GraphQL API

Xray provides a GraphQL API that allows users to perform CRUD operations directly on Xray entities. GraphQL gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.

Requests made to Xray's GraphQL API must also be authenticated using the same end point for REST API authentication.

The authentication process uses a **Client Id** and a **Client Secret**, both of which are based on an API Key created for a specific user in Xray Global Settings: API Keys global settings.

Thus, the first step for you is to obtain the token based on the Client Id and Client Secret of your assigned API Key. You can then use any GraphQL or REST API end points.

Xray GraphQL Schema Documentation

This is just an entry page that contains basic information about GraphQL API. You can access the complete schema documentation in the following URL:

https://us.xray.cloud.getxray.app/doc/graphql/

Xray GraphQL API endpoint

https://xray.cloud.getxray.app/api/v2/graphql

Discovering the Xray Cloud GraphQL API

GraphQL is introspective. This means you can query a GraphQL schema for details about itself.

Query `__schema` to list all types defined in the schema and get details about each:

```
query {
  __schema {
    types {
      name
      kind
      description
      fields {
        name
      }
    }
  }
}
```

Query `__type` to get details about any type:

```
query {
  __type(name: "Test") {
    name
    kind
    description
    fields {
      name
    }
  }
}
```

For more information about performing queries, see "Queries and Mutations."

Xray Cloud GraphQL resource limitations

- Node limit
- Resolver limit

As with any public API, the Xray GraphQL API protects against excessive or abusive calls to Xray Cloud's servers.

Node Limit

To pass schema validation, all Xray Cloud GraphQL API calls must meet these standards:

- Clients must supply a `limit` argument on any connection.
- Values of `limit` must be within 1-100.
- Individual calls cannot request more than 10,000 total items.

Calculating items in a call

These two examples show how to calculate the total number of items in a call.

```
{
  getTests(limit: 50) {
    total
    start
    limit
    results {
      issueId
      projectId
      preconditions(limit: 10) {
        total
        start
        limit
        results {
          issueId
          projectId
        }
      }
    }
  }
}

Calculation:

50         = 50 test issues
+
50 x 10  = 500 precondition issues

            = 550 total items
```

```
{
  getTests(limit: 50) {
    total
    start
    limit
    results {
      issueId
      projectId
      preconditions(limit: 10) {
        total
        start
        limit
        results {
          issueId
          projectId
        }
      }
      testRuns(limit: 50) {
        total
        start
        limit
        results {
          id
          status {
            name
            description
            color
          }
          testExecution {
            issueId
            projectId
            testPlans(limit: 5) {
              total
              start
              limit
              results {
                issueId
                projectId
              }
            }
          }
        }
      }
    }
  }
}

Calculation:

50              = 50 tests
+
50 x 10       = 500 preconditions
+
50 x 50       = 2,500 test runs
+
50 x 50 x 5  = 12,500 test plans

                 = 15,550 total items
```

Resolver limit

To pass schema validation, all Xray Cloud GraphQL API calls must meet these standards:

- Individual calls cannot use more than 25 resolvers.

Calculating resolvers in a call

These example show how to calculate the total number of resolvers in a call.

Query:

```
{
  getTestSets(limit: 50) {
    results {
      issueId
      projectId
      tests(limit: 10) {
        results {
          issueId
          projectId
          testType {
            name
          }
          status {
            name
          }
          preconditions(limit: 10) {
            results {
              issueId
              projectId
            }
          }
        }
      }
    }
  }
}

Calculation:

getTestSets + tests + testType + status + preconditions = 5 resolvers
```

Connection

Connections let you query related objects as part of the same call. With connections, you can use a single GraphQL call where you would have to use multiple calls to a REST API.

It's helpful to picture a graph: dots connected by lines. The dots are nodes, the lines are edges. A connection defines a relationship between nodes.

Item

Item is a generic term for an object. You can look up a item directly, or you can access related items via a connection. If you specify a `item` that does not return a scalar, you must include subfields until all fields return scalars.

Resolver

Resolver is a generic term for a query, mutation or complex field. A complex field is any field in Xray Cloud GraphQL API that is not a scalar with the exception of the fields results.

Guides and other useful links

Here are some links that provide extra information and tools related with GraphQL:

- GraphQL Documentation - here you can find general information about all aspects of GraphQL.
- Insomnia - a powerful HTTP client with native GraphQL support.