

Connect React Django Tutorial



how to connect react and fjango



To connect a React frontend with a Django backend, you can follow these steps:

Step 1: Set Up the Django Backend

1. Install Django and Django REST Framework:

```
bash

pip install django djangorestframework
```

2. Create a Django Project:

```
bash

django-admin startproject myproject
cd myproject
```

3. Create a Django App:

```
bash

python manage.py startapp myapp
```

4. Add `rest_framework` and your app to `INSTALLED_APPS` in `myproject/settings.py`:

```
python

INSTALLED_APPS = [
    ...
    'rest_framework',
    'myapp',
]
```

5. Set up Django REST Framework:

Create a `serializers.py` in your app directory for your models:

```
python
```

```
from rest_framework import serializers
from .models import MyModel

class MyModelSerializer(serializers.ModelSerializer):
    class Meta:
        model = MyModel
        fields = '__all__'
```

Create a `views.py` for your API endpoints:

```
python
```

```
from rest_framework import viewsets
from .models import MyModel
from .serializers import MyModelSerializer

class MyModelViewSet(viewsets.ModelViewSet):
    queryset = MyModel.objects.all()
    serializer_class = MyModelSerializer
```

Add the URLs in `myapp/urls.py`:

```
python
```

```
from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import MyModelViewSet

router = DefaultRouter()
router.register(r'mymodels', MyModelViewSet)

urlpatterns = [
    path('', include(router.urls)),
]
```

Include your app URLs in `myproject/urls.py`:

```
python
```

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('myapp.urls')),
]
```

6. Run Migrations and Start the Server:

```
bash
```

```
python manage.py makemigrations
python manage.py migrate
```

```
python manage.py runserver
```

Step 2: Set Up the React Frontend

1. Create a React App:

```
bash
```

```
npx create-react-app myfrontend
cd myfrontend
```

2. Install Axios for Making HTTP Requests:

```
bash
```

```
npm install axios
```

3. Create a Service to Handle API Requests:

Create a file `src/api.js`:

```
javascript
```

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

4. Fetch Data in a React Component:

Edit `src/App.js` to fetch data from the Django API:

```
javascript
```

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, []);

  return (
    <div className="App">
      <h1>My Models</h1>
      <ul>
```

```

        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;

```

5. Run the React App:

```
bash
```

```
npm start
```

Step 3: Handle CORS (Cross-Origin Resource Sharing)

1. Install `django-cors-headers` in Django:

```
bash
```

```
pip install django-cors-headers
```

2. Add `corsheaders` to `INSTALLED_APPS` and configure middleware in `myproject/settings.py`:

```
python
```

```

INSTALLED_APPS = [
    ...
    'corsheaders',
]

MIDDLEWARE = [
    ...
    'corsheaders.middleware.CorsMiddleware',
    ...
]

CORS_ALLOW_ALL_ORIGINS = True # Note: For production, set this more restrictively

```

Summary

- **Django** serves as the backend API using Django REST Framework.
- **React** is used as the frontend to fetch and display data from the Django backend.
- **Axios** is used in React to make HTTP requests to the Django API.
- **CORS** is configured to allow requests from the React frontend to the Django backend.

With this setup, you can connect a React frontend to a Django backend and create a full-stack application.



in Step 2: Set Up the React Frontend , provide the file location to create react app



To set up the React frontend, you should create the React app in a directory parallel to your Django project. Here are the detailed steps, including the file locations:

Step 2: Set Up the React Frontend

1. Navigate to the Parent Directory of Your Django Project:

Assuming your Django project (`myproject`) is located in `~/projects/myproject`, navigate to the parent directory:

```
bash

cd ~/projects
```

2. Create a React App:

Create a new React app named `myfrontend` in the same directory where your Django project is located:

```
bash

npx create-react-app myfrontend
```

This will create a new directory structure as follows:

```
javascript

~/projects/
├── myproject/
└── myfrontend/
```

