

Creating Hypermedia REST APIs with Apio Architect

Recipes with Apio

@alejandrohdezma

@votilde



Ask any doubt!



Slides available here:

slidr.io/ahdezma/recipes-with-apio

Preparing the environment

- Clone the workshop repo:
github.com/liferay-labs/apio-workshop
- Download Postman (a REST client)
- Download a Liferay CE Portal 7.1 GA2

What are we going to do?

Liferay Portal Headless APP

About what?



I'M HOMER,
I'LL BE YOUR CUSTOMER.

An API that publish Liferay Portal organisations as restaurants

An API that publish Liferay Portal organisations as restaurants. And also publish their recipes.

An API that publish Liferay Portal organisations as restaurants. And also publish their recipes. And follows standards.

An API that publish Liferay Portal organisations as restaurants. And also publish their recipes. And follows standards. **And works!**

Ravenwoodd



JAX-RS



Creating Hypermedia REST APIs with **Apio Architect**?



A close-up photograph of a person's hands working on a printed circuit board (PCB). The hands are using a yellow-handled screwdriver to work on a component. The PCB is densely populated with various electronic components like resistors, capacitors, and integrated circuits. The background is slightly blurred, showing more of the workspace.

**Is important to understand
foundations**

Let's start!

JAX-RS

JAX-RS: Java API for RESTful Web Services

```
@GET  
@Path("/blogs/{id}")  
@Produces(APPLICATION_JSON)  
public Response getBlogsEntry(@PathParam("id") long id) {  
    BlogsEntry blogsEntry = blogsEntryService.getBlogsEntry(id);  
  
    return Response.ok(blogsEntry).build();  
}
```

What do we need to create a JAX-RS API?

- Application
- Resources
- Extensions

Application

- Mandatory
- Manages the API resources

```
@Component  
@ApplicationPath("api")  
public class MyApplication extends Application {  
  
    @Override  
    public Set<Class<?>> getClasses() {  
        return Collections.singleton(new Resource());  
    }  
  
}
```

Resources

- Optional (if you don't want to modularize)
- Manages endpoints in a specific path

```
@Path("/blogs")
public class Resource {

    @GET
    @Path("/{id}")
    @Produces(APPLICATION_JSON)
    public Response getBlogsEntry(@PathParam("id") long id) {
        BlogsEntry blogsEntry = blogsEntryService.getBlogsEntry(id);

        return Response.ok(blogsEntry).build();
    }

}
```

Extensions

- Optional
- Adds additional logic to the API

```
@PreMatching
public class AuthFilter implements ContainerRequestFilter {

    public void filter(ContainerRequestContext ctx) {
        String authHeader = request.getHeaderString(HttpHeaders.AUTHORIZATION);

        if (authHeader == null) {
            throw new NotAuthorizedException("Not authenticated");
        }

        if (!verifyUser(authHeader)) {
            throw new NotAuthorizedException("User not authorized");
        }
    }

    private boolean verifyUser(String authHeader) {}

}
```

Extensions

- Optional
- Adds additional logic to the API

```
@Component
public class UserProvider implements ContextProvider<User> {

    @Override
    public User createContext(Message message) {
        try {
            return _userService.getCurrentUser();
        } catch (PortalException e) {
            throw new NotAuthorizedException(e, "Basic");
        }
    }

    @Reference
    private UserService _userService;

}
```

How do we use JAX-RS in Liferay Portal?

JAX-RS Whiteboard

- Aries JAX-RS Whiteboard is the reference implementation of the [OSGi JAX-RS Services Whiteboard 1.0](#) (JAX-RS for OSGI)
- Component-property based

```
@Component(service = Resource.class, property = JAX_RS_RESOURCE + "=true")
@Path("/blogs")
public class Resource {

    @GET
    @Path("/{id}")
    @Produces(APPLICATION_JSON)
    public Response getBlogsEntry(@PathParam("id") long id) {
        BlogsEntry blogsEntry = blogsEntryService.getBlogsEntry(id);

        return Response.ok(blogsEntry).build();
    }

}
```

Thanks @csierra & @rotty3000!

