

Inline Fragments

Like many other type systems, GraphQL schemas include the ability to define interfaces and union types. [Learn about them in the schema guide.](#)

If you are querying a field that returns an interface or a union type, you will need to use *inline fragments* to access data on the underlying concrete type. It's easiest to see with an example:

```
query HeroForEpisode($ep: Episode!) {  
  hero(episode: $ep) {  
    name  
    ... on Droid {  
      primaryFunction  
    }  
    ... on Human {  
      height  
    }  
  }  
}  
  
{  
  "ep": "JEDI"  
}
```

```
{  
  "data": {  
    "hero": {  
      "name": "R2-D2",  
      "primaryFunction": "Astromech"  
    }  
  }  
}
```

In this query, the `hero` field returns the type `Character`, which might be either a `Human` or a `Droid` depending on the `episode` argument. In the direct selection, you can only ask for fields that exist on the `Character` interface, such as `name`.

To ask for a field on the concrete type, you need to use an *inline fragment* with a type condition. Because the first fragment is labeled as `... on Droid`, the `primaryFunction` field will only be executed if the `Character` returned from `hero` is of the `Droid` type. Similarly for the `height` field for the `Human` type.

Named fragments can also be used in the same way, since a named fragment always has a type attached.

Meta fields

Given that there are some situations where you don't know what type you'll get back from the GraphQL service, you need some way to determine how to handle that data on the client. GraphQL allows you to request `__typename`, a meta field, at any point in a query to get the name of the object type at that point.

```

{
  search(text: "an") {
    __typename
    ... on Human {
      name
    }
    ... on Droid {
      name
    }
    ... on Starship {
      name
    }
  }
}

{
  "data": {
    "search": [
      {
        "__typename": "Human",
        "name": "Han Solo"
      },
      {
        "__typename": "Human",
        "name": "Leia Organa"
      },
      {
        "__typename": "Starship",
        "name": "TIE Advanced x1"
      }
    ]
  }
}

```

In the above query, `search` returns a union type that can be one of three options. It would be impossible to tell apart the different types from the client without the `__typename` field.

GraphQL services provide a few meta fields, the rest of which are used to expose the **Introspection** system.