

<https://www.youtube.com/watch?v=eIQh02xuVw4>

GraphQL Explained in 100 Seconds

[Fireship](#)

<https://www.youtube.com/watch?v=Dr2dDWzThK8&t=0s>

GraphQL Tutorial Beginner - Learn GraphQL in NodeJS / ExpressJS

[PedroTech](#)

<https://www.youtube.com/watch?v=YyUWW04HwKY>

GraphQL With React Tutorial - Apollo Client

[PedroTech](#)

**\*\***[https://www.youtube.com/playlist?list=PLpPqplz6dKxXICtNgHY1tiCPau\\_AwWAJU](https://www.youtube.com/playlist?list=PLpPqplz6dKxXICtNgHY1tiCPau_AwWAJU)

GraphQL Course - Beginner To Expert

[PedroTech](#)

\_ast updated on Sep 30, 2021

<https://welearncode.com/beginners-guide-graphql>

[https://www.tutorialspoint.com/graphql/graphql\\_environment\\_setup.html](https://www.tutorialspoint.com/graphql/graphql_environment_setup.html)

# OVERVIEW

<https://www.apollographql.com/tutorials/lift-off-part1/01-feature-overview-and-setup>

← → ↺

https://www.apollographql.com/tutorials/lift-off-part1/01-feature-overview-and-setup

A ODYSSEY

🔍 🗨️ ⏪

Lift-off I: Basics

Feature overview and setup ✓

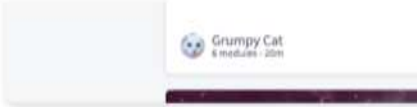
Feature data requirements ✓

Schema definition language (SDL) ✓

Building our schema ✓

Apollo Server ✓

1. FEATURE OVERVIEW AND SETU



To build this feature, we'll use a "schema" which data our client application needs.

- 1. Defining the schema:** We identify whi that data as intuitively as possible.
- 2. Backend implementation:** We build o whichever data sources contain it. In tl connect our app to a live REST data so
- 3. Frontend implementation:** Our client

# 1. start Server/client (FEATURE OVERVIEW AND SETUP)

## 1.1 clone the app's starter repository

```
git clone https://github.com/apollographql/odyssey-lift-off-part1  
C:\Users\terry\Downloads\graphql
```

```
PS C:\Users\terry> cd C:\Users\terry\Downloads\graphql  
PS C:\Users\terry\Downloads\graphql> git clone https://github.com/apollographql/odyssey-lift-off-part1  
Cloning into 'odyssey-lift-off-part1'...  
remote: Enumerating objects: 842, done.  
remote: Counting objects: 100% (840/840), done.  
remote: Compressing objects: 100% (431/431), done.  
remote: Total 842 (delta 442), reused 720 (delta 392), pack-reused 2Receiving objects: 99% (834/842),  
1.98 MiB | 3.93 MiB/s  
Receiving objects: 100% (842/842), 2.94 MiB | 4.42 MiB/s, done.  
Resolving deltas: 100% (442/442), done.
```

## 1.2 server start (4000?)

```
in server/ directory install dependencies  
npm install  
run the app:  
npm start
```

```
PS C:\Users\terry\Downloads\graphql> cd .\odyssey-lift-off-part1\  
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1> cd server  
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> npm install  
added 33 packages, and audited 34 packages in 828ms  
4 packages are looking for funding  
  run `npm fund` for details  
3 moderate severity vulnerabilities  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
Run `npm audit` for details.  
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> npm start  
> catstronauts-server-complete@1.0.0 start  
> nodemon src/index  
[nodemon] 2.0.20  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: js,mjs,json  
[nodemon] starting `node src/index.js`  
[nodemon] clean exit - waiting for changes before restart
```

## 1.3 client start 3000

**client/** install dependencies and start the app:

npm install

npm start

```
PS C:\Users\terry> cd C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client
```

```
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client> npm install
```

```
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
```

```
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
```

```
npm WARN deprecated @types/testing-library__dom@7.5.0: This is a stub types definition. testing-library__dom provides its own type definitions, so you do not need this installed.
```

```
npm WARN deprecated @types/classnames@2.3.1: This is a stub types definition. classnames provides its own type definitions, so you do not need this installed.
```

```
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
```

```
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
```

```
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
```

```
added 737 packages, and audited 738 packages in 16s
```

```
122 packages are looking for funding
```

```
run `npm fund` for details
```

```
2 moderate severity vulnerabilities
```

```
To address all issues, run:
```

```
npm audit fix
```

```
Run `npm audit` for details.
```

```
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client> npm start
```

```
> catstronauts-client@1.0.0 start
```

```
> vite
```

```
VITE v4.4.4 ready in 380 ms
```

```
→ Local: http://127.0.0.1:3000/
```

```
→ press h to show help
```

running app at <http://127.0.0.1:3000/>, or localhost:3000.



