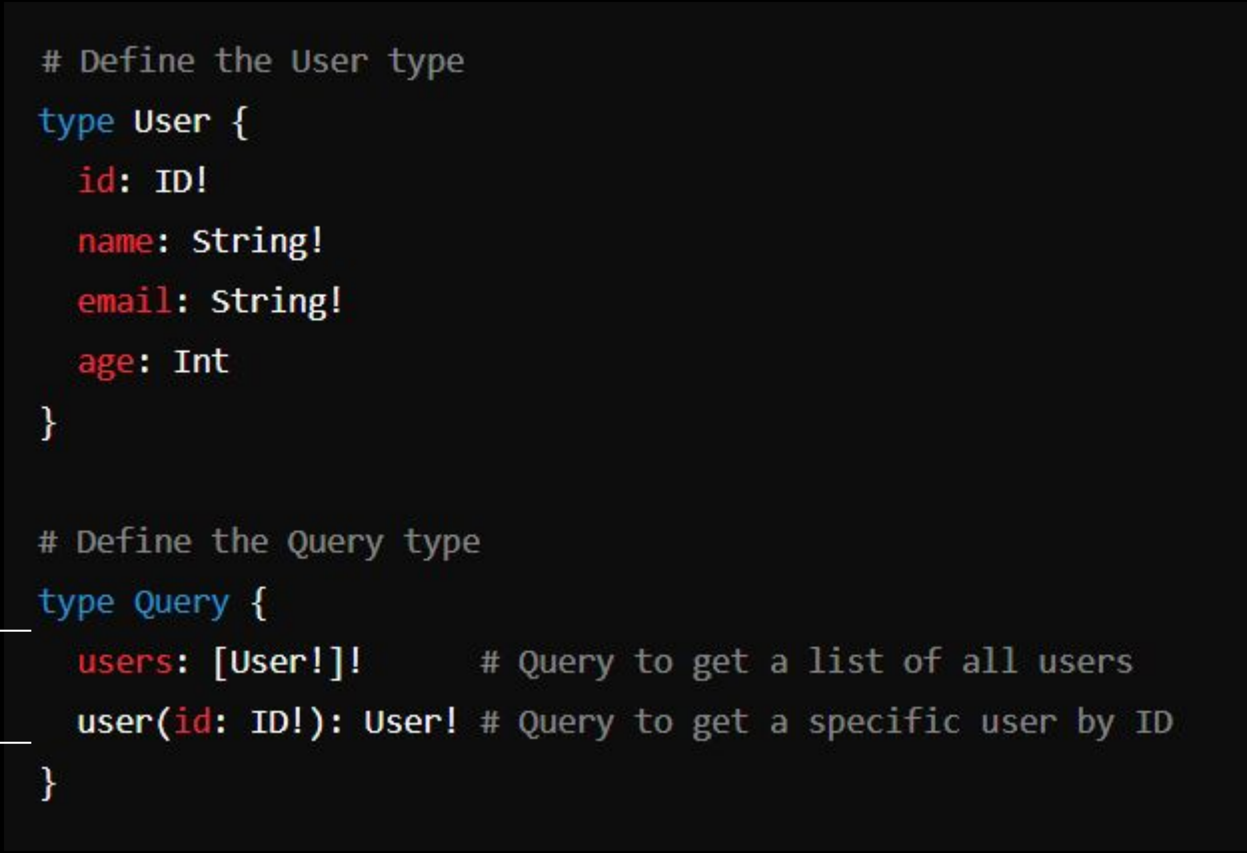
Graph QL

Graph QL Schema

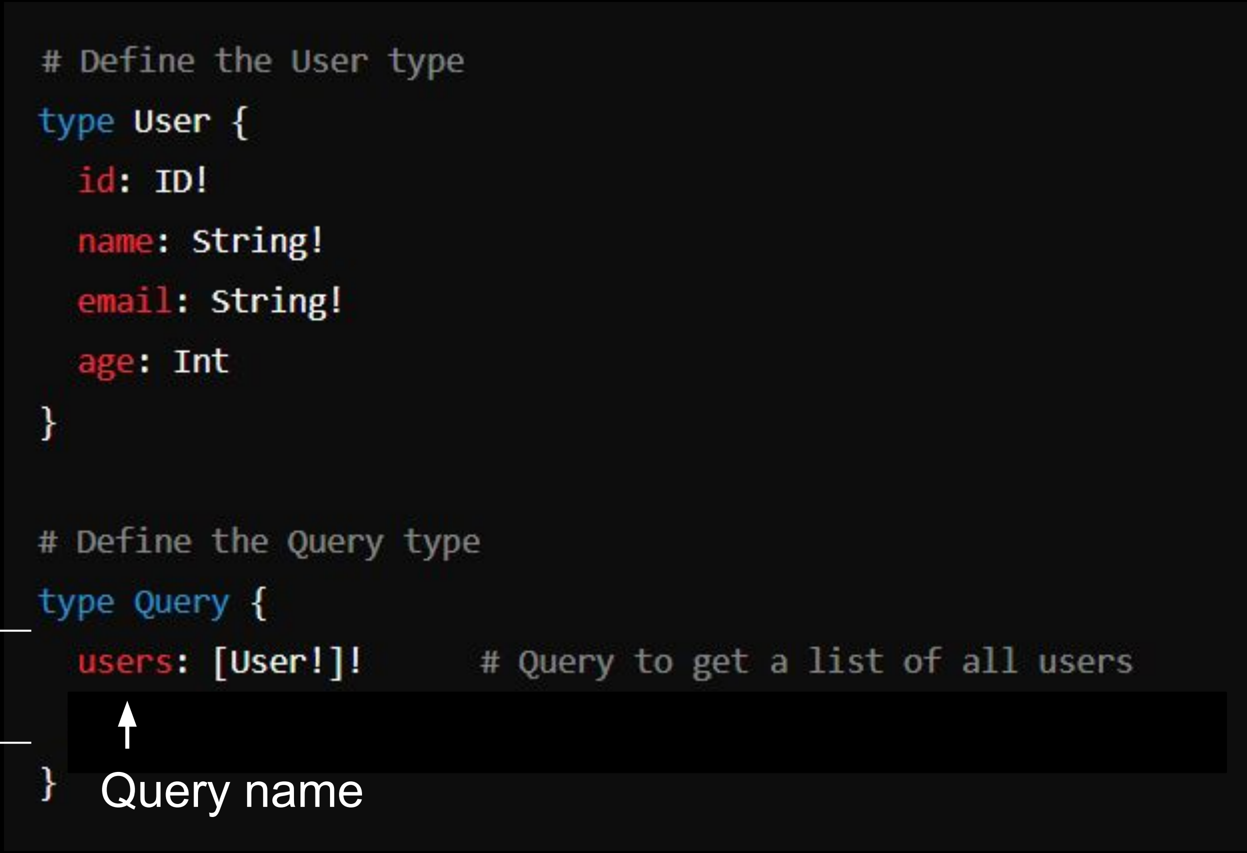
- **Schema** defines the types and structure of data that can be queried
- Each schema typically has one **type Query:**
  - Fields within this type are the queries that can be made by client
- Custom types are often also included in the schema, such as User.
