REST Connector Superclass

Easy development of REST-based connectors is a very attractive feature. Therefore we have created an alternative way: the REST connector superclass. This is a small component in the polygon project:

https://github.com/Evolveum/polygon/tree/master/connector-rest

The component can be used as a superclass for your connector. It will provide basic configuration parameters (URL, username, ...) and connection management. So you will still need only to write the operation methods. But you will do that in Java, with proper type checking, versioning, dependency management, ConnId version checks and so on. All the things that a proper connector should have.

How To Use The Superclass

There are three basic things: dependency, configuration class and connector class.

Dependency

The dependency is simple. Just add the following dependency into your connector `pom.xml` file:

```
        <dependency>
                <artifactId>connector-rest</artifactId>
                <groupId>com.evolveum.polygon</groupId>
                <version>1.4.2.14-SNAPSHOT</version>
        </dependency>
```

(adjust the polygon version with the version that you use for connector parent)

Configuration Class

Your connector configuration class should be a subclass of `AbstractRestConfiguration`:

```
public class ExampleRestConfiguration extends AbstractRestConfiguration {
        // TODO
}
```

The `AbstractRestConfiguration` class defines the basic configuration variables for establishing a connection. Add any configuration variables that are specific to your connector code to this subclass.

Connector Class

Your connector class should be a subclass of `AbstractRestConnector` and it should be parametrized with the specific type of the configuration class:

```
@ConnectorClass(displayNameKey = "connector.example.rest.display", configurationClass =
ExampleRestConfiguration.class)
public class ExampleRestConnector extends AbstractRestConnector<ExampleRestConfiguration> implements TestOp,
SchemaOp {
        // TODO
}
```

The `AbstractRestConnector` class implements the basic initialization and disposal code, connection management and so on.

Implement The Operations

And that's it. Now go heade and implement the operations. E.g.:

```
    @Override
    public void test() {
        URIBuilder uriBuilder = getURIBuilder();
        URI uri;
        try {
            uri = uriBuilder.build();
        } catch (URISyntaxException e) {
            throw new IllegalArgumentException(e.getMessage(), e);
        }
        HttpGet request = new HttpGet(uri);

        HttpResponse response = execute(request);

        processResponseErrors(response);
    }
```

Examples

	Source code	**Description**
Example REST Connector	https://github.com/Evolveum/connector-rest-example	Dummy REST connector that illustrates basic connector structure. It does not really do much.
Drupal Connector	https://github.com/Evolveum/connector-drupal	Fully functional connector for Drupal using its REST API.

See Also

- Connector Development Guide