Catchment Strategies

Since module version:

1.2.0

This page list catchment strategies currently in use by implementations:

- Address hierarchies used to define geographic catchment areas. Boundaries might be artificial, i.e. different districts can be included, based on health care regions
- Location of enrollment
- No catchment areas - full copies stored on children

Development details

All strategies used to filtering objects implement the same generic interface named GenericFeedFilterStrategy.

```
public interface GenericFeedFilterStrategy {

   public String createFilterFeed(OpenmrsObject object);

   public boolean isFilterTagValid(String tag);
}
```

Method *createFilterFeed(OpenmrsObject object)* creates value of feed filter as XML for specific object or return null if object type is not supported.

Method *isFilterTagValid(String tag)* verify if the tag has the expected XML value.

Every feed filter strategy class should extend FeedFilterStrategy abstract class.

```
public abstract class FeedFilterStrategy {

        @Autowired
        private XMLParseService xmlParseService;

        protected XMLParseService getXmlParseService() {
                return xmlParseService;
        }

        protected abstract String getBeanName();
}
```

Class is used to store XMLParseService used to create XML from FeedFilter and vice versa.

Method getBeanName() should return bean name defined in Component annotation of strategy (see LocationFeedFilterStrategy as an example).

Configuration

In order to choose specific strategies you should change the configuration of the atom feed module. The "feedFilterBeans" section (please see example below) contains the list of bean names (name of spring component which is implemented the GenericFeedFilterStrategy interface).

In the default configuration the list is empty which means that neither strategy will be used, so there will be no filter tags included to the atom feed.

Note that feedFilterBeans list on parent instance should include all feedFilterBeans defined on child instances.

```
{
  "feedFilterBeans" : [ "beanName1", "beanName2" ],
  "feedConfigurations" : [ {
    "openMrsClass" : "org.openmrs.Obs",
    "enabled" : false,
    "title" : "Observation",
    "category" : "observation",
    "linkTemplates" : {
      "rest" : "/ws/rest/v1/obs/{uuid}?v=full",
      "fhir" : "/ws/fhir/Obs/{uuid}"
    },
    "feedWriter" : null
  }
  ]
}
```

How to use Strategies?

To create a custom strategy, create a bean of a class extending FeedFilterStrategy class and implementing GenericFeedFilterStrategy interface, and add its name to the Atomfeed configuration (as an element of feedFilters list).

When the atomfeed of an object will be created, it'll include the values returned by createFilterTag method of a bean, to the atomfeed.

Later, when pulling objects will be executed, method *isFilterTagValid* will be used to determine if atomfeed fulfill the prescribed conditions. See this as an example of using the strategy.

If there are multiple stratergies, all of theirs *isFilterTagValid will be executed, so if at least one of them returns false, the object won't be pulled.*

Warning

Be aware that feeds that was filtered out will be lost. That means if you change the configuration after pull was executed, previously filtered feeds won't be processed again!