3. Navigate to the React App Directory:

```
bash

cd myfrontend
```

4. Install Axios for Making HTTP Requests:

```
bash
```

```
npm install axios
```

5. Create a Service to Handle API Requests:

Create a file named `api.js` in the `src` directory:

```
javascript
```

```
// File location: ~/projects/myfrontend/src/api.js

import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

6. Fetch Data in a React Component:

Edit the `src/App.js` file to fetch data from the Django API:

```
javascript
```

```
// File location: ~/projects/myfrontend/src/App.js

import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, []);

  return (
    <div className="App">
      <h1>My Models</h1>
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

7. Run the React App:

```
bash
```

```
npm start
```

With these steps, you have set up a React frontend in a directory parallel to your Django backend and configured it to communicate with the Django API. The directory structure will look like this:

```
bash
```

```
~/projects/
├── myproject/ # Django project
└── myfrontend/ # React app
```



inside the django project file or outside



You should create the React app outside of the Django project directory. This separation keeps the frontend and backend concerns isolated, making it easier to manage and deploy both parts of your application. Here is a more detailed step-by-step guide:

Step 2: Set Up the React Frontend

1. Navigate to the Parent Directory of Your Django Project:

Assuming your Django project (`myproject`) is located in `~/projects/myproject`, navigate to the parent directory:

```
bash
```

```
cd ~/projects
```

2. Create a React App:

Create a new React app named `myfrontend` in the same directory where your Django project is located:

```
bash
```

```
npx create-react-app myfrontend
```

This will create a new directory structure as follows:

```
bash
```

```
~/projects/
├── myproject/ # Django project directory
└── myfrontend/ # React app directory
```

3. Navigate to the React App Directory:

```
bash
```

```
cd myfrontend
```

4. Install Axios for Making HTTP Requests:

```
bash
```

```
npm install axios
```

5. Create a Service to Handle API Requests:

Create a file named `api.js` in the `src` directory:

```
javascript
```

```
// File location: ~/projects/myfrontend/src/api.js

import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

6. Fetch Data in a React Component:

Edit the `src/App.js` file to fetch data from the Django API:

```
javascript
```

```
// File location: ~/projects/myfrontend/src/App.js

import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, []);
```



```
return (
  <div className="App">
    <h1>My Models</h1>
    <ul>
      {data.map((item) => (
        <li key={item.id}>{item.name}</li>
      ))}
    </ul>
  </div>
);
}

export default App;
```

7. Run the React App:

```
bash
```

```
npm start
```

Final Directory Structure

```
ruby
```

```
~/projects/
├── myproject/ # Django project directory
│   ├── myproject/
│   └── myapp/
├── myfrontend/ # React app directory
│   ├── src/
│   │   ├── App.js
│   │   └── api.js
│   └── public/
```

By keeping the React app (`myfrontend`) and the Django project (`myproject`) in separate directories within the same parent directory, you maintain a clean separation of concerns, making it easier to manage both parts of your application.



```
npm install axios
```

```
added 3 packages, and audited 1548 packages in 7s
```

```
262 packages are looking for funding
```

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

```
8 vulnerabilities (2 moderate, 6 high)
```

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

```
npm audit fix --force
```

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



The output you received indicates that some vulnerabilities were found in your project's dependencies. You can address these vulnerabilities by running the suggested commands. Here's what you should do:

Step 1: Run `npm audit`

This command will give you a detailed report of the vulnerabilities found in your project.