  - defines what data is included when a client requests a User.

```graphql
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

Queries that can be made by client



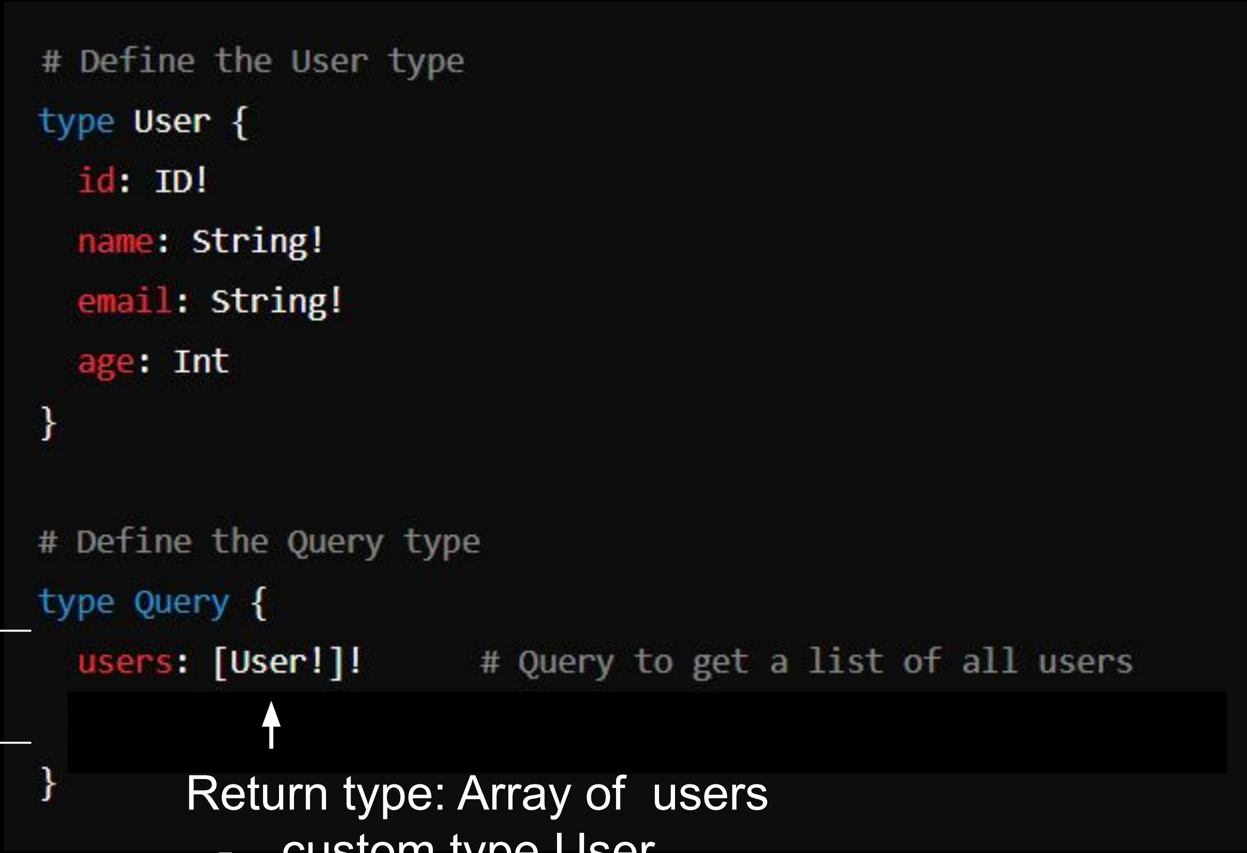
```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```



```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  [redacted]
}
```