## **x2. (text only) FEATURE DATA REQUIREMENT**

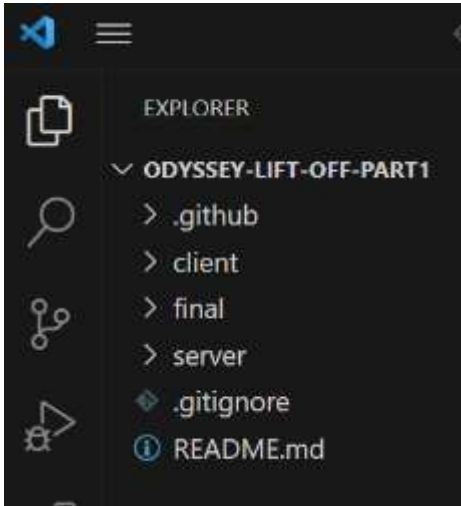
xxx

## **x3. (text only) SCHEMA DEFINITION LANGUAGE (SDL)**

xxxx

## 4. @apollo/server / create server schema.js (BUILDING OUR SCHEMA)

### 4.1 open in IDE



### 4.2 npm install @apollo/server graphql graphql-tag

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
powershell - server

PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1> cd server
P
t-off-part1\server>
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> npm install @apollo/server graphql graphql-tag

added 134 packages, and audited 168 packages in 11s

14 packages are looking for funding
  run `npm fund` for details

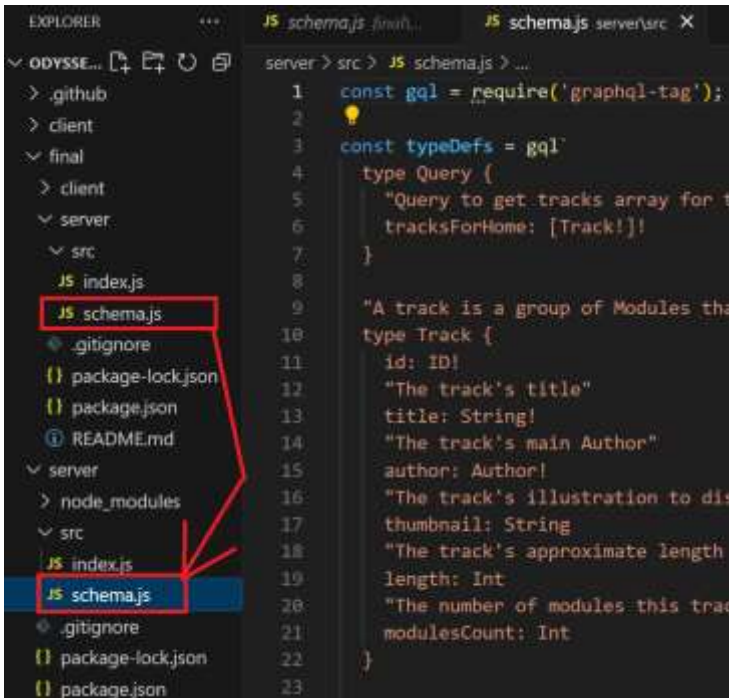
3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> |
```

## 4.3 copy schema.js

TERRY: just copy



schema.js should be this

```
const gql = require("graphql-tag");
```

```
const typeDefs = gql`  
  type Query {  
    "Get tracks array for homepage grid"  
    tracksForHome: [Track!]!  
  }  
`
```

"A track is a group of Modules that teaches about a specific topic"

```
type Track {  
  id: ID!  
  "The track's title"  
  title: String!  
  "The track's main author"  
  author: Author!  
  "The track's main illustration to display in track card or track page detail"  
  thumbnail: String  
  "The track's approximate length to complete, in minutes"  
  length: Int  
  "The number of modules this track contains"  
  modulesCount: Int  
}
```

"Author of a complete Track"

```
type Author {  
  id: ID!  
  "Author's first and last name"  
  name: String!  
  "Author's profile picture url"  
  photo: String  
}
```