```
bash
```

```
npm audit
```

Step 2: Fix Vulnerabilities

You can try to fix the vulnerabilities by running:

```
bash
```

```
npm audit fix
```

If there are still vulnerabilities left after running the above command, you might need to force the fix, which may include breaking changes:

```
bash
```

```
npm audit fix --force
```

Example of the Output and Fix Commands

Here's a typical sequence of commands you might run:

1. Check for vulnerabilities:

```
bash
```

```
npm audit
```

2. Fix vulnerabilities automatically:

```
bash
```

```
npm audit fix
```

3. Force fix remaining vulnerabilities (if any):

```
bash
```

```
npm audit fix --force
```

Additional Steps if Fix Commands Don't Work

Sometimes, `npm audit fix` commands might not resolve all issues, especially if they involve breaking changes or dependencies that are no longer maintained. In such cases, consider the following steps:

1. Update Dependencies Manually:

Check the `package.json` file and update the versions of the packages that have vulnerabilities. You can find the latest versions of the packages on npmjs.com.

2. Check for Deprecated Packages:

Replace deprecated or unmaintained packages with actively maintained alternatives.

3. Test Your Application:

After updating dependencies, thoroughly test your application to ensure that everything works correctly with the new package versions.

Final Note

While addressing vulnerabilities is important, be cautious with force fixes as they can introduce breaking changes to your project. Always test your application after making such changes to ensure it still functions as expected.



```
npm audit fix --force
npm WARN using --force Recommended protections disabled.
npm WARN audit Updating react-scripts to 3.0.1, which is a SemVer major change.
npm WARN deprecated domexception@1.0.1: Use your platform's native DOMException instead
npm WARN deprecated @babel/plugin-proposal-unicode-property-regex@7.18.6: This proposal
has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please
use @babel/plugin-transform-unicode-property-regex instead.
npm WARN deprecated @babel/plugin-proposal-json-strings@7.18.6: This proposal has been
merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use
@babel/plugin-transform-json-strings instead.
npm WARN deprecated @babel/plugin-proposal-async-generator-functions@7.20.7: This proposal
has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please
use @babel/plugin-transform-async-generator-functions instead.
npm WARN deprecated @babel/plugin-proposal-optional-catch-binding@7.18.6: This proposal has
been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use
@babel/plugin-transform-optional-catch-binding instead.
npm WARN deprecated @babel/plugin-proposal-dynamic-import@7.18.6: This proposal has been
merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use
@babel/plugin-transform-dynamic-import instead.
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 resolve-url@0.2.1: <https://github.com/lydell/resolve-url#deprecated>

npm WARN deprecated @babel/plugin-proposal-object-rest-spread@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-object-rest-spread instead.

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

npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained

npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'

npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained

npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained

npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'

npm WARN deprecated acorn-dynamic-import@4.0.0: This is probably built in to whatever tool you're using. If you still need it... idk

npm WARN deprecated fs-write-stream-atomic@1.0.10: This package is no longer supported.

npm WARN deprecated copy-concurrently@1.0.5: This package is no longer supported.

npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.

npm WARN deprecated move-concurrently@1.0.1: This package is no longer supported.

npm WARN deprecated figgy-pudding@3.5.2: This module is no longer supported.

npm WARN deprecated har-validator@5.1.5: this library is no longer supported

npm WARN deprecated flatten@1.0.3: flatten is deprecated in favor of utility frameworks such as lodash.

npm WARN deprecated request-promise-native@1.0.9: request-promise-native has been deprecated because it extends the now deprecated request package, see <https://github.com/request/request/issues/3142>

npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScript syntax/features added

npm WARN deprecated request@2.88.2: request has been deprecated, see <https://github.com/request/request/issues/3142>

npm WARN deprecated left-pad@1.3.0: use String.prototype.padStart()

npm WARN deprecated eslint-loader@2.1.2: This loader has been deprecated. Please use eslint-webpack-plugin

npm WARN deprecated babel-eslint@10.0.1: babel-eslint is now @babel/eslint-parser. This package will no longer receive updates.

npm WARN deprecated workbox-google-analytics@4.3.1: It is not compatible with newer versions of GA starting with v4, as long as you are using GAv3 it should be ok, but the package is no longer being maintained

npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See <https://v8.dev/blog/math-random> for details.

