Service Pipelines

SERVICE PIPELINES

Some data that is not stored in the models is available from services. The services use some of the same resource ids as the models. This allows them to be connected to model queries and other services connected to the API.

API-connected Services

Use the service axis to send a request to a service. The right hand side of the service step is the service id, for example, service::text_search. Pass parameters to the service using the filter syntax:

```
service::text_search[query_string$eq'abat']
```

An API service query creates a response that is very similar to a query using models and associations. This similarity includes many of the actual models in the API. The specific parameters, behavior and result schema depends on the individual service.

Pipes Connect Services and the API Models

The pipe axis is used to connect a service query stage to a model query stage. While the results of services are similar to the API data format, they are not identical. Also the services provide some associated data with the results, but not with the flexibility of RMA association paths.

Use the pipe axis with the pipe name on the right hand side to specify a pipe stage. Currently the only pipe is pipe::list. Use the filter syntax to pull parts of a query response out and assign them to a comma separated list.

```
pipe::list[probe_id$is'probes/probe/id']
```

The above pipe takes the result of a model::probe query from the API or a service result that contains a set of probes. It collects the id attributes of the probes into a comma separated list. It then assigns that list to the probe_id variable in the scope of the service pipeline query.

The pipe variable can be passed into an API model stage by using a "$" before the variable name in the filter syntax. The examples are shown on multiple lines for better readability.

```
service::differential[set$eqmouse][domain1$eq688][domain1_threshold$eq0,50][domain2$eq315]
[domain2_threshold$eq1,50][sort_order$eqdesc],
pipe::list[xid$eq'id'],
model::Gene[id$in$xid]
```

A pipe can also be used to pass data from a model query to a service.

```
model::Structure,rma::criteria,[acronym$il'ctx'],ontology[name$il'mouse*'],
pipe::list[sid$eq'id'],
service::differential[set$eqmouse][domain1$eq$sid][domain1_threshold$eq0,50][domain2$eq315]
[domain2_threshold$eq1,50][sort_order$eqdesc]
```

Multiple pipes can be used to create a more interesting service pipeline.

```
model::Structure,rma::criteria,[acronym$il'ctx'],ontology[name$il'mouse*'],
pipe::list[pid$eq'id'],
model::Structure[id$eq$pid],rma::include,child_structures,
pipe::list[cid$eq'child_structures/*/id'],
service::differential[set$eqmouse][domain1$eq$pid][domain1_threshold$eq0,50][domain2$in$cid]
[domain2_threshold$eq1,50][sort_order$eq'desc']
```