Queries that can be made by client

Query name



```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  [redacted]
}
```

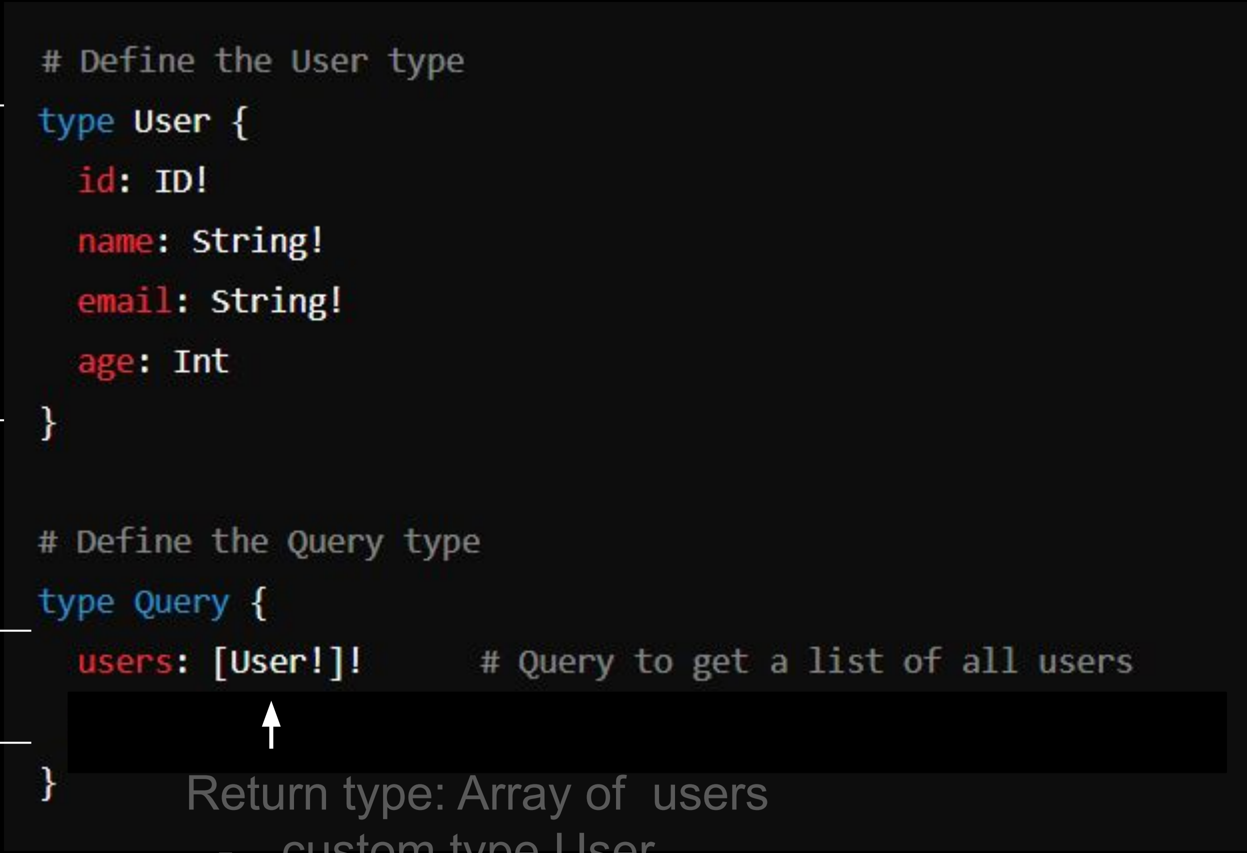
Queries that can be made by client

Return type: Array of users

