GraphQL API for use with Content Fragments

The GraphQL API used with Content Fragments is heavily based on the standard, open source GraphQL API.

The GraphQL API

"GraphQL is a data query language and specification developed internally by Facebook in 2012 before being publicly open sourced in 2015. It provides an alternative to REST-based architectures with the purpose of increasing developer productivity and minimizing amounts of data transferred. GraphQL is used in production by hundreds of organizations of all sizes..." See GraphQL Foundation.

For further information about the GraphQL API, see the following sections (amongst many other resources):

- At graphql.org:
  - Introduction to GraphQL
  - The GraphQL Specification
- At graphql.com:
  - Guides
  - Tutorials
  - Case Studies

Use Cases for Author and Publish Environments

The use cases can depend on the type of AEM as a Cloud Service environment:

- Publish environment; used to:
  - Query data for JS application (standard use-case)
- Author environment; used to:
  - Query data for "content management purposes":
    - GraphQL in AEM as a Cloud Service is currently a read-only API.
    - The REST API can be used for CR(u)D operations.

Schema Generation

The data schemas correlate to (are based on) the Content Fragment Models. The data schema caches are refreshed when you update a Content Fragment Model.

Permissions

The permissions are those required for accessing Assets.

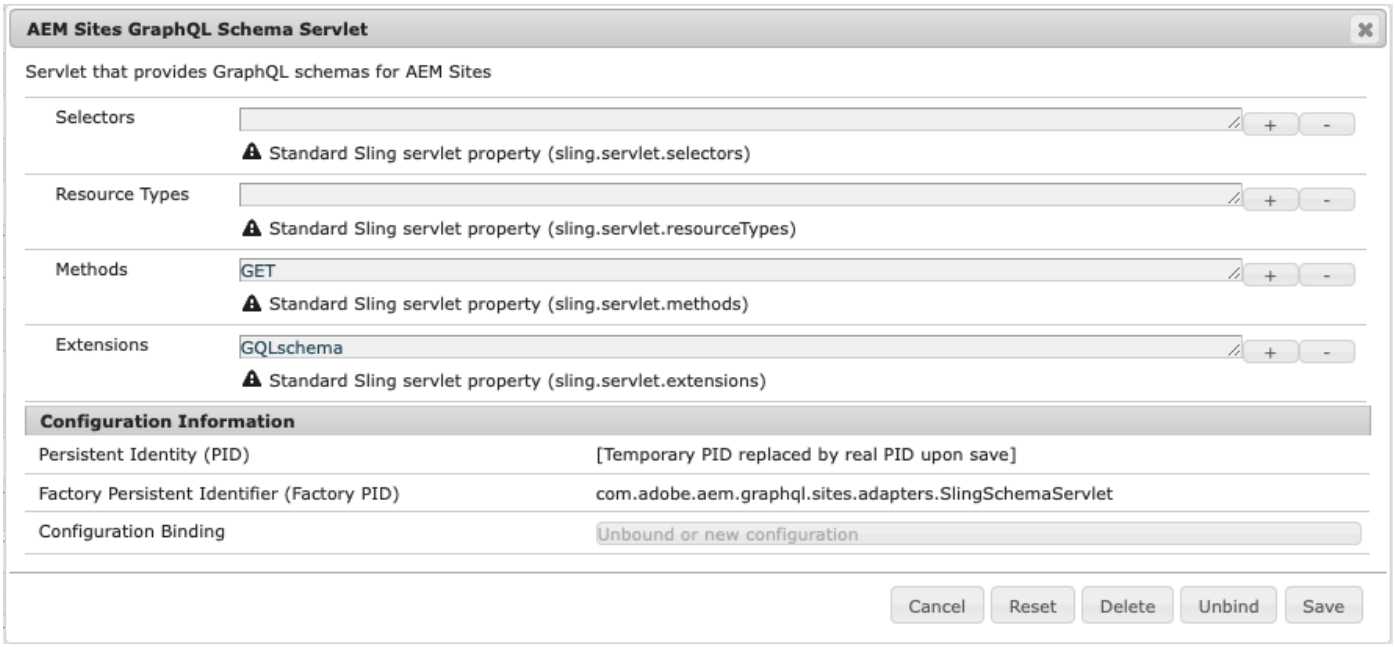
Filtering

See the Sample Queries for examples.

End-Points

To have access to GraphQL servlets in AEM you need to configure an endpoint. This also includes two OSGi configurations.