```
`;
```

```
module.exports = typeDefs;
```

## 5. setup mock (APOLLO SERVER)

### 5.1 VSC terminal install mock/schema

PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> npm install @graphql-tools/mock  
@graphql-tools/schema

added 6 packages, changed 3 packages, and audited 174 packages in 4s

14 packages are looking for funding  
run `npm fund` for details

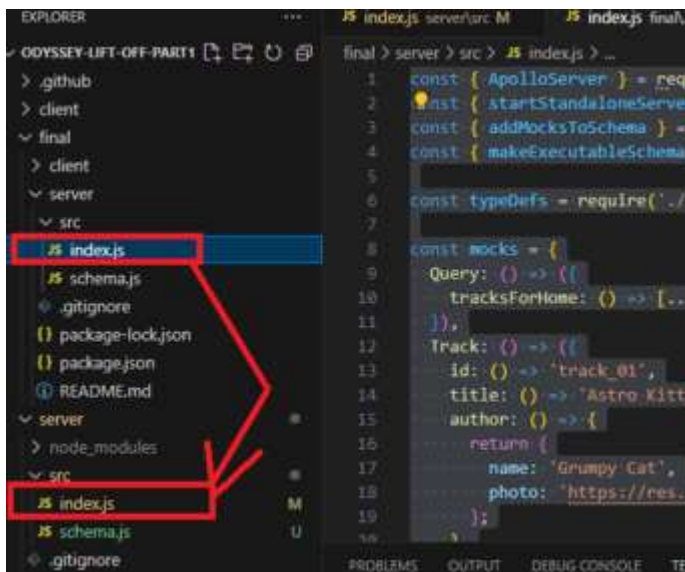
3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:  
npm audit fix --force

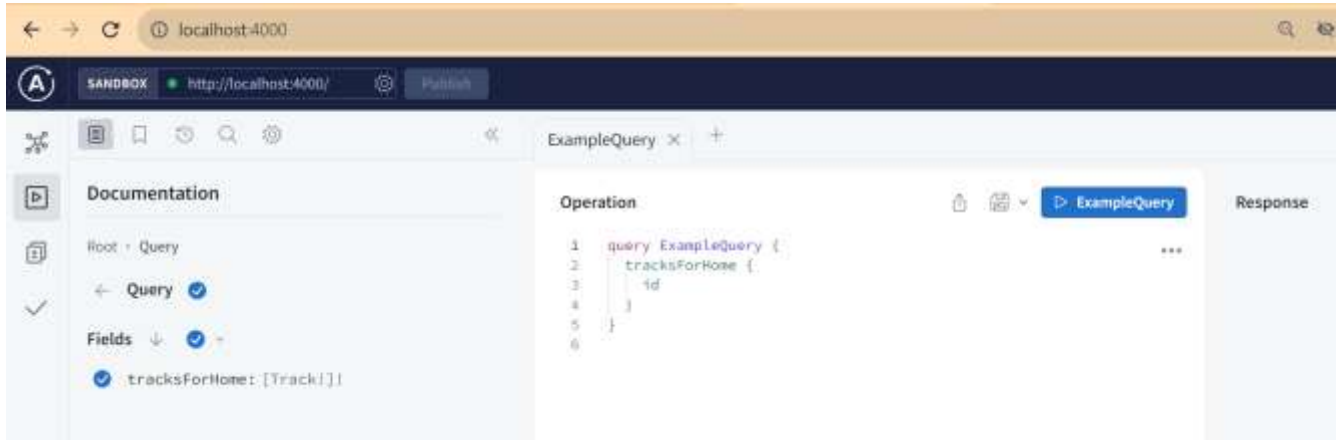
Run `npm audit` for details.

PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server>

### 5.2 copy mock code



## 5.3 refresh localhost:4000 (explorer)



## 6. using explorer (APOLLO EXPLORER)

### 6.1 example query

Fields ↓ ✓ ↓

✓ tracksForHome: [Track!]! →

Query to get tracks array for the homepage grid

Documentation

Root > Query > tracksForHome

← tracksForHome: [Track!]! ✓

Query to get tracks array for the homepage grid

Metadata for Track Type

A track is a group of Modules that teaches about a specific topic

Fields ↓ ⊕ ⊖

⊕ author: Author!  
✓ id: ID!  
⊕ length: Int  
⊕ modulesCount: Int  
⊕ thumbnail: String  
⊕ title: String!

Operation 📄 📄 ▶ ExampleQuery

```
1 query ExampleQuery {  
2   tracksForHome {  
3     id  
4   }  
5 }  
6
```

Response 🔍 📄



```
{  
  "data": {  
    "tracksForHome": [  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      },  
      {  
        "id": "track_01"  
      }  
    ]  
  }  
}
```

## 6.2 Add title field to query

Fields   

 tracksForHome: [Track!]! 




Query to get tracks array for the homepage grid


 tracksForHome: [Track!]! 


Query to get tracks array for the homepage grid


Metadata for Track Type


A track is a group of Modules that teaches about a specific topic


Fields   


 author: Author!

 id: ID!

 length: Int

 modulesCount: Int



 thumbnail: String

 title: String!

Add to query

Documentation




Root • Query • tracksForHome


 tracksForHome: [Track!]! 


Query to get tracks array for the homepage grid


Metadata for Track Type


A track is a group of Modules that teaches about a specific topic


Fields   


 author: Author!




 id: ID!

 length: Int



 modulesCount: Int

 thumbnail: String

 title: String!

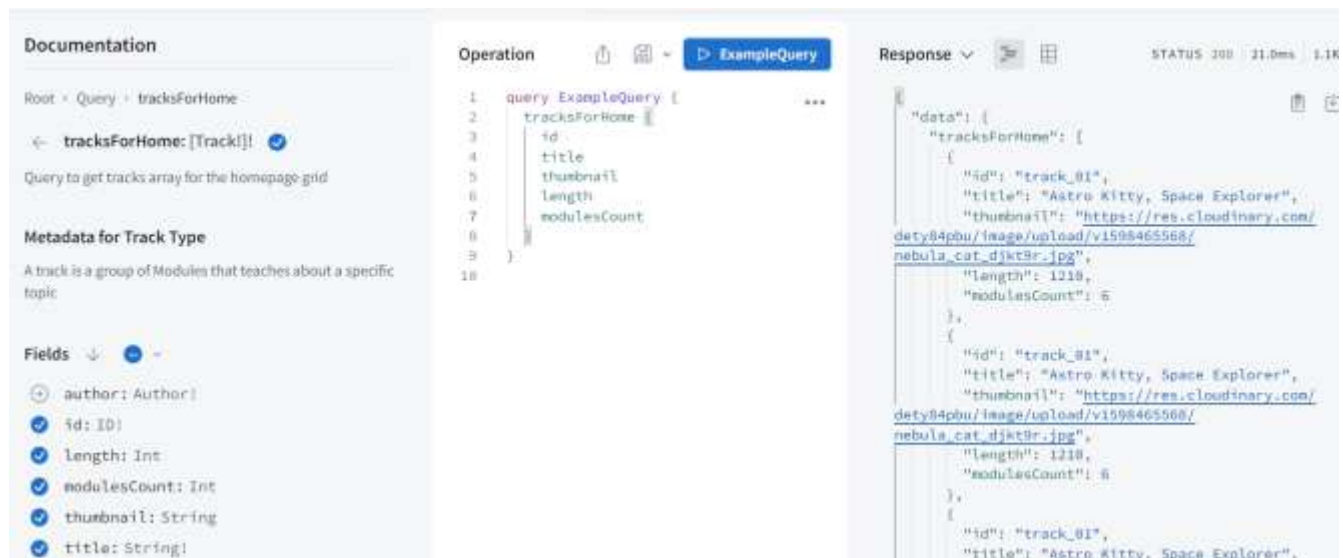
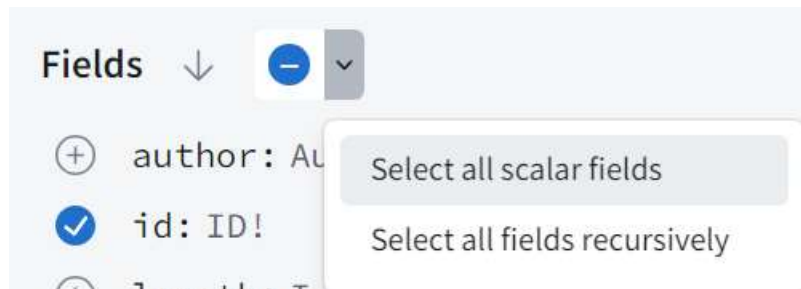
Operation    ExampleQuery

