How to get REST With Grails and Plugins

Jean Barmash

@jbarmash
jean@ctoschool.org

Agenda

- Native Capabilities
  - Converters
  - withFormat
  - Request Binding
- Grails Plugins
  - json-rest-api
  - marshallers
  - jax-rs
- One More Plugin...

Grails Features

Grails Features - Converters

```
import grails.converters.*

obj as JSON
obj as XML
```

- Configuration - http://grails.org/Converters+Reference

Grails Features – Custom Marshaller

```
import grails.converters.*
JSON.registerObjectMarshaller(Person) { p->
        return [name:”$p.first $p.last”]
}
obj as JSON
```

- Can also register a custom generic converter for domain classes (i.e. to remove default class attribute)

Marshshaller with Named Config

Register

- JSON.createNamedConfig('summary') { namedConfig ->

```
    namedConfig.registerObjectMarshaller(Person)
     return [first:it.firstName,
                last:it.lastName,
                phones: it.phones]
  }
}
```

- **Use**

```
JSON.use('summary') {
   person as JSON
}
```

- http://jwicz.wordpress.com/2011/07/11/grails-custom-xml-marshaller/
- http://adhockery.blogspot.com/2009/08/json-rendering-your-classes.html

Grails Features - withFormat

```
def list() {
    def books = Book.list()
    withFormat {
        html bookList:books
        json { render books as JSON }
        xml { render books as XML }
    }
}
```

Grails Features - Request Binding

- request.JSON
- request.XML

- def book = new Book(request.JSON)
- def book = new Book(request.XML)

Don't forget old friends

- Scaffolding
- i.e. Angular Plugin work by Robert Fletcher
-
- Base Controller Class

Helpful Plugins

Jax-RS

- http://grails.org/plugin/jaxrs
- compile ':jaxrs:0.6'
- Allows a lot of customization
- Nice scripts to help generate resources
- Feels Java-Like
- Seems Very Powerful
-

Marshallers Plugin

- http://grails.org/plugin/marshallers
- SRC http://github.com/pedjak/grails-marshallers
- DSL for Marshallers Configuration

```
class Author {
    static marshalling={
        xml {
            export {
                ignoreIdentifier true
                attribute 'dob'
                deep 'books'
                virtual {
                    popularBooks {author,xml->
                        author.findPopularBooks().each{ popularBook->
                            xml.startNode(xml.getElementName(popularBook))
                            xml.lookupObjectMarshaller(popularBook).marshalObject(popularBook,xml)
                            xml.end()
```

Json-Rest-API

- http://grails.org/plugin/json-rest-api
- Very nice model – you have to do nothing
- Expose domain class
  - static expose = 'books'
-

GET on /context/api/books – list instances

- POST on /context/api/books creates a new instance
- GET on /context/api/books/id retrieves the given instance
- PUT on /context/api/books/id updates the given instance by ID
- DELETE on /context/api/books/id deletes the given instance

Json-Rest-API

- Custom Retrieval Functions

```
static api = [
    excludedFields: [ "fullName" ]
    list : { params -> Person.list(params) },
    count: { params -> Person.count() }
 ]
```

- Relations Handling (Only GETs)
- Originally created for EXT-JS Integration by Matthias Hryniszak (padcom on github)
- Works with 2.X
- Registers its own custom marshaller

Json-Rest-API - Issues

- Does not support relations POST / PUTS
- No Searching
- No field selection from client side
- Author started another effort, restful GORM
- Numeric Ids only (i.e. /book/3)
-

Padcom's Restful-Gorm

- Some nice base functionality
- Parsing of URL (i.e. /authors/4/books/2)
- Generates an HQL Query
- Retrieves Data from GORM
-

One More Plugin...

Restful-GORM

- Based on Padcom's work
- Development Sponsored by Sefaira
  - Energy Modeling Engine
  - Cloud-Based API
  - Sefaira Concept – Energy Modeling for Architects
  -

RESTful-GORM

- Goal – easy to expose your domain model
- Preserve the approach of JSON-REST-API
- Convention over Configuration
- Takes advantage of the GORM Domain Classes API
- Uses default converters (so can override marshallers)
- Works for both XML & JSON (config)

RESTful-GORM – GET & Search

- List
  - Paging /books?max=10&offset=10
  - Multiple Objects /books/3,7
  - Fields Support /books?fields=title,pubDate
- Single Resource & Relations
  - /books/3
  - /books/3/authors
  - /books/3/authors/7
  - Relations based on association name
- Search - /search/books?attribute1=title&term1=”Peace”

POST

- POST creates a Resource
- Send JSON in Request Body
- POST /api/books – adds a new book
- POST /api/books/3/authors –
  - adds a new author and adds it to association of book
  - both hasMany and hasMany w/ belongsTo
-

PUT & DELETE

- PUT - update a resource, idempotent
  - PUT /books/3
- Linking to exiting object
  - PUT /books/3/authors/2
  - Links Book(3) to Existing author 2
- Anybody heard of HTTP PATCH (RFC 5789)
-
- DELETE
  - DELETE /books/3
  - Only simple deletes supported

Response Formatting - Success

- {

"status":200,

"code":300,

"message":"Object found",

"developerMessage":"",

"data": [

{

"id":3,"addresses":[],

"firstName":"Clint",

"lastName":"Eastwood",

"phones":[]}

]}

Response Formatting - Error


```
{       "status":404,
        "code":32,
        "message":"Object not found",
        "developerMessage":"",
        "error":
          {
                "status":404,
                "code":32,
                "message":"Object not found",
                "developerMessage":""}
                }
```

Extensibility

- Can customize the rendered JSON
- Can modify the request processing more
- Always return 200
  - ?returnHTTPOK=true or header variable
- Will need more extensibility / modularity
- Make more RESTful

Limitations / Future Directions

- No Caching
- Currently have to pick JSON or XML
- No Links / Locations by default
- White- / Blacklist of attributes
- Can support richer object creation by convention
- Currently no locking

Some Practical Thoughts

- Restful GORM is nice to get you started
- Contains a bunch of reusable components / conventions, i.e. how to format the response
- Usually need to provide custom APIs
- Extend RestfulGormController
-

Summary

- Grails has some very cool capabilities for RESTful API creation
  - Converters & Marshallers
  - Automatic Request Binding (request.JSON)
  - withFormat
- Greatly enhanced by plugin ecosystem

Questions ?