- custom type User
- ! means non-nullable

Custom User Type

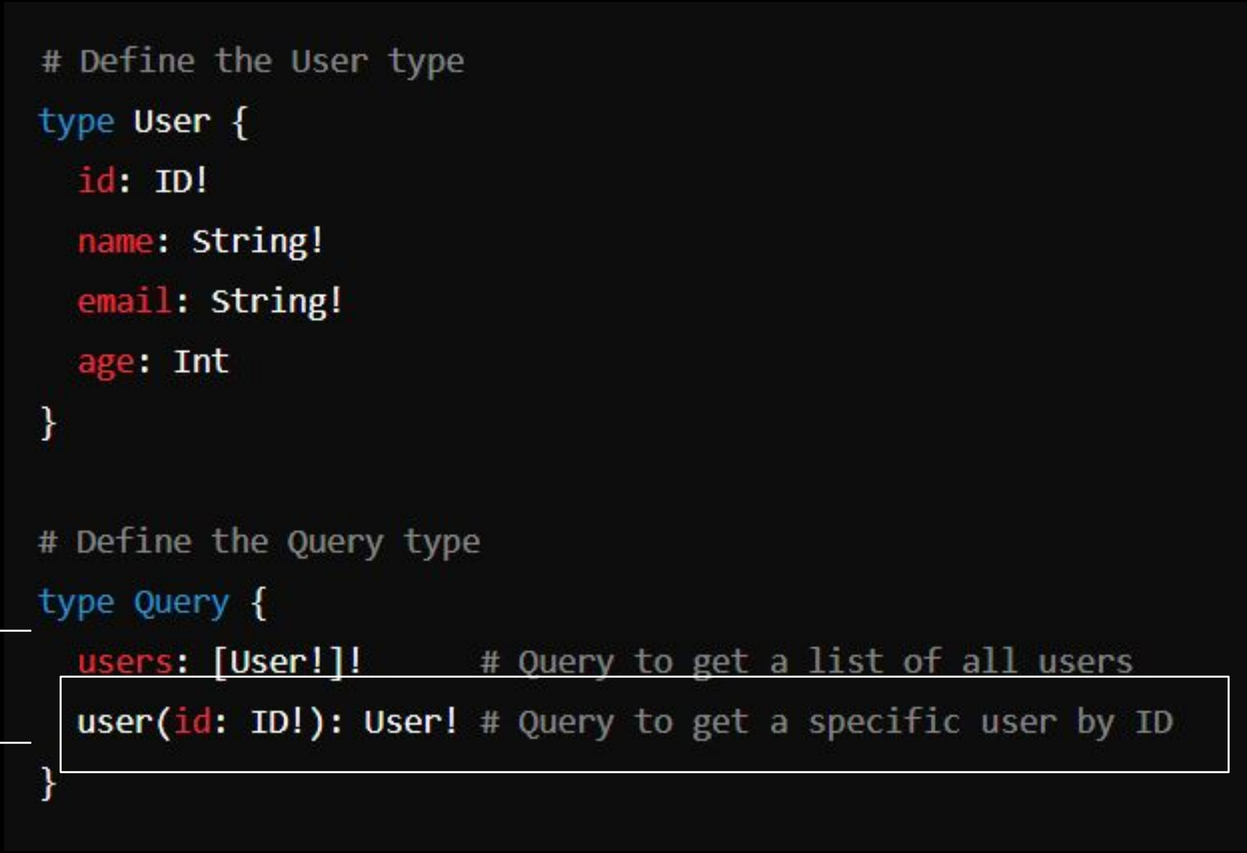
Queries that can be made by client



```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  [redacted]
}
```