Completing the environment

- Open the project **workshop** (included in the repository)
- Set your Liferay Home
- Run `./gradlew prepare`
- Start Liferay Portal: `path/to/liferay_home/tomcat-9.0.10/bin/catalina.sh jpda run`
- Wait for starting...
- Run `./gradlew deploy`

Let's test the APIs!

- Click on the [▶ Run in Postman](#) button
- Follow the instructions of our handsome speaker

What do we have?

Restaurants



Chefs



Kitchen Assistants



Recipes



Our tools



GoGo Shell

telnet localhost 11311

```
vagrant@vagrant:~$ telnet localhost 11311
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.  
-----
```

```
Welcome to Apache Felix Gogo
```

```
g! help
blade:usercount
dependencymanager:dm
equinox:b
equinox:bundle
equinox:bundles
equinox:classSpaces
equinox:close
equinox:diag
equinox:disconnect
```

GoGo Shell

— apio:restaurants

ID	Name	Chef Email
35301	Diverxo	the.chef@diverxo.com

— apio:assistants

Kitchen Assistant Email	Organizations
the.assistant@diverxo.com	Diverxo

Postman

EntryPoint

New Import Runner + My Workspace Invite

No Environment Examples (0)

Filter History Trash

Api Architect Workshop JAX-RS 15 requests

JAX-RS

Restaurants

GET Retrieve all restaurants

GET Retrieve first restaurant

GET Not Found Restaurant

Chef

Recipes

Api Architect

Restaurants

GET Retrieve all restaurants

GET Retrieve first restaurant

GET Not Found Restaurant

Chef

Recipes

EntryPoint

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code

KEY VALUE DESCRIPTION

Key Value Description

Body Cookies (1) Headers (11) Test Results Status: 200 OK Time: 132 ms Size: 527 B Save Download

Pretty Raw Preview JSON

```
1 {  
2   "organization": "http://localhost:8080/o/api/organization"  
3 }
```

Build Browse

Filter History Trash

Api Architect Workshop JAX-RS 15 requests

JAX-RS

Restaurants

GET Retrieve all restaurants

GET Retrieve first restaurant

GET Not Found Restaurant

Chef

Recipes

Api Architect

Restaurants

GET Retrieve all restaurants

GET Retrieve first restaurant

GET Not Found Restaurant

Chef

Recipes

EntryPoint

Our IDE

- Eclipse
- Liferay Dev Studio
- IntelliJ
- Visual Studio Code
- ...

Demo time



Interactive demo time



Apio Architect

What is Apio Architect?

- Server-side library that facilitates the creation of Hypermedia REST APIs
- OSGi
- JAX-RS as foundation

APIO Architect Goals

- Evolvable
- Reduce boilerplate
- Auto-documentation
- Fun!

Hypermedia

HATEOAS?

How many of you have used Hypermedia today?

All of you

Test the API

▶ Run in Postman

In essence hypermedia is just another name for everything that we see, hear, and interact with on the Web.

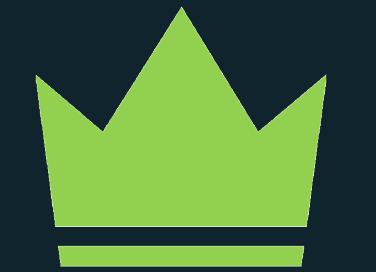
This is not a new concept

REST style is an abstraction of the architectural elements within a distributed **hypermedia** system