```
1 query ExampleQuery {
2   tracksForHome {
3     id
4     title
5   }
6 }
7
```

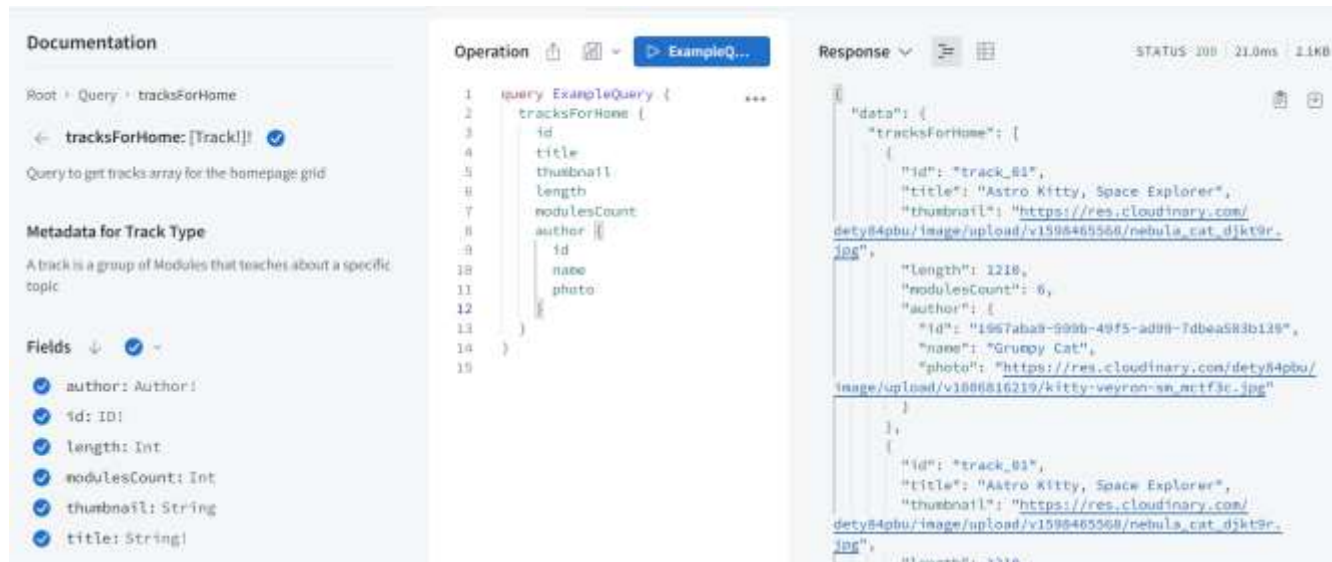
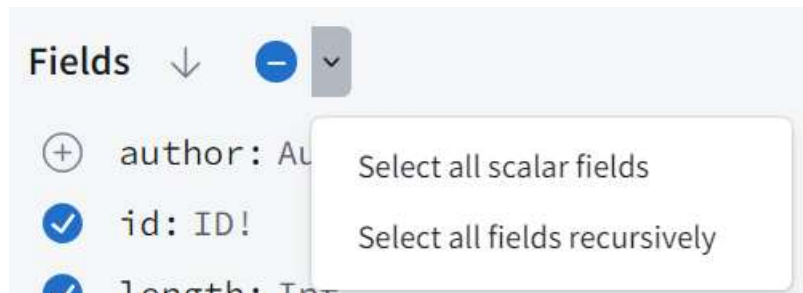
Response   STATUS 200 30.0ms