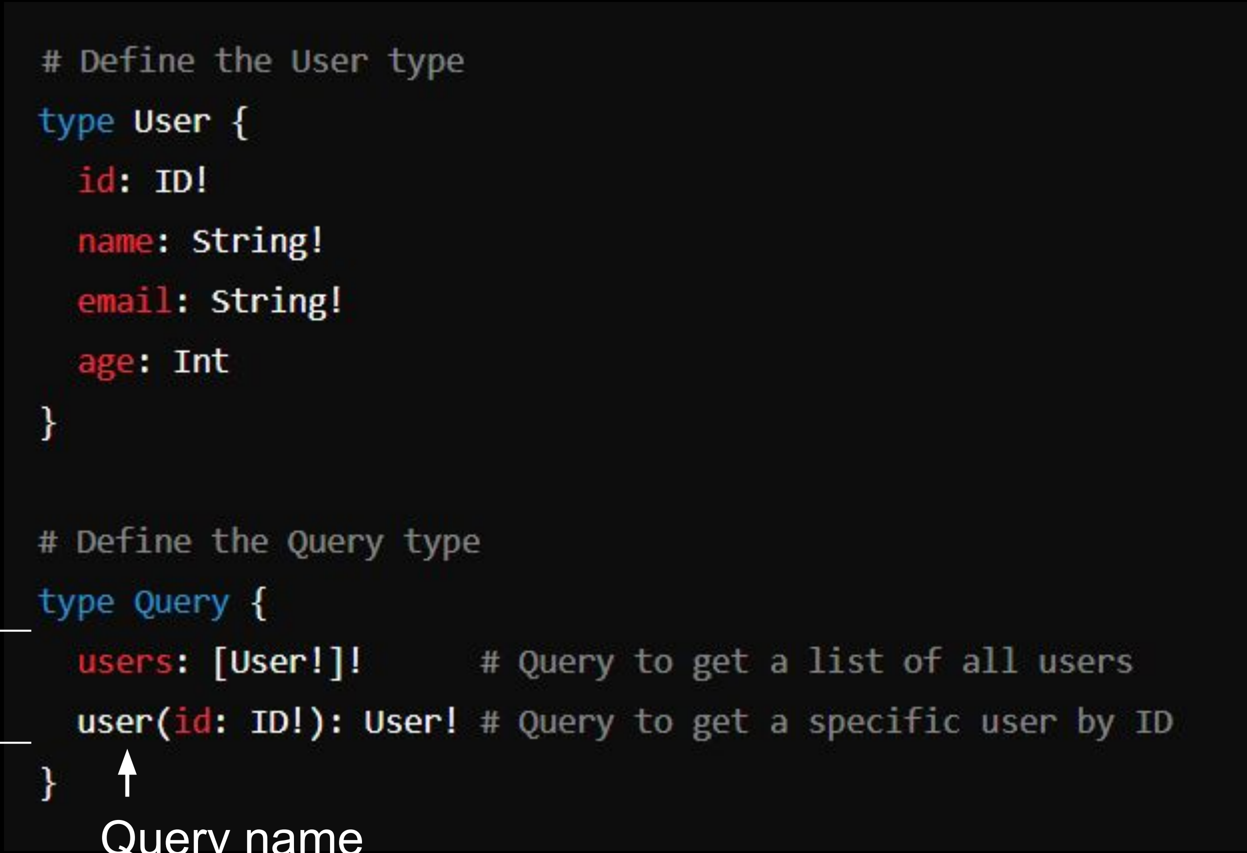
Return type: Array of users

- custom type User
- ! means non-nullable



```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

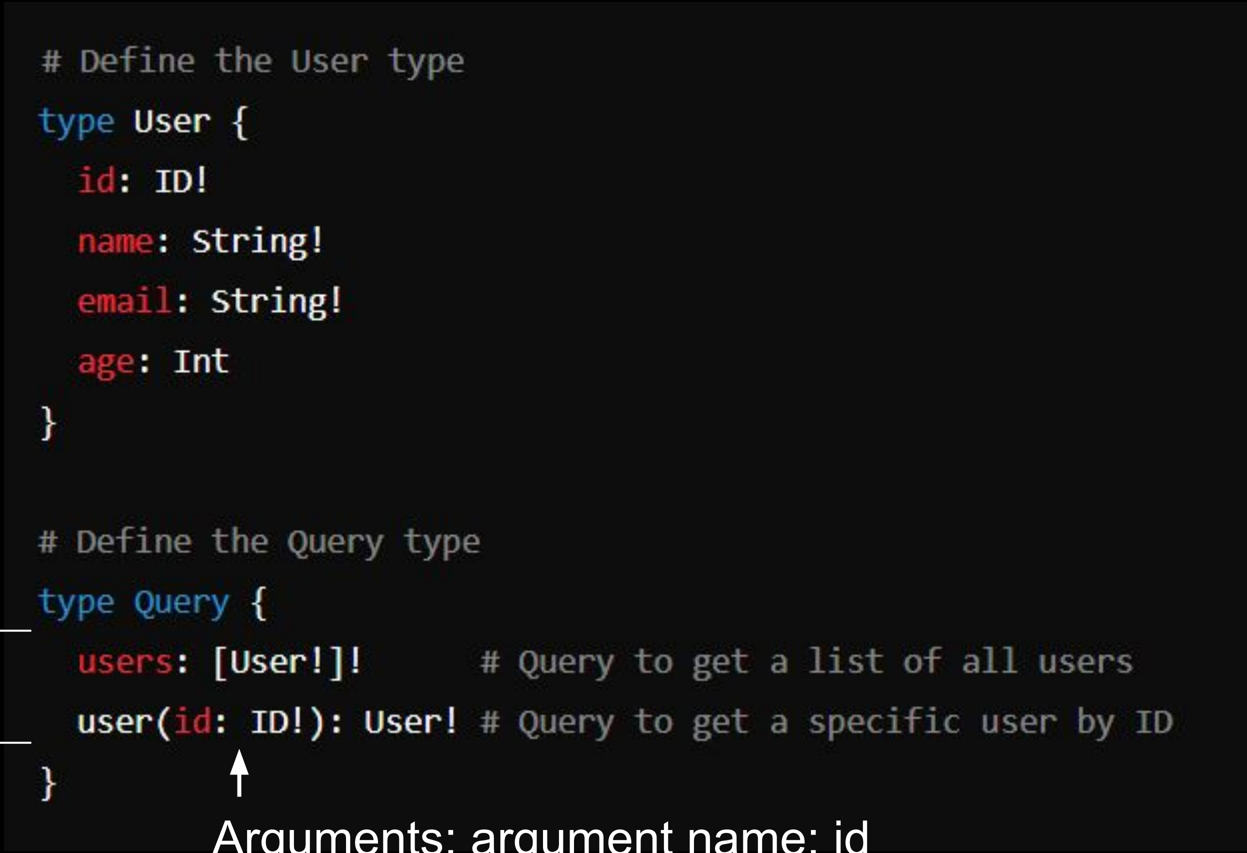
Queries that can be made by client



