

<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>

The screenshot shows a web browser window with the URL <https://www.apollographql.com/tutorials/lift-off-part1/01-feature-overview-and-setup>. The page title is "Lift-off I: Basics". On the left, there's a sidebar with a navigation menu:

- Feature overview and setup ✓
- Feature data requirements ✓
- Schema definition language (SDL) ✓
- Building our schema ✓
- Apollo Server ✓

The main content area is titled "1. FEATURE OVERVIEW AND SETU". It features a small image of a grumpy cat and the text: "To build this feature, we'll use a "schema" which data our client application needs." Below this, there are three numbered steps:

- 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 t 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 2
Receiving 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#depre-  
cated  
npm WARN deprecated @types/testing-library__dom@7.5.0: This is a stub types defini-  
tion. 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#depre-  
cated  
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**

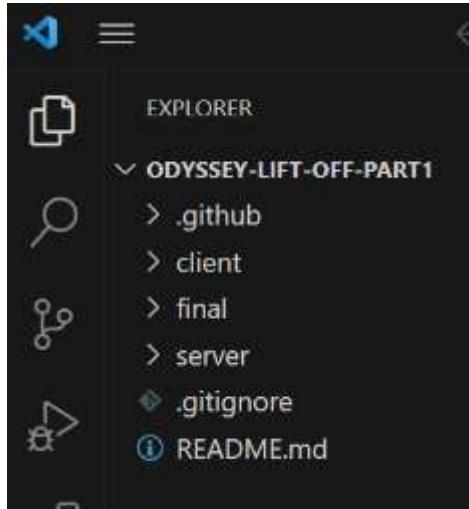
xxx

## **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
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

```
EXPLORER      ...   JS schema.js /init...   JS schema.js server/src X
✓ ODYSSE... [+] ⏪ ⏴ ⏵ ⏶
> .github
> client
> final
> client
> server
> src
  JS index.js
  JS schema.js
  .gitignore
  package-lock.json
  package.json
  README.md
  server
    > node_modules
    > src
      JS index.js
      JS schema.js
      .gitignore
      package-lock.json
      package.json
      1  const gql = require('graphql-tag');
      2
      3  const typeDefs = gql`
```

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

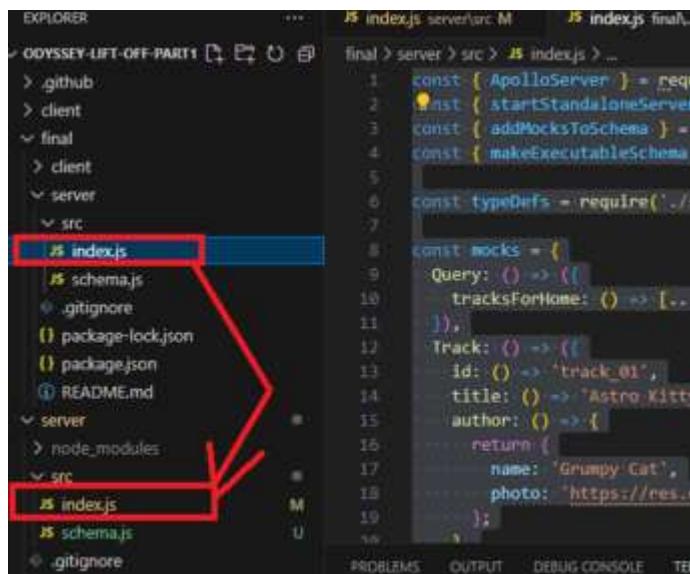
To address all issues (including breaking changes), run:

```
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



```
index.js server/src/M index.js final/M
final > server > src > index.js > ...
1: const { ApolloServer } = require('...
2: const { startStandaloneServer }
3: const { addMocksToSchema }
4: const { makeExecutableSchema }

5: const typeDefs = require('./...
6: const mocks = {
7:   Query: () => ({
8:     tracksForHome: () => [ ...
9:       ...
10:      ]),
11:     Track: () => ({
12:       id: () => 'track_01',
13:       title: () => 'Astro Kitt...
14:       author: () => {
15:         ...
16:         return {
17:           name: 'Grumpy Cat',
18:           photo: 'https://res...
19:         };
20:       };
21:     });
22:   });
23: });