```
{
  "data": {
    "tracksForHome": [
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer"
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer"
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer"
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer"
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer"
      }
    ]
  }
}
```

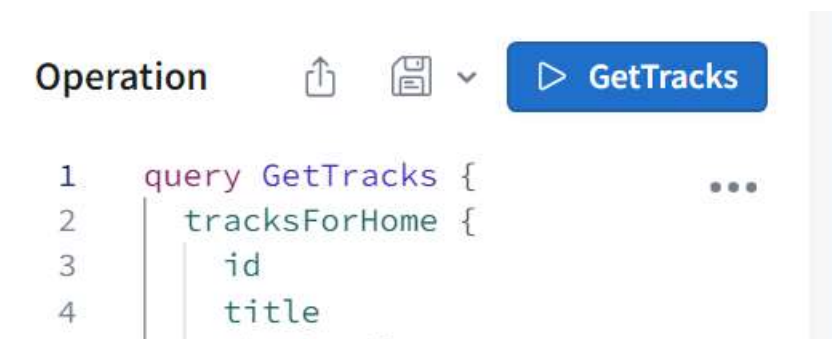
## 6.3 select all scalar fields



## 6.4 select all fields recursively



## 6.5 rename (select and change)



## 6.6 Save operation

Operation



▶ GetTracks

```
1 query G
2   tracksForHome {
3     id
```

Save operation



**Log in to your Studio account**  
to save your operation

Log in

APOLLO



[Redacted]  
[Redacted]@gmail.com

**Authorize access to your Studio account**

<http://localhost:4000> needs access to your Studio account .

Cancel

Authorize



Log in to your Studio account  
to save your operation

Log in

Error here... you have to always go to login page when clicking save button

I never see this dialog

## Save Operation

Operation Name\*

TracksForHome

Which collection would you like to save this operation to?\*

 Sandbox ▾ Sandbox / My Collection ▾

Cancel

Save

Operation



▶ ExampleQuery

```
1 query ExampleQuery {  
2   | tracksForHome {  
3     | id  
4   | }  
5 }  
6
```



## x7. Frontend first steps

TERRY: this was already started.

```
client > src > JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import GlobalStyles from './styles';
4  import Pages from './pages';
5
6  ReactDOM.render(
7    <React.StrictMode>
8      <GlobalStyles />
9      <Pages />
10    </React.StrictMode>,
11    document.getElementById('root')
12  );
13
```

## 8. Apollo Client 3

From the `client/` directory, run the following command:

```
npm install graphql @apollo/client
```

```
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client> npm install graphql @apollo/client
```

added 13 packages, and audited 751 packages in 7s

122 packages are looking for funding

run ``npm fund`` for details

2 moderate severity vulnerabilities

To address all issues, run:

```
npm audit fix
```

Run ``npm audit`` for details.

```
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client>
```

change `index.js` to this

```
import React from "react";
import ReactDOM from "react-dom";
import GlobalStyles from "./styles";
import Pages from "./pages";
import { ApolloProvider, ApolloClient, InMemoryCache } from "@apollo/client";
```

```
const client = new ApolloClient({
  uri: "http://localhost:4000",
  cache: new InMemoryCache()
});
```

```
ReactDOM.render(  
  <div>  
    <GlobalStyles />  
    <Pages />  
  </div>  
)
```

```
<ApolloProvider client={client}>
<GlobalStyles />
<Pages />
</ApolloProvider>,
document.getElementById("root")
);
```

```
client > src > JS index.js > ...
1  import React from "react";
2  import ReactDOM from "react-dom";
3  import GlobalStyles from "../styles";
4  import Pages from "../pages";
5  import { ApolloProvider, ApolloClient, InMemoryCache } from "@apollo/client";
6
7  const client = new ApolloClient({
8    uri: "http://localhost:4000",
9    cache: new InMemoryCache()
10 });
11
12 ReactDOM.render(
13   <ApolloProvider client={client}>
14     <GlobalStyles />
15     <Pages />
16   </ApolloProvider>,
17   document.getElementById("root")
18 );|
```

127.0.0.1:3000



**Catstronaut**  
Kitty space academy

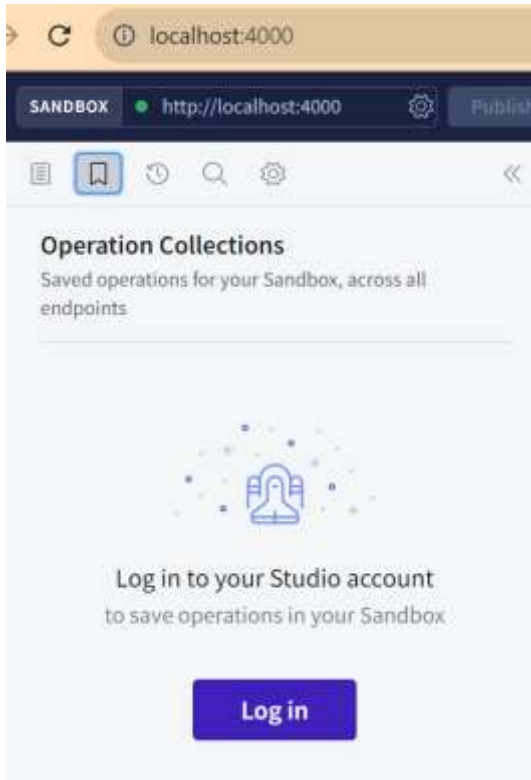
## 9. defining a query

### 9.1 tracks.js