```
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

Queries that can be made by client

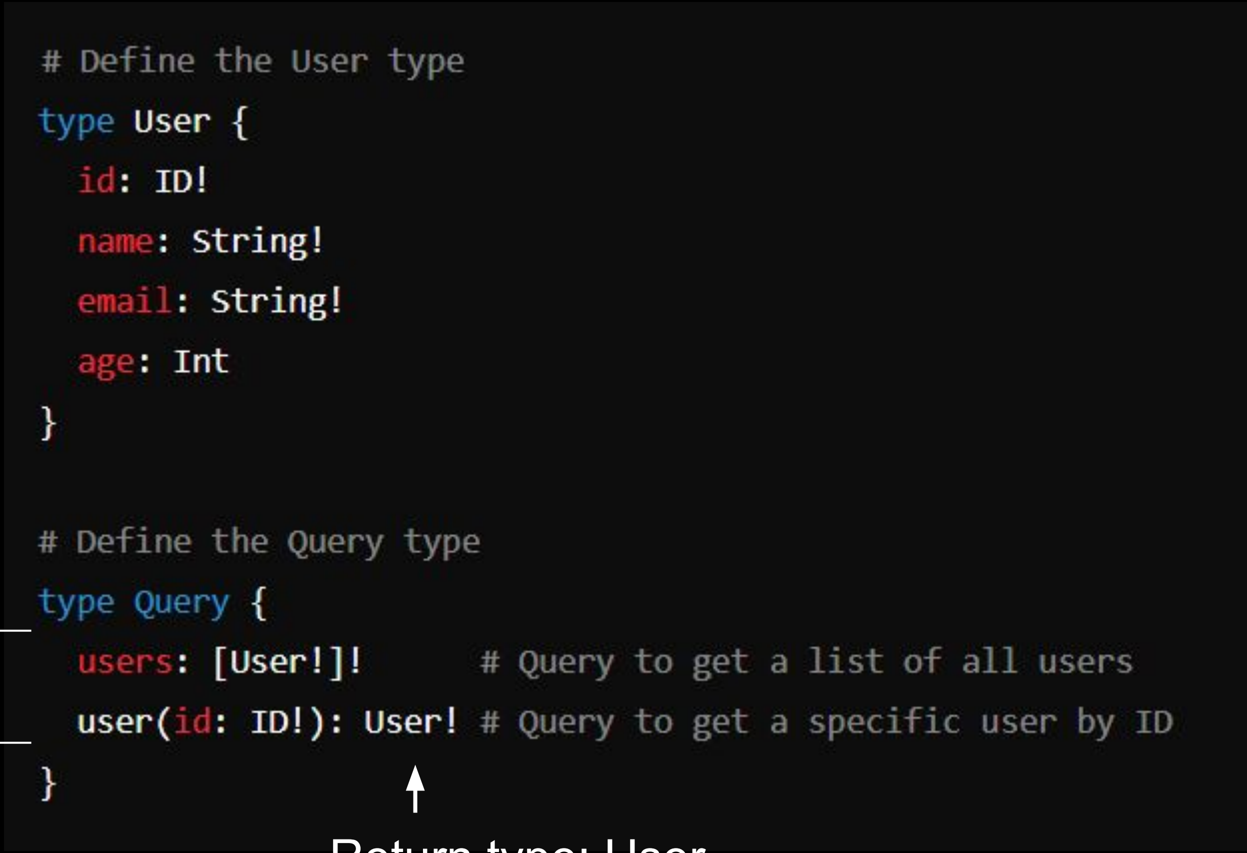
Query name



```graphql
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

Queries that can be made by client

Arguments: argument name: id
argument type: ID → built in type

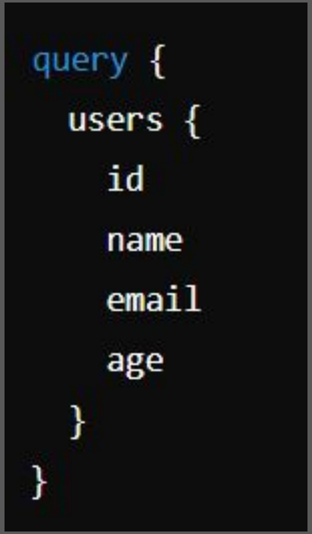


