Sample GraphQL queries

|  |
| --- |
| Query Schema |
| {  \_\_schema {  queryType {  fields {  name  description  }  }  }  } |

|  |
| --- |
| Query Application Types |
| {  \_\_type(name: "Application") {  kind  fields {  type {  name  description  }  }  }  } |

|  |
| --- |
| Using Fragments and Conditions. Notice the use of alias “svcname” |
| fragment appInfo on Application {  services(limit: 10) {  svcname: nodes {  name  deploymentType  }  }  createdBy {  name  id  }  description  }  query getAppInfo($input: String!, $withId: Boolean!) {  applicationByName(name: $input) {  name  id @include(if: $withId)  ...appInfo  }  } |

|  |
| --- |
| Using named queries |
| mutation createApp {  createApplication(input: {name: "test\_demoapp"}){  clientMutationId  }  }  query getAppInfo {  applicationByName(name: "test\_demoapp") {  name  id  description  }  }  mutation updateApp {  updateApplication(input: {applicationId: "F0Qyzrq5Rby0Iy0kgYXoXg", description: "Testing update"}){  clientMutationId  }  } |

|  |
| --- |
| Using “batch” mode to perform multiple operations, notice the use of alias |
| mutation {  first: updateApplication(input: {applicationId: "Tmx5DEZZS-ONlw40s0bAFw", description: "Testing first"}) {  clientMutationId  }  second: updateApplication(input: {applicationId: "U4EZCETFSLCJLA-iF6cW6A", description: "Testing second"}) {  clientMutationId  }  } |

|  |
| --- |
| Use of “...” notation to access implementation |
| {  audits(limit: 2, offset: 8) {  nodes {  changes {  appName  operationType  }  ... on UserChangeSet {  id  triggeredAt  triggeredBy {  id  name  }  }  }  }  } |