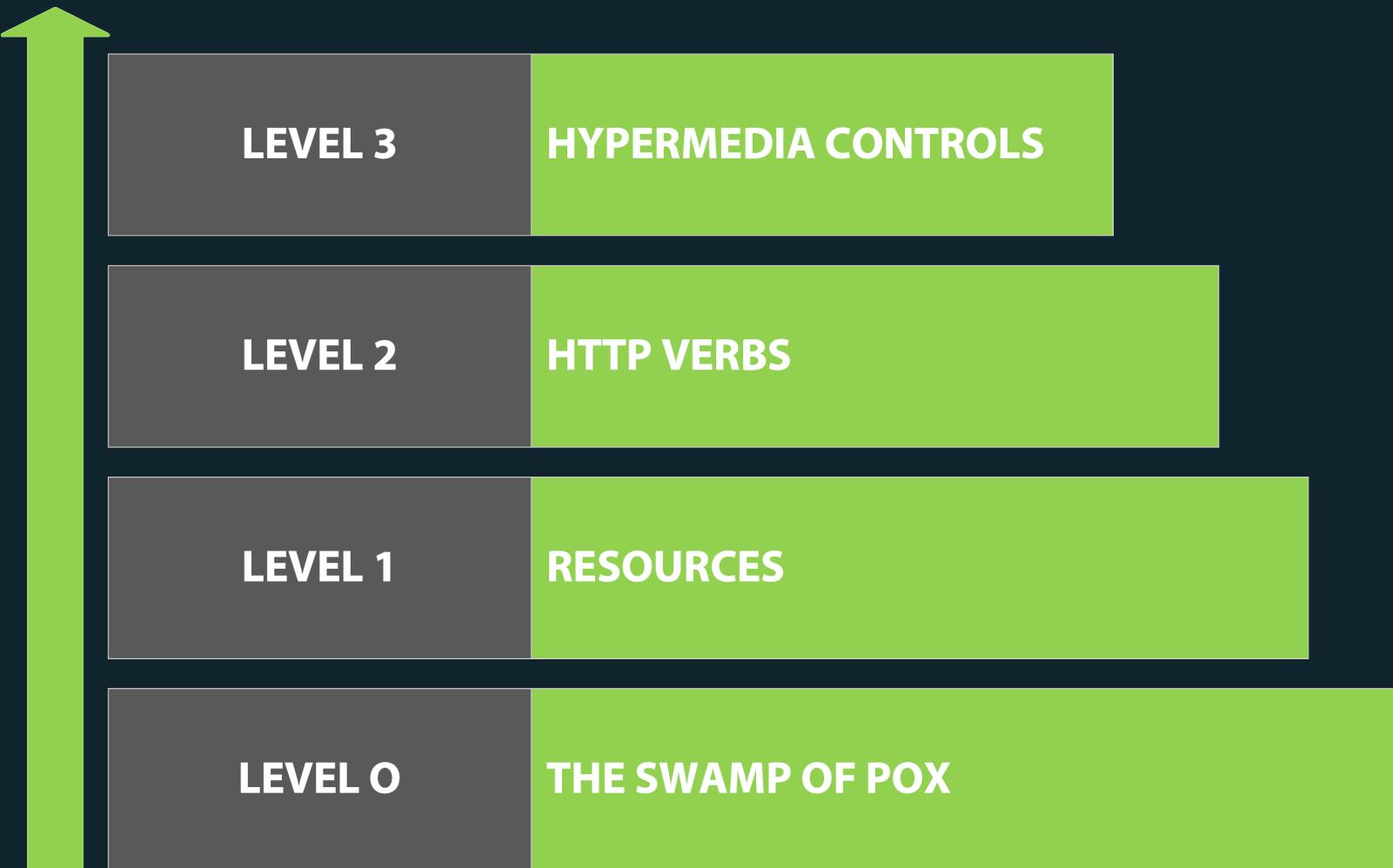
— Roy Fielding, 2000

REST "well" done
(Richardson Maturity Model)

— Martin Fowler's blog



The glory of REST



How are we solving this?



Home URL

**Consumers must only know ONE URL
And how to navigate from it**

```
{  
  "@context": "https://apiosample.wedeploy.io/doc",  
  "@id": "https://apiosample.wedeploy.io",  
  "@type": "EntryPoint",  
  "collection": [  
    {  
      "@id": "https://apiosample.wedeploy.io/blog-postings",  
      "@type": "Collection",  
      "manages": {  
        "object": "schema:BlogPosting",  
        "property": "rdf:type"  
      }  
    },  
    {  
      "@id": "https://apiosample.wedeploy.io/people",  
      "@type": "Collection",  
      "manages": {  
        "object": "schema:Person",  
        "property": "rdf:type"  
      }  
    }  
  ]  
}
```

Entry point with all the "root" APIs (JSON-LD)



Affordance types

**Contract with consumer defines affordance types
(relations, actions, ...)**

Start with IANA's 80 relation types

```
{  
  "total": 43,  
  "count": 30,  
  "_links": {  
    "collection": {  
      "href": "http://apiosample.wedeploy.io/p/people"  
    },  
    "first": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"  
    },  
    "last": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=2&per_page=30"  
    },  
    "next": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=2&per_page=30"  
    },  
    "self": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"  
    }  
  },  
}
```