```graphql
# Define the User type
type User {
  id: ID!
  name: String!
  email: String!
  age: Int
}

# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

Queries that can be made by client

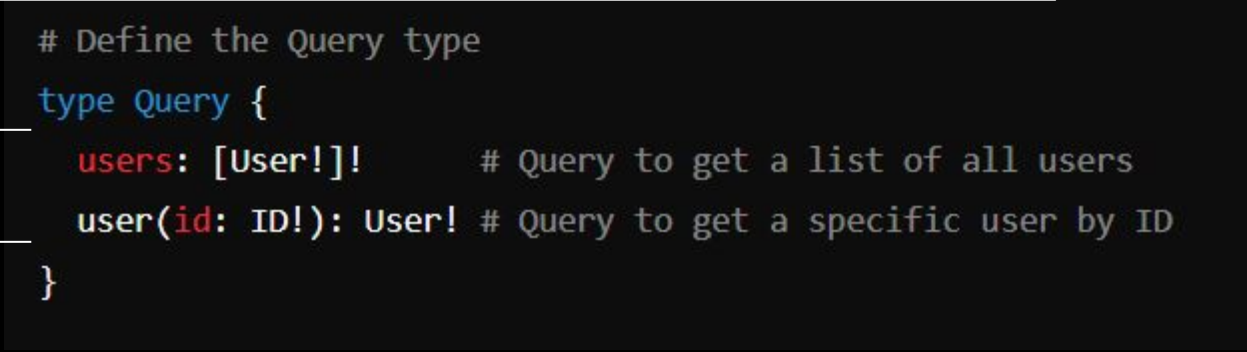
Return type: User

Get all users:



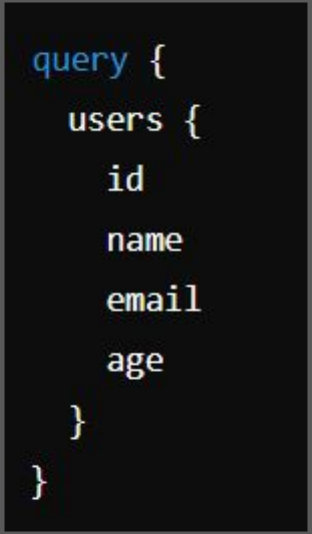
```graphql
query {
  users {
    id
    name
    email
    age
  }
}
```

Queries that can be made by client



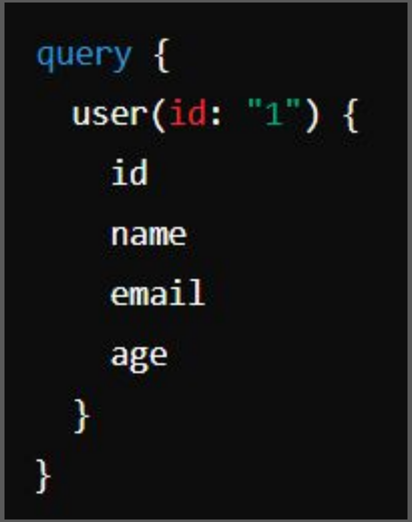
```graphql
# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```

Get all users:



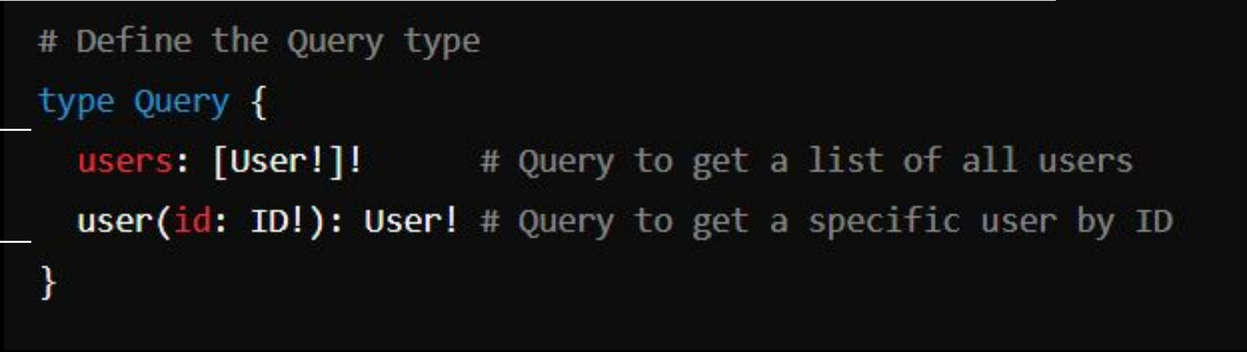
```
query {
  users {
    id
    name
    email
    age
  }
}
```

Get one specific user:

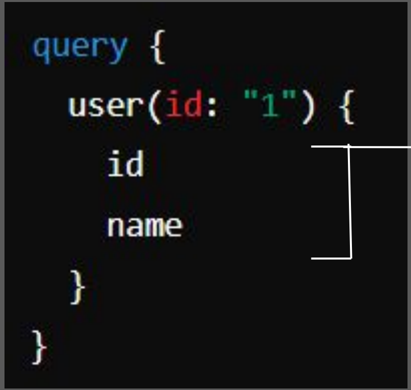


