

we write about the things we build and the things we consume

 written by Fred van den Driessche on 17 April 2013 in Atlas, engineering

topics and annotations

Since the initial release of the 4.0 `/schedule` end-point late last year we've been hard at work on the next version of Atlas. Along with re-implementing the Cassandra persistence layer to use Protocol Buffers we've also been fleshing out the new `/topics` and `/topics/{id}/content` resources.

As Jonathan wrote in his post on API design, the overall aim is to make the Atlas 4.0 API highly flexible. The interface should be both simple and easy to learn but also give clients fine-grain control over data populated in query responses and so reduce the need for complex caching systems.

We've been implementing two of the features Jonathan described as we've been developing the new topics resources: making sure common resources are reusable as subordinate resources and enabling annotations on all end-points. The former is essentially a fancy way of saying the `/topics/{id}/content` resource should behave in the same way as the top-level `/content` resource but with the content set restricted to those related to the specified topic. In fact, `/topics/{id}/content` should resolve the same content as `/content?topics.topic.id={id}`. The latter, annotations, permit fine-grain control over the data returned from an end-point so a client can pick and choose only the aspects relevant to their needs. This post discusses how we think annotations will work in relation to the topics resources.

As a quick example, here's a sample response from the `/topics/{id}/content` end-point for reference:

one rule to, err, rule them all

Annotations apply to a field or set of fields of a particular resource type. For instance, `content.description` applies to `content` resources and exposes the `description` related fields (currently title, synopsis, image and thumbnail).

Content resources also have a `topics` array field activated by the `content.topics` annotation. Each of the objects in the array have a `topic` field containing a `topic` resource. Resources of type `topic` also have their own annotations, so how do we activate those `topic`-specific annotations nested inside `content`? The answer, currently, is to specify the path through the document to the resource you want to activate the annotations on, which in this case for the `topic id` annotation would be `content.topics.topic.id`. So that's the rule for activating an annotation: follow the fields through the object to the resource.

what are you implying?

Annotations applying to the request resource type can be specified without their type

prefix. For example, for requests to anything ending `/content` annotations are implicitly in the *content* context unless explicitly stated otherwise. So we can say `description` and that's implicitly converted to `content.description`. Likewise, `/topics` end-points imply a `topics` prefix.

When we're dealing with a multi-resource response for example in a `/topics/{id}/content` request the same rules apply. Because the request ends `/content` we're implicitly in a *content* context. So how do we activate annotations on the single topic in the response? Answer: we follow the path! So `topic.description`, for example.

an alternative

One alternative to the above scheme which we're considering is to apply an annotation to **all** resources present in the response related to that annotation. So `topic.id` would apply to both the single topic in a `/topics/{id}/content` request and the topics within the content in the response. We're currently undecided as to whether you'd need to explicitly provide the `content.topics` annotation to show the topics array or if that's implied by the presence of `topic.id`.

This method gets slightly more interesting in a *schedule* response. In a schedule we have both a channel and a list of content each of which has a broadcast which in turn have an attached channel. So when we add a channel annotation that would also apply to all the channels within the broadcasts in the content, the fun being that they'll all be the same channel because it's a schedule request.

unset in stone

The topics resources in Atlas 4.0 should be ready for an initial release in the not too distant future. As you might have guessed, no aspect of this annotations mechanism is at all final. As we move some of our internal projects over to use it we'll see how it fits but we'd also really appreciate any thoughts or input that you might have on the topic. Please get in touch with us, either on the [mailing list](#) or below in the comments.