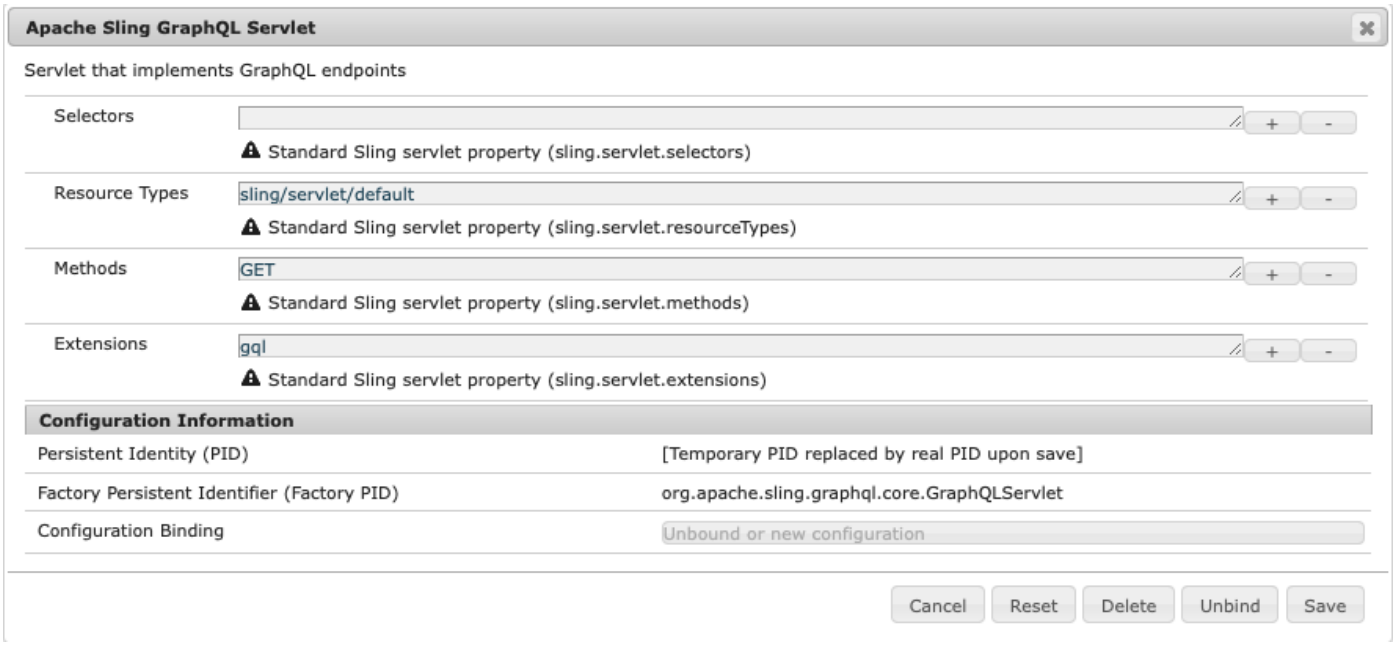
1. The Sling schema servlet that responds to requests to retrieve the GraphQL schema:



   - **Selectors** (`sling.servlet.selectors`)
     Must be left blank.
   - **Resource Types** (`sling.servlet.resourceTypes`)
     Define the resource type that the GraphQL servlet should listen to.
     For example:
     `graphql-enablement/components/endpoint` .

- **Methods**
  (`sling.servlet.methods^) The HTTP method the servlet should listen to; usually G ET`.
- **Extensions** (`sling.servlet.extensions`)
  Specify the extension that the Schema Servlet should respond to. In this case it is `GQLschema` , to be compatible with the GraphQL specifications.

2. The servlet that responds to graphql requests :



   - **Selectors** (`sling.servlet.selectors`)
     Must be left blank.
   - **Resource Type** (`sling.servlet.resourceTypes`)
     The resource type the GraphQL servlet should respond to.
     For example, `graphql-enablement/components/endpoint` .
   - **Methods** (`sling.servlet.methods`)
     The HTTP methods the GraphQL servlet should respond to, usually `GET` and `POST` .
   - **Extensions** (`sling.servlet.extensions`)
     The extension to listen for GraphQL requests, usually `gql` .
3. You now need to create an endpoint - a node of the sling:resourceType defined in these configurations.
   For example, to create an endpoint for retrieving the GraphQL Schema create a new node under `/apps/<my-site>/graphql` :
   - Name: `endpoint`
   - Primary Type: `nt:unstructured`
   - sling:resourceType: `graphql-enablement/components/endpoint`