

Quand nos requête ont le même nom, il faut créer un **alias** pour pas qu'il y ait de conflits.

localhost:4000/salutGraphQL?query=query%7B%0A%20%20appleInfo%20%3A%20company(id%3A"1")%20%7B%0A

GraphiQL



Prettify

History

```
1 query{
2   appleInfo : company(id:"1") {
3     name,
4     user {
5       firstName,
6       age
7     }
8   },
9   googleInfo : company(id:"2") {
10    name,
11    user {
12      firstName,
13      age
14    }
15  }
16 }
17
```

```
{
  "data": {
    "appleInfo": {
      "name": "Apple",
      "user": [
        {
          "firstName": "Mathieu",
          "age": 26
        },
        {
          "firstName": "Mickey",
          "age": 29
        }
      ]
    },
    "googleInfo": {
      "name": "Google",
      "user": [
        {
          "firstName": "Sarah",
          "age": 25
        }
      ]
    }
  ]
}
```

Le fragment nous permet de mettre nos infos dans un array qu'on éclate ensuite avec l'opérateur spread de es6.

GraphQL



Prettify

History

```
1 query{
2   appleInfo : company(id:"1") {
3     name,
4     user {
5       ...userDetails
6     }
7   },
8   googleInfo : company(id:"2") {
9     name,
10    user {
11      ...userDetails
12    }
13  }
14 }
15
16 fragment userDetails on User{
17   firstName,
18   age,
19   id
20 }
21
```

```
{
  "data": {
    "appleInfo": {
      "name": "Apple",
      "user": [
        {
          "firstName": "Mathieu",
          "age": 26,
          "id": "2"
        },
        {
          "firstName": "Mickey",
          "age": 29,
          "id": "4"
        }
      ]
    },
    "googleInfo": {
      "name": "Google",
      "user": [
        {
          "firstName": "Sarah",
          "age": 25,
          "id": "3"
        }
      ]
    }
  ]
}
```

Docume

Q Search Schema

A GraphQL schema
kind of operation.

ROOT TYPES

query: RootQuery