```
client > src > pages > JS tracks.js > ...
 1  import React from 'react';
 2  import { Layout } from '../components';
 3  import { gql } from '@apollo/client';
 4  ⚡
 5  const TRACKS = gql`|
 6  /**
 7   * Tracks Page is the Catstronauts home pa
 8   * We display a grid of tracks fetched wit
 9   */
10  const Tracks = () => {
11    return <Layout grid> </Layout>;
12  };
13
14  export default Tracks;
15
```

## 9.2 i cant view operation collections

always asks me to login



## 10. useQuery hook

```
client > src > pages > JS tracks.js > [⌘] Tracks
1  import React from 'react';
2  import { Layout } from '../components';
3  import { gql } from '@apollo/client';
4
5  /** TRACKS query to retrieve all tracks */
6  const TRACKS = gql`
7    query GetTracks {
8      tracksForHome {
9        id
10       title
11       thumbnail
12       length
13       modulesCount
14       author {
15         id
16         name
17         photo
18       }
19     }
20   }
21 `;
22 /**
23  * Tracks Page is the Catstronauts home page.
24  * We display a grid of tracks fetched with useQuery with
25  * ⚡/
26  const Tracks = () => {
27    const { loading, error, data } = useQuery(TRACKS);
28
29    if (loading) return "Loading...";
30
31    if (error) return `Error! ${error.message}`;
32
33    return <Layout grid>{JSON.stringify(data)}</Layout>;
34  };
35
36  export default Tracks;
37
```

save autorestart... nothing

npm start in vscode .... nothing

```
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client> npm start

> catstronauts-client@1.0.0 start
> vite

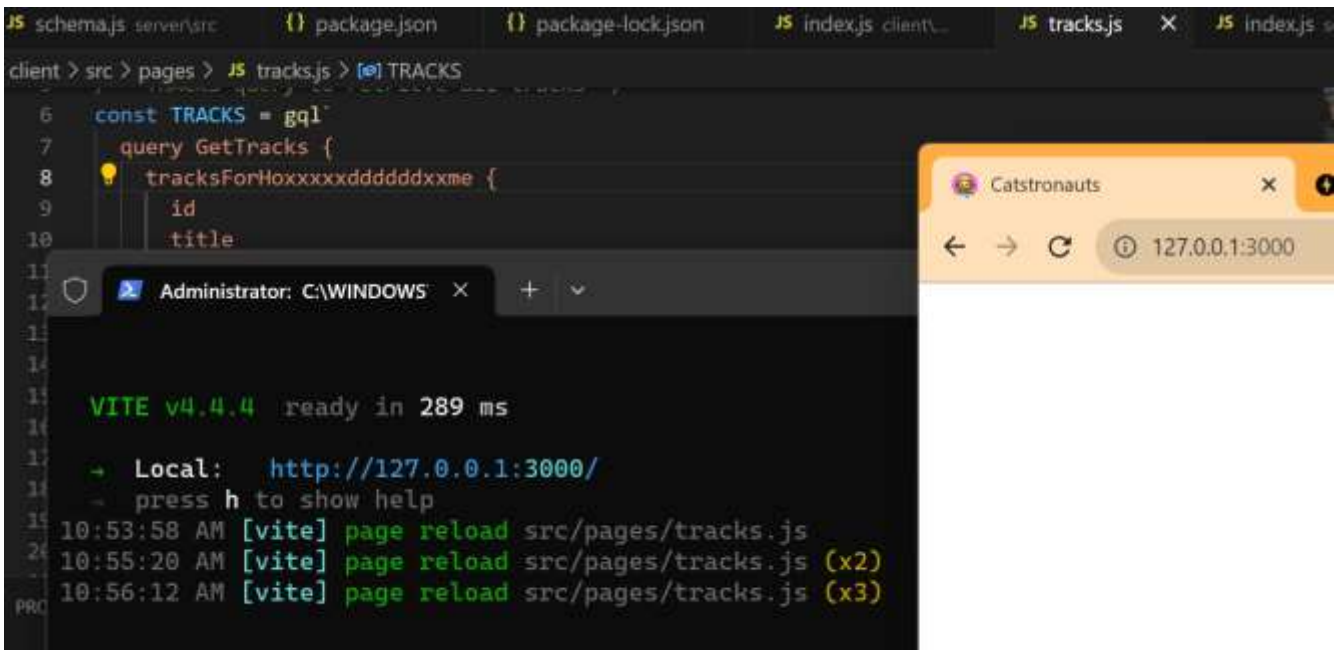
Port 3000 is in use, trying another one...