npm WARN deprecated rimraf@2.7.1: Rimraf versions prior to v4 are no longer supported

npm WARN deprecated html-webpack-plugin@4.0.0-beta.5: please switch to a stable version

npm WARN deprecated rimraf@2.6.3: Rimraf versions prior to v4 are no longer supported

npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15x fewer dependencies

npm WARN deprecated @babel/plugin-proposal-nullish-coalescing-operator@7.8.3: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-nullish-coalescing-operator instead.

npm WARN deprecated @babel/plugin-proposal-numeric-separator@7.8.3: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-numeric-separator instead.

npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.9.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.

npm WARN deprecated @babel/plugin-proposal-class-properties@7.8.3: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-class-properties instead.

added 761 packages, removed 551 packages, changed 375 packages, and audited 1758 packages in 4m

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

npm audit report

ansi-html <0.0.8

Severity: high

Uncontrolled Resource Consumption in ansi-html - <https://github.com/advisories/GHSA-whgm-jr23-g3j9>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/ansi-html

webpack-dev-server <=4.7.4

Depends on vulnerable versions of ansi-html

Depends on vulnerable versions of bonjour

Depends on vulnerable versions of chokidar

Depends on vulnerable versions of http-proxy-middleware

Depends on vulnerable versions of ip

Depends on vulnerable versions of selfsigned

Depends on vulnerable versions of sockjs

Depends on vulnerable versions of webpack

Depends on vulnerable versions of webpack-dev-middleware

Depends on vulnerable versions of yargs

node_modules/webpack-dev-server

react-scripts >=0.1.0

Depends on vulnerable versions of @svgr/webpack

Depends on vulnerable versions of @typescript-eslint/eslint-plugin

Depends on vulnerable versions of @typescript-eslint/parser

Depends on vulnerable versions of babel-jest

Depends on vulnerable versions of css-loader

Depends on vulnerable versions of eslint-config-react-app

Depends on vulnerable versions of file-loader

Depends on vulnerable versions of html-webpack-plugin

Depends on vulnerable versions of jest
 Depends on vulnerable versions of jest-environment-jsdom-fourteen
 Depends on vulnerable versions of jest-watch-typeahead
 Depends on vulnerable versions of mini-css-extract-plugin
 Depends on vulnerable versions of optimize-css-assets-webpack-plugin
 Depends on vulnerable versions of postcss-flexbugs-fixes
 Depends on vulnerable versions of postcss-loader
 Depends on vulnerable versions of postcss-normalize
 Depends on vulnerable versions of postcss-preset-env
 Depends on vulnerable versions of postcss-safe-parser
 Depends on vulnerable versions of react-dev-utils
 Depends on vulnerable versions of semver
 Depends on vulnerable versions of terser-webpack-plugin
 Depends on vulnerable versions of webpack
 Depends on vulnerable versions of webpack-dev-server
 Depends on vulnerable versions of workbox-webpack-plugin
 node_modules/react-scripts

braces <3.0.3

Severity: high

Uncontrolled resource consumption in braces - <https://github.com/advisories/GHSA-grv7-fg5c-xmjb>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/@jest/core/node_modules/braces