Pagination (HAL)

```
{  
  "total": 43,  
  "count": 30,  
  "_links": {  
    "collection": {  
      "href": "http://apiosample.wedeploy.io/p/people"  
    },  
    "first": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"  
    },  
    "last": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"  
    },  
    "self": {  
      "href": "http://apiosample.wedeploy.io/p/people?page=1&per_page=30"  
    }  
  },  
}
```

Standard link types (IANA Link Relations)

```
{  
  "@id": "http://apiosample.wedeploy.io/p/people",  
  "@type": [  
    "Collection"  
,  
  "members": [  
    ...  
,  
  "numberOfItems": 10,  
  "operation": [  
    {  
      "@id": "people/create",  
      "@type": "Operation",  
      "expects": "http://apiosample.wedeploy.io/f/c/people",  
      "method": "POST"  
    }  
  ]  
}
```

Actions (Hydra + JSON-LD)

```
{  
  "@id": "http://apiosample.wedeploy.io/p/people",  
  "@type": [  
    "Collection"  
,  
  "members": [  
    ...  
,  
  "numberOfItems": 10,  
  "operation": [  
    {  
      "@id": "people/create",  
      "@type": "Operation",  
      "expects": "http://apiosample.wedeploy.io/f/c/people",  
      "method": "POST"  
    }  
  ]  
}
```

Actions (Hydra + JSON-LD)

```
{
  "title": "Apio Sample API",
  "description": "This API allows developers try a Hypermedia API without creating one",
  "entrypoint": "https://apiosample.wedeploy.io",
  "supportedClass": [
    {
      "@id": "Comment",
      "@type": "Class",
      "title": "Comment",
      "supportedOperation": [
        {
          "@id": "_:comments/retrieve",
          "@type": [
            "Operation"
          ],
          "method": "GET",
          "returns": "Comment",
          "comment": "Return the list of comments."
        },
        ...
      ],
      "supportedProperty": [
        {
          "@type": "SupportedProperty",
          "property": "dateCreated",
          "comment": "The date on which the Comment was created or the item was added to a DataFeed."
        },
        ...
      ]
    }
  ],
  "@id": "/doc",
  "@type": "ApiDocumentation"
}
```

Profile (Hydra + JSON-LD)

```
{  
  "headline": "A blog",  
  "alternativeHeadline": "A subtitle",  
  "_links": {  
    "creator": {  
      "href": "http://apiosample.wedeploy.io/p/people/1"  
    },  
    "self": {  
      "href": "http://apiosample.wedeploy.io/p/blog-postings/12"  
    }  
  },  
}
```

Links to other resources (HAL)

How is APIO Architect enabling this?

Annotations

Vocabulary annotations

- @Vocabulary.Type
- @Vocabulary.Field
- @Vocabulary.LinkedModel
- @Vocabulary.RelativeURL
- @Vocabulary.RelatedCollection
- @Id

```
@Type("BlogPosting")
public interface BlogPosting extends Identifier<Long> {

    @Id
    public Long getId();

    @Field("alternativeHeadline")
    public String getAlternativeHeadline();

    @Field("articleBody")
    public String getArticleBody();

    @Field("creator")
    @LinkedModel(Person.class)
    public Long getCreatorId();

    @Field("headline")
    public String getHeadline();

    @Field("review")
    public List<Review> getReviews();

}
```

Exercise 1: Returning Apio Types

- Split our POJOs into interface and implementation
- Annotate interface with Apio Vocabulary annotations

Commit on Nov 22, 2018

final branch

Step 7 - Use standard types

 ahdezma committed 11 minutes ago



7ae3ec0



Step 6 - Transform POST operation into Apio Architect actions

 ahdezma committed 15 hours ago



87ab8cc



Step 5 - Add navigation to recipes

 ahdezma committed 15 hours ago



a5af4d1



Step 4 - Update to Apio Architect's ExceptionMapper

 ahdezma committed 16 hours ago



e285b1d



Step 3 - Pagination

 ahdezma committed 22 minutes ago



74a96fb



Step 2 - Transform GET operations into Apio Architect actions

 ahdezma committed 16 hours ago



173998c



Step 1 - Create Apio Architect Type Interfaces

 ahdezma committed 17 hours ago



65ba95a



Initial commit

 victorg1991 authored and ahdezma committed 14 days ago



f41dfbf



Action annotations

- @Actions.Action
- @Actions.Retrieve
- @Actions.Create
- @Actions.Remove
- @Actions.Replace
- @Actions.EntryPoint

