DSpace 7 Projections

Where they fit and what's been done.
From DSpace 7 WG meeting, Oct 17, 2019
Projections vs Excerpts

- **Projection**: An altered view of the domain model
- **Excerpt**: An automatic projection applied to a collection
Object scope

- Database
- DAOs
- Services
- Rest Repositories
- Controllers
- REST API
- JPA Entities
- Rest Objects
- HAL Resources
Object lifecycle

- **JPA Entity**

  **Model Conversion**
  - *converts properties*

  **Rest Object**
  - Type-specific Converters
  - Mostly from Repositories
  - Sometimes from Converters

- **Resource Construction**
  - *adds links and embeds*

  **HAL Resource**
  - Type-specific Resources
  - Mostly from Repositories
  - Sometimes from Controllers
Where do projections fit?

JPA Entity

Model Conversion
- converts properties

Rest Object

Resource Construction
- adds links and embeds

HAL Resource
1. Before conversion

Model Conversion
- converts properties

Resource Construction
- adds links and embeds

JPA Entity → (Entity -> Entity)

Most efficient place to omit properties

HAL Resource
2. During conversion

- Model Conversion: converts properties
  - (Entity->Rest) Highly flexible, but could get messy

- Resource Construction: adds links and embeds
3. After conversion

- **Model Conversion**: converts properties
- **Resource Construction**: adds links and embeds

_JPA Entity_ → _Rest Object_

_Rest Object_ → _HAL Resource_

(Rest -> Rest) _Easy place to add properties_
4. During resource construction

JPA Entity

Model Conversion
- converts properties

Rest Object

Resource Construction
- adds links and embeds

HAL Resource

(Rest -> Resource)
Easy place to omit or add links and embeds
What's been done

Model Conversion
- converts properties

(JPA Entity)
Most efficient place to omit properties

(Rest -> Rest)
Easy place to add properties

(Rest -> Resource)
Easy place to omit or add links or embeds

What’s been done

(JPA Entity)
Most efficient place to omit properties

(Rest -> Rest)
Easy place to add properties

(Rest -> Resource)
Easy place to omit or add links or embeds
Recommendation: Use Link Repositories for anything that returns a list

- More efficient than using a property on the Rest class
  - see notes in RestResourceController.findRellInternal
- Allows for late embedding
- Will make it easy to opt-out of including that data in certain projections (the "list" excerpt projection).
**Flowchart for Linking a Resource**

1. **Need to link a resource**
   - Yes: **Subresource?**
     - Yes: **Also embed?**
       - Yes: Use `HalLinkFactory`
       - No: **Simple and unique?**
         - Yes: **Conditionally?**
           - Yes: Use `@LinkRest` method annotation and Converter
           - No: **Pageable?**
             - Yes: Use `@LinkRest` method annotation and Converter
             - No: **Link manually from Resource or RestController**
   - No: **Embed manually from HalResource or RestController**

2. **Conditionally?**
   - Yes: Use `@LinkRest` method annotation and Converter
   - No: **Pageable?**
     - Yes: Use `@LinkRest` method annotation and Converter
     - No: **Link manually from Resource or RestController**

**Feedback Wanted:**
https://wiki.duraspace.org/pages/viewpage.action?pageId=164724742