24: const schema = new ApolloServer({
25:   typeDefs,
26:   mocks,
27:   context: () => ({ ...
28:     ...
29:   })
30: });

31: schema.listen({ port: 4001 }, () => {
32:   console.log(`Server is running at http://localhost:4001`);
33: });

34: module.exports = schema;
```

## 5.3 refresh localhost:4000 (explorer)

The screenshot shows the GraphQL Explorer interface at <http://localhost:4000/>. The left sidebar under 'Documentation' shows a tree structure: 'Root' > 'Query' (selected) > 'Fields'. Under 'Fields', 'tracksForHome: [Track!]!' is selected. The main panel displays the 'Operation' code:

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

The 'Response' panel is currently empty.

# 6. using explorer (APOLLO EXPLORER)

## 6.1 example query

The screenshot shows the Apollo Explorer interface with the following sections:

- Fields:** A dropdown menu showing "tracksForHome: [Track!]!" with a checkmark.
- Documentation:** A sidebar with the path "Root > Query > tracksForHome". Below it is a link to "tracksForHome: [Track!]!" with a checkmark, and a note: "Query to get tracks array for the homepage grid".
- Operation:** A code editor containing the GraphQL query:

```
query ExampleQuery {
  tracksForHome {
    id
  }
}
```
- Response:** A JSON preview of the query result:

```
{"data": {"tracksForHome": [{"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 ↓ ↴

- ① author: Author!
- ✓ id: ID!
- ④ length: Int
- ④ modulesCount: Int
- ④ thumbnail: String
- ④ 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 ↓ ↴

① author: Author!  
✓ id: ID!  
④ length: Int  
④ modulesCount: Int  
④ thumbnail: String  
✓ title: String!

Operation

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

Response

```
"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"  
    }  
  ]  
}
```

STATUS 200 38.0ms

## 6.3 select all scalar fields

The screenshot shows a GraphQL playground interface with the following sections:

- Fields**: A dropdown menu with options to "Select all scalar fields" or "Select all fields recursively".
- Documentation**: Information about the `tracksForHome` query.
- Operation**: The `ExampleQuery` with the following GraphQL code:

```
query ExampleQuery {
  tracksForHome {
    id
    title
    thumbnail
    length
    modulesCount
  }
}
```
- Response**: The API response in JSON format, showing three items in the `data.tracksForHome` array. Each item has fields: `id`, `title`, `thumbnail` (with a URL), `length`, and `modulesCount`.

```
{
  "data": {
    "tracksForHome": [
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer",
        "thumbnail": "https://res.cloudinary.com/dety84pbu/image/upload/v1598465568/nebula_cat_djkr9r.jpg",
        "length": 1218,
        "modulesCount": 6
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer",
        "thumbnail": "https://res.cloudinary.com/dety84pbu/image/upload/v1598465568/nebula_cat_djkr9r.jpg",
        "length": 1218,
        "modulesCount": 6
      },
      {
        "id": "track_01",
        "title": "Astro Kitty, Space Explorer",
        "thumbnail": "https://res.cloudinary.com/dety84pbu/image/upload/v1598465568/nebula_cat_djkr9r.jpg",
        "length": 1218,
        "modulesCount": 6
      }
    ]
  }
}
```

## 6.4 select all fields recursively

The screenshot shows the GraphQL playground interface with the following details:

- Fields** dropdown:
  - author: Author!** Select all scalar fields
  - id: ID!** Select all fields recursively
  - length: Int**
- Documentation** sidebar:
  - Root > Query > tracksForHome
  - tracksForHome: [Track!]!** (selected) 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** dropdown:
    - author: Author!**
    - id: ID!**
    - length: Int**
    - modulesCount: Int**
    - thumbnail: String**
    - title: String!**
- Operation** panel:

```
query ExampleQuery {  
  tracksForHome {  
    id  
    title  
    thumbnail  
    length  
    modulesCount  
    author {  
      id  
      name  
      photo  
    }  
  }  
}
```
- Response** panel:

```
data: {  
  tracksForHome: [  
    {  
      id: "track_81",  
      title: "Astro Kitty, Space Explorer",  
      thumbnail: "https://res.cloudinary.com/dety84phu/image/upload/v1598465560/nebula_cat_djkt9r.jpg",  
      length: 1218,  
      modulesCount: 6,  
      author: {  
        id: "1987abaa-599b-49f5-ad88-7dbea58db128",  
        name: "Grumpy Cat",  
        photo: "https://res.cloudinary.com/dety84phu/image/upload/v188616219/kitty-veyron-sw_mtf3c.jpg"  
      }  
    },  
    {  
      id: "track_83",  
      title: "Astro Kitty, Space Explorer",  
      thumbnail: "https://res.cloudinary.com/dety84phu/image/upload/v1598465560/nebula_cat_djkt9r.jpg",  
      length: 1218,  
    }  
  ]  
}
```
- Status**: STATUS 200 21.0ms 2.1KB

## 6.5 rename (select and change)

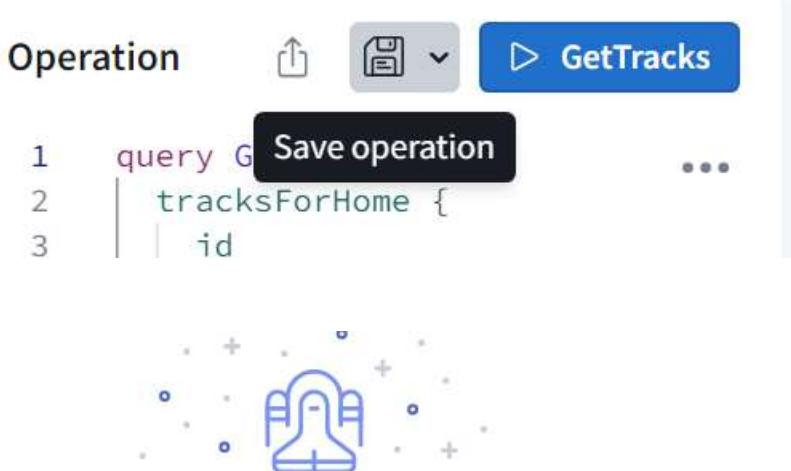
The screenshot shows the GraphQL playground interface with the following details:

- Operation** panel:

GetTracks

```
query GetTracks {  
  tracksForHome {  
    id  
    title  
  }  
}
```

## 6.6 Save operation



## Log in to your Studio account

to save your operation

[Log in](#)



## 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(
```

```
<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 );|
```



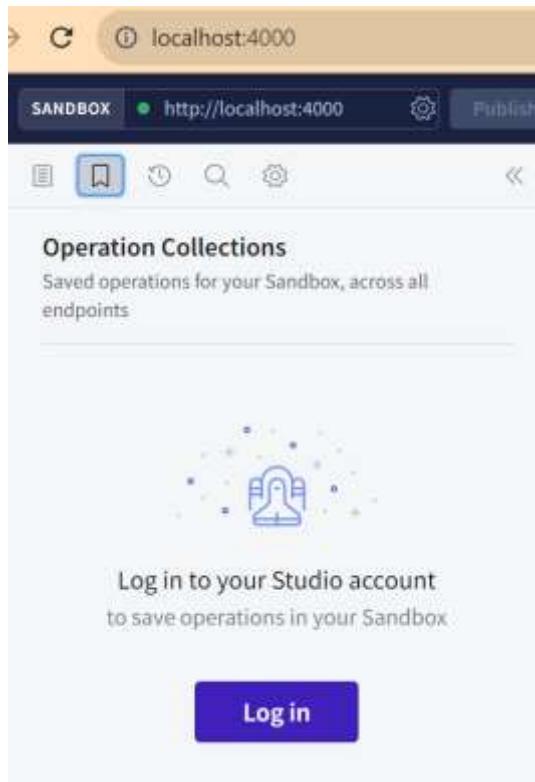
## 9. defining a query

### 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 page
8   * We display a grid of tracks fetched with
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 > [e] 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`query GetTracks {
 7   tracksForHome {
 8     id
 9     title
10     thumbnail
11     length
12     modulesCount
13     author {
14       id
15       name
16       photo
17     }
18   }
19 }
20 `;
21 /**
22  * Tracks Page is the Catstronauts home page.
23  * We display a grid of tracks fetched with useQuery with
24  */
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
```