Abstract away REST

```
@Component
public class BlogPostingActionRouter implements ActionRouter<BlogPosting> {

    @Retrieve
    public BlogPosting retrieve(@Id long id);

    @EntryPoint
    @Retrieve
    public PageItems<BlogPosting> retrievePage(Pagination pagination);

}
```

Exercise 2: Transforming JAX-RS GET endpoints

- Use Actions annotations to transform GET endpoints into Apio Architect actions.

Let's try the API

Exercise 3: Pagination

- Return PageItems instead of List
- Add Pagination param

Which format is better?

```
{  
  "gender": "female",  
  "familyName": "Hart",  
  "givenName": "Sophia",  
  "jobTitle": "Junior Executive",  
  "name": "Sophia Hart",  
  "birthDate": "1965-04-12T00:00Z",  
  "email": "sophia.hart@example.com",  
  "_links": {  
    "self": {  
      "href": "http://localhost:8080/o/api/p/people/30723"  
    }  
  }  
}
```

HAL

```
{  
  "class": "BlogPosting",  
  "properties": {  
    "headline": "Hello DEVCON!",  
    "content": "The content of this blog posting"  
  },  
  "actions": [  
    {  
      "name": "delete-blog-posting",  
      "title": "Delete Blog Posting",  
      "method": "DELETE",  
      "href": "http://localhost:8080/o/p/blogs/32400"  
    }  
  ],  
}
```

SIREN

```
{  
  "gender": "female",  
  "familyName": "Hart",  
  "givenName": "Sophia",  
  "jobTitle": "Junior Executive",  
  "name": "Sophia Hart",  
  "birthDate": "1965-04-12T00:00Z",  
  "email": "sophia.hart@example.com",  
  "@id": "http://localhost:8080/o/api/p/people/30723",  
  "@type": "Person",  
  "@context": "http://schema.org"  
}
```

JSON-LD

Which format is better?

All of them

All of them

or none

It depends on your case

Whats the solution then?

Apio Architect

Try changing the Accept header to:

- application/json
- application/hal+json
- application/ld+json

Exception handling in Apio Architect

Exercise 4: Transforming JAX-RS ExceptionMapper into Apio Architect's

With the **Vocabulary annotations** internal models are **transformed** to types and stored as a **generic representation** that can be then **mapped** to the representation format chosen by the user

What are the advantages of this approach?

- Enforce the mapping

- Enforce the mapping
- Abstract from URLs

- Enforce the mapping
- Abstract from URLs
- Generic representation

- Enforce the mapping
- Abstract from URLs
- Generic representation
- Link relations

- Enforce the mapping
- Abstract from URLs
- Generic representation
- Link relations
- Navigability

Navigability?

Exercise 5: Add recipes field to Organization

Exercise 6: Transform non-GET endpoints

- @ParentId
- @Body

How do we design APIs that are capable of evolving?

Evolvable APIs

Hypermedia

Shared
Vocabularies

REST Foundation



Standard types

schema.org: 597 types y 867 properties

schema.org

BlogPosting

Canonical URL: <http://schema.org/BlogPosting>

Thing > CreativeWork > Article > SocialMediaPosting > BlogPosting

A blog post.

Usage: Over 1,000,000 domains

[more...]

