

Spring Data REST

Configuration, Pagination and Sorting



REST Endpoints

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
 - First character of Entity type is lowercase

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
 - First character of Entity type is lowercase
 - Then just adds an "s" to the entity

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
 - First character of Entity type is lowercase
 - Then just adds an "s" to the entity

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {
```

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
 - First character of Entity type is lowercase
 - Then just adds an "s" to the entity

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}
```

REST Endpoints

- By default, Spring Data REST will create endpoints based on entity type
- Simple pluralized form
 - First character of Entity type is lowercase
 - Then just adds an "s" to the entity

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}
```



/employees

Pluralized Form

Pluralized Form

- Spring Data REST pluralized form is VERY simple

Pluralized Form

- Spring Data REST pluralized form is VERY simple
 - Just adds an "s" to the entity

Pluralized Form

- Spring Data REST pluralized form is VERY simple
 - Just adds an "s" to the entity
- The English language is VERY complex!

Pluralized Form

- Spring Data REST pluralized form is VERY simple
 - Just adds an "s" to the entity
- The English language is VERY complex!
 - Spring Data REST does NOT handle

Pluralized Form

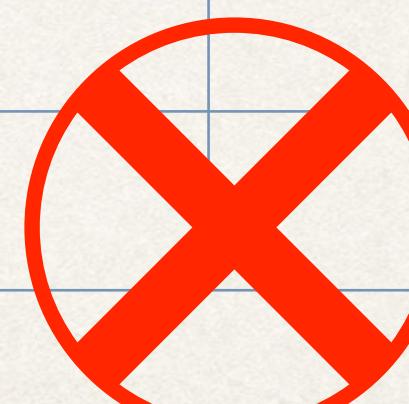
- Spring Data REST pluralized form is VERY simple
 - Just adds an "s" to the entity
- The English language is VERY complex!
 - Spring Data REST does NOT handle

Singular	Plural
Goose	Geese
Person	People
Syllabus	Syllabi
...	...

Pluralized Form

- Spring Data REST pluralized form is VERY simple
 - Just adds an "s" to the entity
- The English language is VERY complex!
 - Spring Data REST does NOT handle

Singular	Plural
Goose	Geese
Person	People
Syllabus	Syllabi
...	...



Problem

Problem

- Spring Data REST does not handle complex pluralized forms

Problem

- Spring Data REST does not handle complex pluralized forms
 - In this case, you need to specify plural name

Problem

- Spring Data REST does not handle complex pluralized forms
 - In this case, you need to specify plural name
- What if we want to expose a different resource name?

Problem

- Spring Data REST does not handle complex pluralized forms
 - In this case, you need to specify plural name
- What if we want to expose a different resource name?
 - Instead of **/employees** ... use **/members**

Solution

Solution

- Specify plural name / path with an annotation

Solution

- Specify plural name / path with an annotation

```
@RepositoryRestResource(path="members")
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {
}
```

Solution

- Specify plural name / path with an annotation

```
@RepositoryRestResource(path="members")
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {
}
```

<http://localhost:8080/members>

Pagination

Pagination

- By default, Spring Data REST will return the first 20 elements
 - Page size = 20

Pagination

- By default, Spring Data REST will return the first 20 elements
 - Page size = 20
- You can navigate to the different pages of data using query param

Pagination

- By default, Spring Data REST will return the first 20 elements
 - Page size = 20
- You can navigate to the different pages of data using query param

```
http://localhost:8080/employees?page=0
```

```
http://localhost:8080/employees?page=1
```

...

Pagination

- By default, Spring Data REST will return the first 20 elements
 - Page size = 20
- You can navigate to the different pages of data using query param

```
http://localhost:8080/employees?page=0
```

```
http://localhost:8080/employees?page=1
```

...

Pages are
zero-based

Spring Data REST Configuration

- Following properties available: application.properties

Name	Description
<code>spring.data.rest.base-path</code>	Base path used to expose repository resources
<code>spring.data.rest.default-page-size</code>	Default size of pages
<code>spring.data.rest.max-page-size</code>	Maximum size of pages
...	...

Spring Data REST Configuration

- Following properties available: application.properties

Name	Description
<code>spring.data.rest.base-path</code>	Base path used to expose repository resources
<code>spring.data.rest.default-page-size</code>	Default size of pages
<code>spring.data.rest.max-page-size</code>	Maximum size of pages
...	...

More properties available

www.luv2code.com/spring-boot-props

Spring Data REST Configuration

- Following properties available: application.properties

Name	Description
<code>spring.data.rest.base-path</code>	Base path used to expose repository resources
<code>spring.data.rest.default-page-size</code>	Default size of pages
<code>spring.data.rest.max-page-size</code>	Maximum size of pages
...	...

More properties available

www.luv2code.com/spring-boot-props

`spring.data.rest.*`

Sample Configuration

Sample Configuration

File: application.properties

```
spring.data.rest.base-path=/magic-api  
spring.data.rest.default-page-size=50
```

Sample Configuration

http://localhost:8080/magic-api/employees

File: application.properties

```
spring.data.rest.base-path=/magic-api  
spring.data.rest.default-page-size=50
```

Sample Configuration

http://localhost:8080/magic-api/employees

File: application.properties

```
spring.data.rest.base-path=/magic-api  
spring.data.rest.default-page-size=50
```

Returns 50
elements per page

Sorting

Sorting

- You can sort by the property names of your entity

Sorting

- You can sort by the property names of your entity
 - In our Employee example, we have: **firstName**, **lastName** and **email**

Sorting

- You can sort by the property names of your entity
 - In our Employee example, we have: **firstName**, **lastName** and **email**
- Sort by last name (ascending is default)
`http://localhost:8080/employees?sort=lastName`

Sorting

- You can sort by the property names of your entity
 - In our Employee example, we have: **firstName**, **lastName** and **email**
- Sort by last name (ascending is default)
`http://localhost:8080/employees?sort=lastName`
- Sort by first name, descending
`http://localhost:8080/employees?sort=firstName,desc`

Sorting

- You can sort by the property names of your entity
 - In our Employee example, we have: **firstName**, **lastName** and **email**
- Sort by last name (ascending is default)
`http://localhost:8080/employees?sort=lastName`
- Sort by first name, descending
`http://localhost:8080/employees?sort=firstName,desc`
- Sort by last name, then first name, ascending
`http://localhost:8080/employees?sort=lastName,firstName,asc`