node_modules/@jest/transform/node_modules/braces

node_modules/anymatch/node_modules/braces

node_modules/chokidar/node_modules/braces

node_modules/fast-glob/node_modules/braces

node_modules/fork-ts-checker-webpack-plugin/node_modules/braces

node_modules/http-proxy-middleware/node_modules/braces

node_modules/jest-config/node_modules/braces

node_modules/jest-haste-map/node_modules/braces

node_modules/jest-message-util/node_modules/braces

node_modules/readdirp/node_modules/braces

node_modules/sane/node_modules/braces

node_modules/webpack/node_modules/braces

chokidar 1.3.0 - 2.1.8

Depends on vulnerable versions of anymatch

Depends on vulnerable versions of braces

Depends on vulnerable versions of readdirp

node_modules/chokidar

fork-ts-checker-webpack-plugin 0.4.7 - 4.1.6

Depends on vulnerable versions of chokidar

Depends on vulnerable versions of micromatch

node_modules/fork-ts-checker-webpack-plugin

react-dev-utils 0.4.0 - 12.0.0-next.60

Depends on vulnerable versions of browserslist

Depends on vulnerable versions of fork-ts-checker-webpack-plugin

Depends on vulnerable versions of globby

Depends on vulnerable versions of loader-utils

Depends on vulnerable versions of recursive-readdir
 Depends on vulnerable versions of shell-quote
 node_modules/react-dev-utils
 watchpack-chokidar2 *
 Depends on vulnerable versions of chokidar
 node_modules/watchpack-chokidar2
 watchpack 1.7.2 - 1.7.5
 Depends on vulnerable versions of watchpack-chokidar2
 node_modules/watchpack
 micromatch 0.2.0 - 3.1.10
 Depends on vulnerable versions of braces
 node_modules/@jest/core/node_modules/micromatch
 node_modules/@jest/transform/node_modules/micromatch
 node_modules/anymatch/node_modules/micromatch
 node_modules/fast-glob/node_modules/micromatch
 node_modules/fork-ts-checker-webpack-plugin/node_modules/micromatch
 node_modules/http-proxy-middleware/node_modules/micromatch
 node_modules/jest-config/node_modules/micromatch
 node_modules/jest-haste-map/node_modules/micromatch
 node_modules/jest-message-util/node_modules/micromatch
 node_modules/readdirp/node_modules/micromatch
 node_modules/sane/node_modules/micromatch
 node_modules/webpack/node_modules/micromatch
 @jest/core <=25.5.4
 Depends on vulnerable versions of @jest/reporters
 Depends on vulnerable versions of @jest/transform
 Depends on vulnerable versions of jest-config
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-message-util
 Depends on vulnerable versions of jest-resolve-dependencies
 Depends on vulnerable versions of jest-runner
 Depends on vulnerable versions of jest-runtime
 Depends on vulnerable versions of jest-snapshot
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of jest-watcher
 Depends on vulnerable versions of micromatch
 node_modules/@jest/core
 jest-cli 23.5.0 - 24.9.0
 Depends on vulnerable versions of @jest/core
 Depends on vulnerable versions of jest-config
 Depends on vulnerable versions of jest-util
 node_modules/jest-cli
 jest 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of jest-cli
 node_modules/jest
 @jest/transform <=24.9.0
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of micromatch
 node_modules/@jest/transform
 @jest/environment <=24.9.0

Depends on vulnerable versions of @jest/fake-timers
 Depends on vulnerable versions of @jest/transform
 node_modules/@jest/environment
 jest-environment-jsdom 10.0.2 - 25.5.0
 Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of @jest/fake-timers
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of jsdom
 node_modules/jest-environment-jsdom
 jest-config 12.1.1-alpha.2935e14d - 25.5.4
 Depends on vulnerable versions of @jest/test-sequencer
 Depends on vulnerable versions of babel-jest
 Depends on vulnerable versions of jest-environment-jsdom
 Depends on vulnerable versions of jest-environment-node
 Depends on vulnerable versions of jest-jasmine2
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of micromatch
 node_modules/jest-config
 jest-runner 21.0.0-alpha.1 - 24.9.0
 Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of jest-config
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-jasmine2
 Depends on vulnerable versions of jest-message-util
 Depends on vulnerable versions of jest-runtime
 Depends on vulnerable versions of jest-util
 node_modules/jest-runner
 jest-runtime 18.1.0 - 24.9.0
 Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of @jest/transform
 Depends on vulnerable versions of jest-config
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-message-util
 Depends on vulnerable versions of jest-snapshot
 Depends on vulnerable versions of jest-util
 node_modules/jest-runtime
 jest-environment-node 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of @jest/fake-timers
 Depends on vulnerable versions of jest-util
 node_modules/jest-environment-node
 jest-jasmine2 18.5.0-alpha.7da3df39 - 24.9.0
 Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of expect
 Depends on vulnerable versions of jest-each
 Depends on vulnerable versions of jest-message-util
 Depends on vulnerable versions of jest-runtime
 Depends on vulnerable versions of jest-snapshot
 Depends on vulnerable versions of jest-util
 node_modules/jest-jasmine2
 @jest/reporters <=26.4.0