```
query {
  user(id: "1") {
    id
    name
    email
    age
  }
}
```

Queries that can be made by client

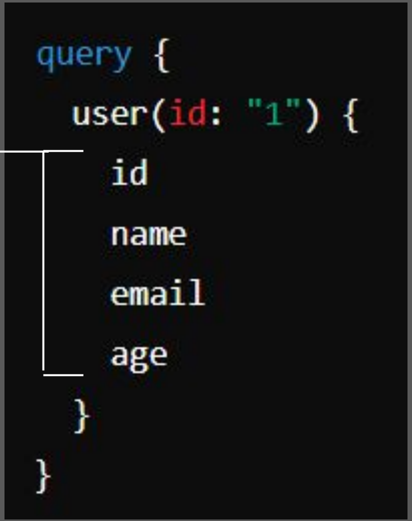


```
# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```



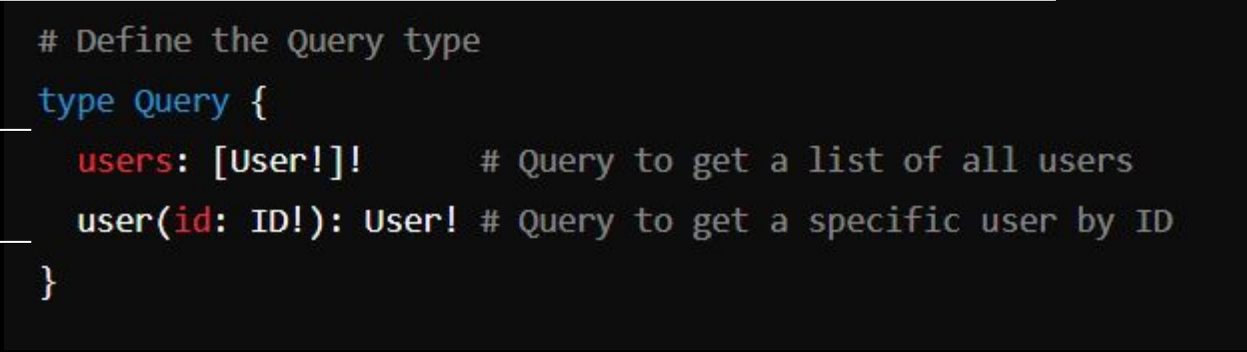
```
query {
  user(id: "1") {
    id
    name
  }
}
```

Requested fields

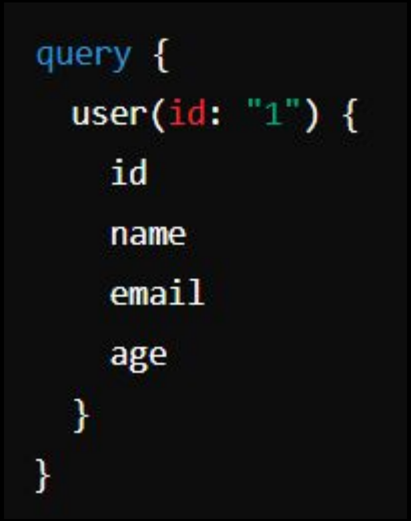


```
query {
  user(id: "1") {
    id
    name
    email
    age
  }
}
```

Queries that can be made by client



```
# Define the Query type
type Query {
  users: [User!]!      # Query to get a list of all users
  user(id: ID!): User! # Query to get a specific user by ID
}
```



```
query {
  user(id: "1") {
    id
    name
    email
    age
  }
}
```

This is an unnamed query, we can also name it

```graphql
query GetUser($userId: ID!) {
  user(id: $userId) {
    id
    name
    email
  }
}
```

```
[redacted] {
    "userId": "123"
  }
```