VITE v4.4.4 ready in 486 ms

➔ Local:   http://127.0.0.1:3001/
➔ press h to show help
```

← → ↻ ⓘ 127.0.0.1:3001

npm start command line .... nothing



The screenshot shows a development environment with VS Code and a web browser. In VS Code, the file `src/pages/tracks.js` is open, showing a GraphQL query `GetTracks` with a variable `tracksForHoxxxxxdddddxxme`. The query fields are `id` and `title`. A web browser window titled "Catstronauts" is open at `127.0.0.1:3000`. In the background, a terminal window shows the Vite development server running on port 3000, with several "page reload" messages for `src/pages/tracks.js`.

```
client > src > pages > JS tracks.js > [⌘] TRACKS
6 const TRACKS = gql`
7   query GetTracks {
8     tracksForHoxxxxxdddddxxme {
9       id
10      title
11    }
12  }
13 `
14
15 VITE v4.4.4 ready in 289 ms
16
17 ➔ Local:   http://127.0.0.1:3000/
18 ➔ press h to show help
19 10:53:58 AM [vite] page reload src/pages/tracks.js
20 10:55:20 AM [vite] page reload src/pages/tracks.js (x2)
21 10:56:12 AM [vite] page reload src/pages/tracks.js (x3)
```

## final file version

```
client > src > pages > JS tracks.js > [0] Tracks
1  import React from "react";
2  import { useQuery, gql } from "@apollo/client";
3  import TrackCard from "../containers/track-card";
4  import { Layout } from "../components";
5
6  /** TRACKS gql query to retrieve all tracks */
7  const TRACKS = gql`
8    query GetTracks {
9      tracksForHome {
10        id
11        title
12        thumbnail
13        length
14        modulesCount
15        author {
16          id
17          name
18          photo
19        }
20      }
21    }
22 `;
23
24 /**
25  * Tracks Page is the Catstronauts home page.
26  * We display a grid of tracks fetched with useQuery with the TRACKS
27  */
28 const Tracks = () => {
29   const { loading, error, data } = useQuery(TRACKS);
30
31   return (
32     <Layout grid>
33       <QueryResult error={error} loading={loading} data={data}>
34         {data?.tracksForHome?.map((track) => (
35           <TrackCard key={track.id} track={track} />
36         ))}
37       </QueryResult>
38     </Layout>
39   );
40 };
41
42 export default Tracks;
```

restart everything... still same error ... have something to do with login error?

The image shows a development environment with three main components:

- Terminal (Top Left):** Displays the process of starting the Apollo Server. The user navigates to `C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server` and runs `npm start`. The output shows the server is running on `http://localhost:4000/`.
- Apollo Server Interface (Top Right):** A web browser window showing the Apollo Server Sandbox at `localhost:4000`. It displays a "Log in to your Studio account" prompt and a "Log in" button. The "Operation" tab shows a query: 

```
query ExampleQuery {
  tracksForView {
    id
  }
}
```
- Terminal (Bottom Left):** Displays the process of starting the Apollo Client. The user navigates to `C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client` and runs `npm start`. The output shows the client is ready in 365 ms and running on `http://127.0.0.1:3000/`.
- Apollo Client Interface (Bottom Right):** A web browser window showing the Apollo Client at `127.0.0.1:3000`. The title is "Catstronauts" and the URL is `127.0.0.1:3000`.

## 11 restart pc

```
C:\WINDOWS\system32\cmd. X + v

Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\terry> cd C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1> cd server
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\server> npm start

> catstronauts-server-complete@1.0.0 start
> nodemon src/index

[nodemon] 2.0.20
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node src/index.js'

🔥 Server is running
🔗 Query at http://localhost:4000/
```

```
C:\WINDOWS\system32\cmd. X + v

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\terry> cd C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1> cd .\client\
PS C:\Users\terry\Downloads\graphql\odyssey-lift-off-part1\client> npm start

> catstronauts-client@1.0.0 start
> vite

VITE v4.4.4 ready in 387 ms

➔ Local:   http://127.0.0.1:3000/
➔ press h to show help
```

Apollo Server

localhost:4000

SEARCH

LOG OUT

SANDBOX

http://localhost:4000/

PUBLISH

LOG IN

ExampleQuery

+

Documentation

Root > Query

Query

Fields

tracksForHome: [Track!]!

Operation

ExampleQuery