Depends on vulnerable versions of @jest/environment
 Depends on vulnerable versions of @jest/transform
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-runtime
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of node-notifier
 node_modules/@jest/reporters
 babel-jest 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of @jest/transform
 node_modules/babel-jest
 anymatch 1.2.0 - 2.0.0
 Depends on vulnerable versions of micromatch
 node_modules/anymatch
 jest-haste-map 18.1.0 - 26.6.2
 Depends on vulnerable versions of anymatch
 Depends on vulnerable versions of jest-util
 Depends on vulnerable versions of micromatch
 Depends on vulnerable versions of sane
 node_modules/jest-haste-map
 @jest/test-sequencer <=24.9.0
 Depends on vulnerable versions of jest-haste-map
 Depends on vulnerable versions of jest-runner
 Depends on vulnerable versions of jest-runtime
 node_modules/@jest/test-sequencer
 sane 1.5.0 - 4.1.0
 Depends on vulnerable versions of anymatch
 Depends on vulnerable versions of micromatch
 node_modules/sane
 fast-glob <=2.2.7
 Depends on vulnerable versions of micromatch
 node_modules/fast-glob
 globby 8.0.0 - 9.2.0
 Depends on vulnerable versions of fast-glob
 node_modules/globby
 http-proxy-middleware 0.3.0 - 0.19.2
 Depends on vulnerable versions of micromatch
 node_modules/http-proxy-middleware
 jest-message-util 18.5.0-alpha.7da3df39 - 24.9.0
 Depends on vulnerable versions of micromatch
 node_modules/jest-message-util
 @jest/fake-timers <=24.9.0
 Depends on vulnerable versions of jest-message-util
 node_modules/@jest/fake-timers
 jest-util 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of @jest/fake-timers
 node_modules/jest-util
 jest-each 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of jest-util
 node_modules/jest-each
 jest-environment-jsdom-fourteen *
 Depends on vulnerable versions of jest-util

Depends on vulnerable versions of jsdom
 node_modules/jest-environment-jsdom-fourteen
 jest-watcher 24.2.0-alpha.0 - 24.9.0
 Depends on vulnerable versions of jest-util
 node_modules/jest-watcher
 jest-watch-typeahead 0.2.2-0 - 0.4.2
 Depends on vulnerable versions of jest-watcher
 node_modules/jest-watch-typeahead
 expect 21.0.0-beta.1 - 24.9.0
 Depends on vulnerable versions of jest-message-util
 node_modules/expect
 jest-snapshot 23.3.0 - 24.9.0
 Depends on vulnerable versions of expect
 Depends on vulnerable versions of jest-message-util
 node_modules/jest-snapshot
 jest-resolve-dependencies 23.3.0 - 24.9.0
 Depends on vulnerable versions of jest-snapshot
 node_modules/jest-resolve-dependencies
 readdirp 2.2.0 - 2.2.1
 Depends on vulnerable versions of micromatch
 node_modules/readdirp
 webpack 4.0.0-alpha.0 - 5.0.0-rc.6
 Depends on vulnerable versions of micromatch
 node_modules/webpack
 css-loader 0.15.0 - 4.3.0
 Depends on vulnerable versions of icss-utils
 Depends on vulnerable versions of postcss
 Depends on vulnerable versions of postcss-modules-extract-imports
 Depends on vulnerable versions of postcss-modules-local-by-default
 Depends on vulnerable versions of postcss-modules-scope
 Depends on vulnerable versions of postcss-modules-values
 Depends on vulnerable versions of webpack
 node_modules/css-loader
 file-loader 3.0.0 - 4.3.0
 Depends on vulnerable versions of webpack
 node_modules/file-loader
 html-webpack-plugin 1.4.0 - 4.0.4
 Depends on vulnerable versions of html-minifier
 Depends on vulnerable versions of webpack
 node_modules/html-webpack-plugin
 mini-css-extract-plugin <=0.9.0
 Depends on vulnerable versions of webpack
 node_modules/mini-css-extract-plugin
 optimize-css-assets-webpack-plugin <=1.3.2 || >=3.0.0
 Depends on vulnerable versions of cssnano
 Depends on vulnerable versions of webpack
 node_modules/optimize-css-assets-webpack-plugin
 terser-webpack-plugin <=2.2.1
 Depends on vulnerable versions of serialize-javascript
 Depends on vulnerable versions of terser
 Depends on vulnerable versions of webpack