Property	Expected Type	Description
Properties from SocialMediaPosting		
<code>sharedContent</code>	CreativeWork	A CreativeWork such as an image, video, or audio clip shared as part of this posting.
Properties from Article		
<code>articleBody</code>	Text	The actual body of the article.
<code>articleSection</code>	Text	Articles may belong to one or more 'sections' in a magazine or newspaper, such as Sports, Lifestyle, etc.
<code>pageEnd</code>	Integer or Text	The page on which the work ends; for example "138" or "xvi".
<code>pageStart</code>	Integer or Text	The page on which the work starts; for example "135" or "xiii".
<code>pagination</code>	Text	Any description of pages that is not separated into pageStart and pageEnd; for example, "1-6, 9, 55" or "10-12, 46-49".
	SpeakableSpecification or URL	Indicates sections of a Web page that are particularly 'speakable' in the sense of being highlighted as being especially appropriate for text-to-speech conversion. Other sections of a page may also be usefully spoken in particular circumstances; the 'speakable' property serves to indicate the parts most likely to be generally useful for speech. The <i>speakable</i> property can be repeated an arbitrary number of times, with three kinds of possible 'content-locator' values: 1.) <i>id</i> or <i>URI</i> of the element containing the content to be spoken. This is best

Liferay DXP Software Reviews | Gartner Peer Insights

<https://www.gartner.com/reviews/market/.../liferay/.../liferay-dxp> ▾ Traducir esta página

 Valoración: 4,2 - 13 reseñas

Choose business IT software and services with confidence. Read verified Liferay DXP Digital Experience Platforms (formerly Horizontal Portal Software) **Reviews** from the IT community.

Web Content Review Date? - Foros de la Comunidad | Liferay

https://web.liferay.com/es/community/forums/-/message_boards/.../15661982 ▾

16 ago. 2012 - I'm talking about the default Liferay **review** date functionality for web content, under the "Schedule" section, not review notifications for Kaleo Workflow. I've unfortunately been able to find very little documentation on this. I've set several relevant portal properties, and my Liferay instance is successfully able ...

```
<div class="reviewSnippetCard" itemtype="http://schema.org/Review">
  <span itemprop="publisher" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="Gartner, Inc.">
    <meta itemprop="url" content="https://www.gartner.com">
  </span>
  <span itemprop="provider" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="liferay">
    <meta itemprop="url" content="/reviews/market/horizontal-portals/vendor/liferay">
  </span>
  <span class="product-names" itemprop="itemReviewed" itemtype="http://schema.org/Product">
    <h3 itemprop="name">Liferay DXP</h3>
  <span>
    <div itemprop="reviewRating" itemtype="http://schema.org/Rating">
      <meta itemprop="worstRating" content="1">
      <meta itemprop="bestRating" content="5">
      <meta itemprop="ratingValue" content="4">
    </div>
  </span>
</div>
```

```
<div class="reviewSnippetCard" itemtype="http://schema.org/Review">
  <span itemprop="publisher" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="Gartner, Inc.">
    <meta itemprop="url" content="https://www.gartner.com">
  </span>
  <span itemprop="provider" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="liferay">
    <meta itemprop="url" content="/reviews/market/horizontal-portals/vendor/liferay">
  </span>
  <span class="product-names" itemprop="itemReviewed" itemtype="http://schema.org/Product">
    <h3 itemprop="name">Liferay DXP</h3>
  <span>
    <div itemprop="reviewRating" itemtype="http://schema.org/Rating">
      <meta itemprop="worstRating" content="1">
      <meta itemprop="bestRating" content="5">
      <meta itemprop="ratingValue" content="4">
    </div>
  </span>
</div>
```

```
<div class="reviewSnippetCard" itemtype="http://schema.org/Review">
  <span itemprop="publisher" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="Gartner, Inc.">
    <meta itemprop="url" content="https://www.gartner.com">
  </span>
  <span itemprop="provider" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="liferay">
    <meta itemprop="url" content="/reviews/market/horizontal-portals/vendor/liferay">
  </span>
  <span class="product-names" itemprop="itemReviewed" itemtype="http://schema.org/Product">
    <h3 itemprop="name">Liferay DXP</h3>
  <span>
    <div itemprop="reviewRating" itemtype="http://schema.org/Rating">
      <meta itemprop="worstRating" content="1">
      <meta itemprop="bestRating" content="5">
      <meta itemprop="ratingValue" content="4">
    </div>
  </span>
</div>
```

```
<div class="reviewSnippetCard" itemtype="http://schema.org/Review">
  <span itemprop="publisher" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="Gartner, Inc.">
    <meta itemprop="url" content="https://www.gartner.com">
  </span>
  <span itemprop="provider" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="liferay">
    <meta itemprop="url" content="/reviews/market/horizontal-portals/vendor/liferay">
  </span>
  <span class="product-names" itemprop="itemReviewed" itemtype="http://schema.org/Product">
    <h3 itemprop="name">Liferay DXP</h3>
  <span>
    <div itemprop="reviewRating" itemtype="http://schema.org/Rating">
      <meta itemprop="worstRating" content="1">
      <meta itemprop="bestRating" content="5">
      <meta itemprop="ratingValue" content="4">
    </div>
  </span>
</div>
```

```
<div class="reviewSnippetCard" itemtype="http://schema.org/Review">
  <span itemprop="publisher" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="Gartner, Inc.">
    <meta itemprop="url" content="https://www.gartner.com">
  </span>
  <span itemprop="provider" itemtype="http://schema.org/Organization">
    <meta itemprop="name" content="liferay">
    <meta itemprop="url" content="/reviews/market/horizontal-portals/vendor/liferay">
  </span>
  <span class="product-names" itemprop="itemReviewed" itemtype="http://schema.org/Product">
    <h3 itemprop="name">Liferay DXP</h3>
  <span>
    <div itemprop="reviewRating" itemtype="http://schema.org/Rating">
      <meta itemprop="worstRating" content="1">
      <meta itemprop="bestRating" content="5">
      <meta itemprop="ratingValue" content="4">
    </div>
  </span>
</div>
```

