How to make a resource auditable

The template-service now includes infrastructure to enable auditing state changes, along with an example of how to do so. Similar setup has already been introduced to the reference-data service, and will be added to the requisition service soon. The intent of this configuration is to easily allow:

1) Individual resources to be marked as auditable, and

2) The audit logs for these resources to be exposed via an endpoint like /api/someResources/{id}/auditLog

We’ve adopted JaVers for audit logging. To tell it to automatically log all state changes made to a resource via its Spring Repository, simply annotate the repository with @JaversSpringDataAuditable. This can currently be seen, for example, within the template service’s WidgetRepository.

The next step is to add an annotation like @TypeName(**"**myClassName**"**) to the class being audited. This annotation is similar to Spring’s @TypeAlias in that it provides an alternate name for a class. If @TypeName is specified, JaVers will use it rather that the class’s fully qualified name, thereby giving you the option to refactor the class’s name/package at a later time.

The next step is to create an endpoint to expose these logs. The /widgets/{id}/auditLog endpoint in the template service’s WidgetController illustrates how. Very little additional code is required.

Finally, a service’s RAML should be updated to reflect its new endpoint. This change is also designed to be minimal. Assuming the RAML already includes the definition for a resource like /widgets, the following is the only addition necessary to create an endpoint to expose the audit log for a specified instance of the widgets collection:

 /{id}/auditLog:
 type: instanceAuditLog

That’s all there is to it! A few additional pointers can be found in the “Audit Logging” section of the template service’s style guide. Of course, I hope you’ll let me know of any questions.

Thank you,

Ben