
  }
}
retina:~ gturnquist$ █
```

# How to get

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.security.oauth</groupId>
    <artifactId>spring-security-oauth2</artifactId>
    <version>2.0.6.RELEASE</version>
</dependency>
```



# Spring Data REST DEMO



spring  
by Pivotal

# Links

- [github.com/gregturn/spring-a-gram](https://github.com/gregturn/spring-a-gram)
- <https://github.com/royclarkson/spring-rest-service-oauth>
- [twitter.com/springcentral](https://twitter.com/springcentral)
- [spring.io](https://spring.io)
  - [spring.io/guides](https://spring.io/guides)
  - [spring.io/video](https://spring.io/video)
  - [spring.io/questions](https://spring.io/questions)

# Spring Data REST: Data Meets Hypermedia

Unless otherwise indicated, these slides are © 2013-2014 Pivotal Software, Inc. and licensed under a Creative Commons Attribution-NonCommercial license: <http://creativecommons.org/licenses/by-nc/3.0/>