How do we port this to APIs?

```
{  
  "blogsEntryId": "24557",  
  "subtitle": "Vero ex excepturi exercitationem.",  
  "content": "Exercitationem exercitationem temporibus eum quasi aliquid.",  
  "userId": "23422",  
  "create_date": "2017-12-30T17:31Z",  
  "modified_date": "2017-30-12",  
  "title": "Recalled to Life"  
}
```

Our good old JSON-WS

```
{  
  "blogsEntryId": "24557",  
  "subtitle": "Vero ex excepturi exercitationem.",  
  "content": "Exercitationem exercitationem temporibus eum quasi aliquid.",  
  "userId": "23422",  
  "create_date": "2017-12-30T17:31Z",  
  "modified_date": "2017-30-12",  
  "title": "Recalled to Life"  
}
```

Our coupled old JSON-WS

```
{  
  "blogsEntryId": "24557",  
  "subtitle": "Vero ex excepturi exercitationem.",  
  "content": "Exercitationem exercitationem temporibus eum quasi aliquid.",  
  "userId": "23422",  
  "create_date": "2017-12-30T17:31Z",  
  "modified_date": "2017-30-12",  
  "title": "Recalled to Life"  
}
```

Our too coupled old JSON-WS

```
{  
  "@context": "http://apiosample.wedeploy.io/doc",  
  "@id": "https://apiosample.wedeploy.io/p/blog-postings/24557",  
  "@type": "BlogPosting",  
  "alternativeHeadline": "Vero ex excepturi exercitationem.",  
  "articleBody": "Exercitationem exercitationem temporibus eum quasi aliquid.",  
  "creator": {  
    "@id": "https://apiosample.wedeploy.io/p/people/23422",  
    "@type": "Person"  
  },  
  "dateCreated": "2017-12-30T17:31Z",  
  "dateModified": "2017-12-30T17:31Z",  
  "headline": "Recalled to Life"  
}
```

Shared vocabularies

Mapping internal BlogsEntry with [schema.org](#) BlogPosting

Defining **types** and their **mapping** to **internal models** and actions is the most important API design activity

Let's use standard types!

Exercise 7: Use standard types

- Look up schema.org **Recipe**, **Restaurant**,
PostalAddress and **Person**
- Update @Vocabulary.Field & @Vocabulary.Type
annotations accordingly

Altering the data

- Go to: path/to/liferay/home/osgi/configs/com.liferay.recipes.demo.internal.RecipesDemo-default.cfg
- Tweak the numbers to generate more data