```
1 query ExampleQuery {
2   tracksForHome {
3     id
4   }
5 }
6
```

Response

TT

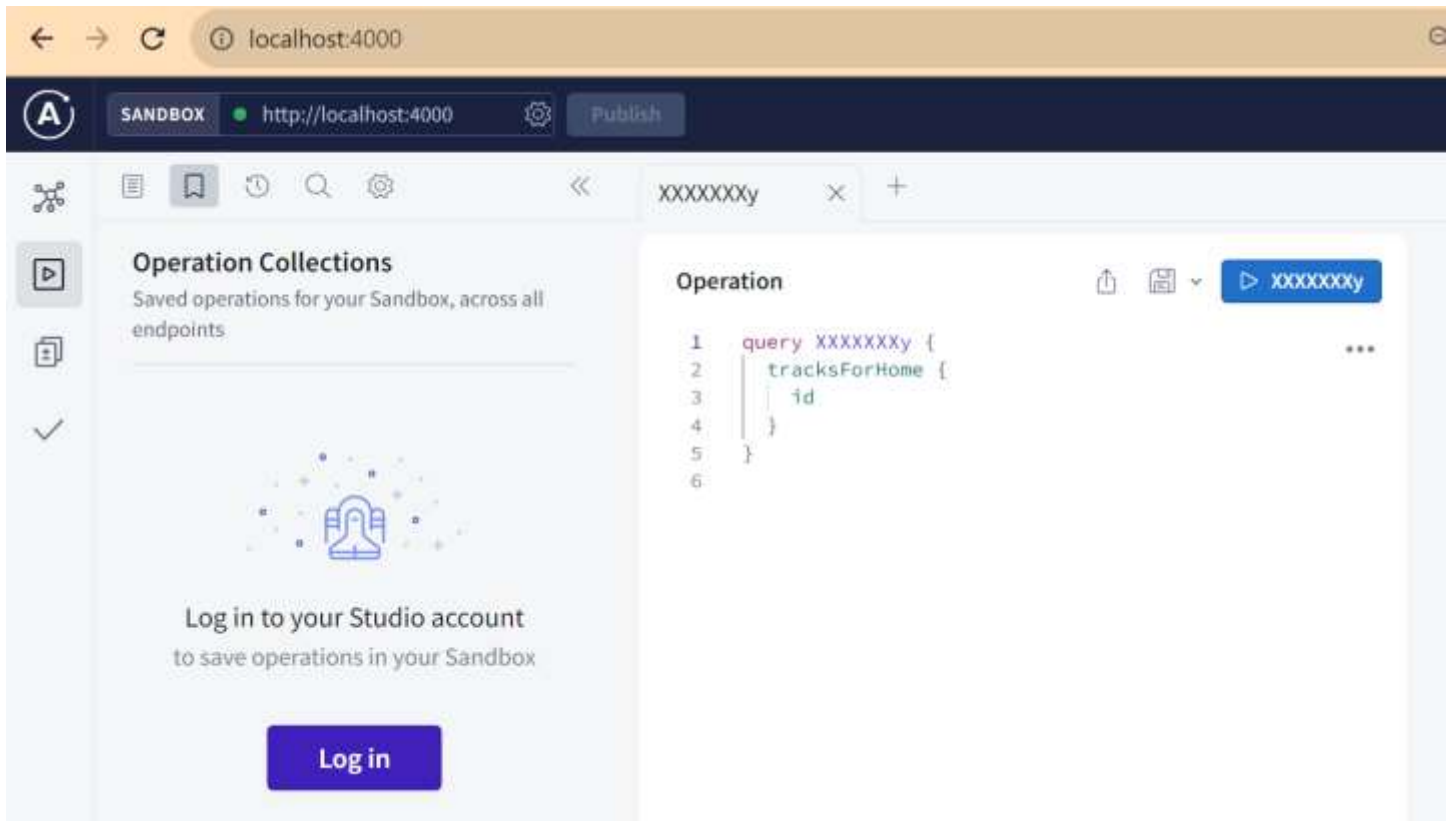
t taylor  
terrytaylorbonn@gmail.com

# Authorize access to your Studio account

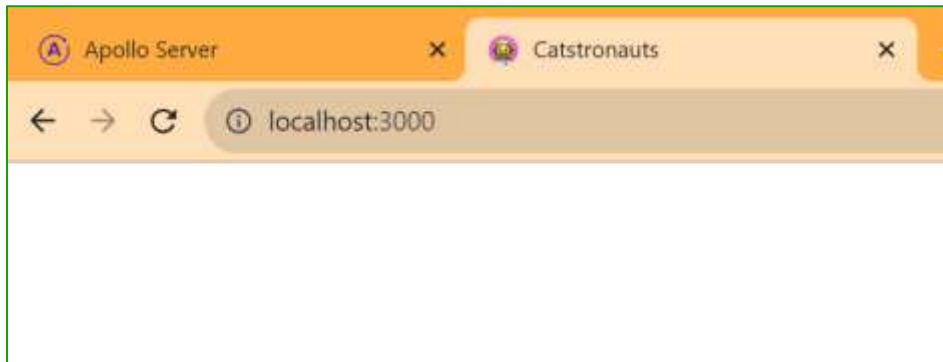
<http://localhost:4000> needs access to your Studio account .

Cancel

Authorize



The login error still there... i cant change the name of the operation.



this not work either

EXPLORER

...stage.json {} package-lock.json JS index.js client\src JS tracks.js X

ODYSSE...  
> .github  
✓ client  
 > node\_modules  
 > public  
 ✓ src  
 > assets  
 > components  
 > containers  
 ✓ pages  
 JS index.js  
 JS tracks.js

client > src > pages > JS tracks.js > [e] TRACKS  
1 import React from "react";  
2 import { useQuery, gql } from "@apollo/client";  
3 import TrackCard from "../containers/track-card";  
4 import { Layout } from "../components";  
5  
6 /\*\* TERRY changed GetTracks back to exampequery  
7 \* TRACKS gql query to retrieve all tracks \*/  
8 const TRACKS = gql`  
9 query ExampleQuery {  
10 tracksForHome {  
11 id  
12 title  
13 thumbnail