## npm start command line .... nothing

A screenshot of a terminal window and a browser window. The terminal window shows the command "npm start" being run, followed by the Vite startup message and the local server URL. The browser window shows the URL "127.0.0.1:3000" in the address bar, and its content area is also blank.

## final file version

```
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 /** TRACKS gql query to retrieve all tracks */
  7 const TRACKS = gql`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 /**
 24 * Tracks Page is the Catstronauts home page.
 25 * We display a grid of tracks fetched with useQuery with the TRACKS
 26 */
 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         ))</QueryResult>
 37       </Layout>
 38     );
 39   };
 40
 41
 42 export default Tracks;
```

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

C:\WINDOWS\system32\cmd. x + ~

PS C:\Users\terry> cd ..\Downloads\  
PS C:\Users\terry\Downloads> cd ..\graphql\  
PS C:\Users\terry\Downloads\graphql> ls

Directory: C:\Users\terry\Downloads\graphql

Mode	LastWriteTime	Length	Name
d----	9/15/2023 9:24 AM		odyssey-lift-off-part1

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 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/

Apollo Server - localhost:4000

ExampleQuery

Operation Collections

Operation

```
query ExampleQuery { tracksService { id } }
```

Login to your Studio account to save operations in your Sandbox.

Login Variables Headers Script

1 Add Files

C:\WINDOWS\system32\cmd. x + ~

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 365 ms

- Local: http://127.0.0.1:3000/  
- press h to show help

Catstronauts - 127.0.0.1:3000

## 11 restart pc

```
C:\WINDOWS\system32\cmd. × + ▾ 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. × + ▾ 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
```

The screenshot shows the Apollo Server interface in a browser window. The title bar says "Apollo Server" and the address bar says "localhost:4000". The main area is titled "ExampleQuery". On the left, there's a "Documentation" sidebar with a tree view: Root > Query > Query. Under "Fields", "tracksForHome: [Track!]!" is selected. The central panel shows the GraphQL schema:

```
query ExampleQuery {
  tracksForHome {
    id
  }
}
```

A dark blue modal dialog box is centered on the screen. It contains a circular profile picture with initials "TT", the name "t taylor", and the email "terrytaylorbonn@gmail.com". The main text reads "Authorize access to your Studio account". Below this, it says "http://localhost:4000 needs access to your Studio account." At the bottom are two buttons: "Cancel" (white background) and "Authorize" (dark blue background).

The screenshot shows the Apollo Studio interface at localhost:4000. The top navigation bar includes a logo, 'Sandbox' tab, URL, and 'Publish' button. Below the header are standard browser controls (back, forward, search). A sidebar on the left titled 'Operation Collections' displays a single operation named 'xxxxxxxy'. The main panel shows the operation code:

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

A message below the code states: 'Log in to your Studio account to save operations in your Sandbox' with a 'Log in' button. The overall interface is light blue and white.

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

The screenshot shows the Apollo Server interface at localhost:3000. The top navigation bar includes a logo, 'Apollo Server' tab, URL, and a 'Catstronauts' tab. Below the header are standard browser controls. The main content area is completely blank, indicating no data or errors.

this not work either

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

✓ ODYSSE... D+ E+ ⌂ ⌂ client > src > pages > JS tracks.js > [o] TRACKS

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

```
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 examplequery
7 * TRACKS gql query to retrieve all tracks */
8 const TRACKS = gql` 
9   query ExampleQuery {
10     tracksForHome {
11       id
12       title
13       thumbnail
14     }
15   }
16 
```