Apio Architect API features

- embedded
- fields

The future

Java Bean Validations

```
@Type("BlogPosting")
public interface BlogPosting extends Identifier<Long> {

    @Field("alternativeHeadline")
    public String getAlternativeHeadline();

    @Field("articleBody")
    @Email(message = "Email should be valid")
    public String getArticleBody();

    @Field("headline")
    @NotNull
    public String getHeadline();

    @Field("review")
    @NotEmpty
    public List<Review> getReviews();

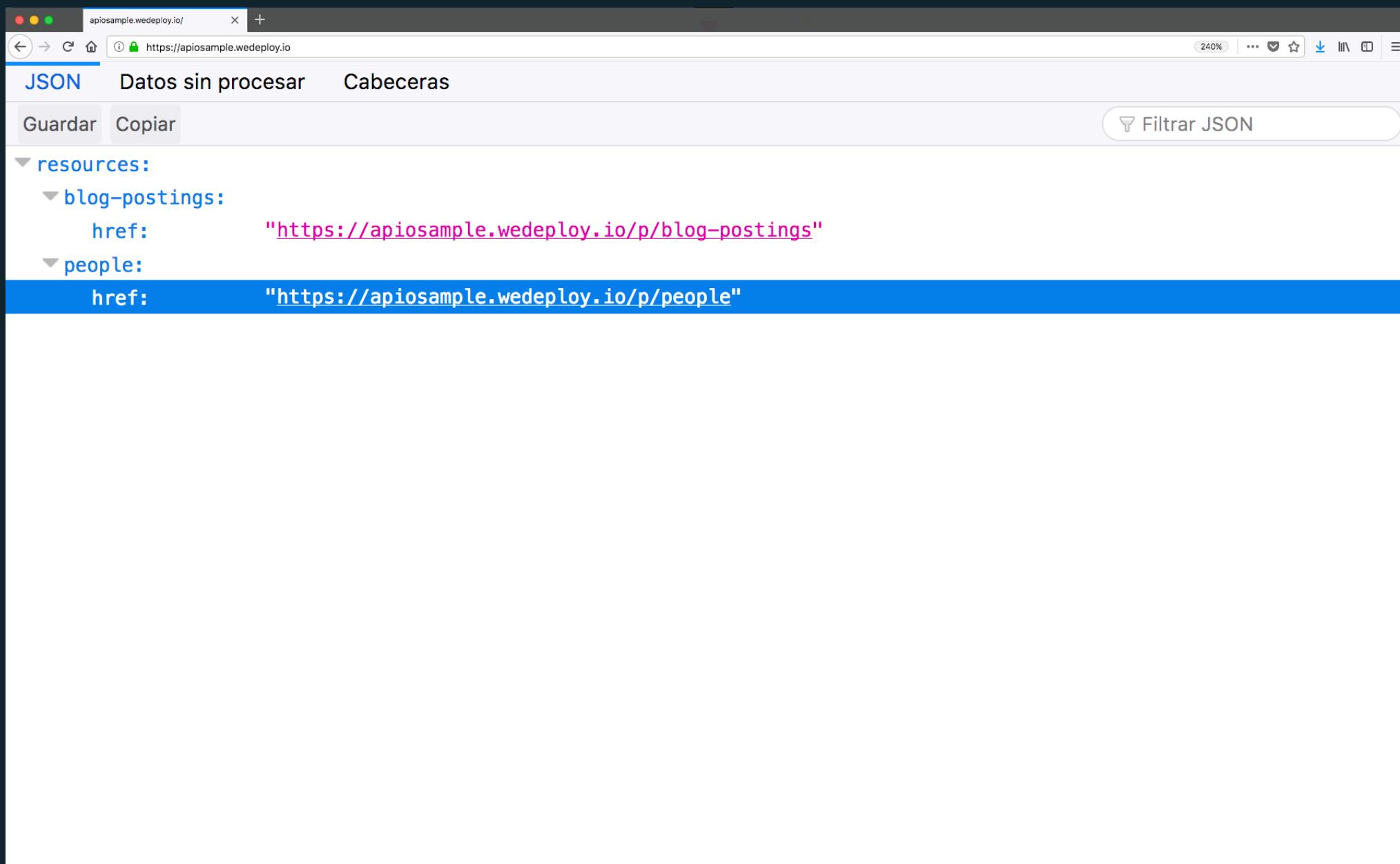
}
```

OpenAPI

Automatically generated by Apio Architect

Documentation

apiosample.wedeploy.io



thank you!

Recipes with Apio

@alejandrohdezma

@votilde