node_modules/terser-webpack-plugin

browserslist 4.0.0 - 4.16.4

Severity: moderate

Regular Expression Denial of Service in browserslist - <https://github.com/advisories/GHSA-w8qv-6jwh-64r5>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/react-dev-utils/node_modules/browserslist

html-minifier *

Severity: high

kangax html-minifier ReDoS vulnerability - <https://github.com/advisories/GHSA-pfq8-rq6v-vf5m>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/html-minifier

ip *

Severity: high

ip SSRF improper categorization in isPublic - <https://github.com/advisories/GHSA-2p57-rm9w-gvfp>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/ip

dns-packet <=5.2.4

Depends on vulnerable versions of ip

node_modules/dns-packet

multicast-dns 6.0.0 - 7.2.2

Depends on vulnerable versions of dns-packet

node_modules/multicast-dns

bonjour >=3.3.1

Depends on vulnerable versions of multicast-dns

node_modules/bonjour

jsdom <=16.5.3

Severity: moderate

Insufficient Granularity of Access Control in JSDom - <https://github.com/advisories/GHSA-f4c9-cqv8-9v98>

Depends on vulnerable versions of request

Depends on vulnerable versions of request-promise-native

Depends on vulnerable versions of tough-cookie

fix available via `npm audit fix`

node_modules/jest-environment-jsdom-fourteen/node_modules/jsdom

node_modules/jsdom

loader-utils <=1.4.1

Severity: critical

Prototype pollution in webpack loader-utils - <https://github.com/advisories/GHSA-76p3-8jx3-jpfq>

loader-utils is vulnerable to Regular Expression Denial of Service (ReDoS) via url variable -

<https://github.com/advisories/GHSA-3rfm-jhwj-7488>

loader-utils is vulnerable to Regular Expression Denial of Service (ReDoS) -

<https://github.com/advisories/GHSA-hhq3-ff78-jv3g>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change
node_modules/react-dev-utils/node_modules/loader-utils

lodash.template *

Severity: high

Command Injection in lodash - <https://github.com/advisories/GHSA-35jh-r3h4-6jhm>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/lodash.template

workbox-build <=5.1.4

Depends on vulnerable versions of lodash.template

node_modules/workbox-build

workbox-webpack-plugin <=5.1.4

Depends on vulnerable versions of workbox-build

node_modules/workbox-webpack-plugin

minimatch <3.0.5

Severity: high

minimatch ReDoS vulnerability - <https://github.com/advisories/GHSA-f8q6-p94x-37v3>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/recursive-readdir/node_modules/minimatch

recursive-readdir 1.2.0 - 2.2.2

Depends on vulnerable versions of minimatch

node_modules/recursive-readdir

node-forge <=1.2.1

Severity: high

Prototype Pollution in node-forge debug API. - <https://github.com/advisories/GHSA-5rrq-pxf6-6jx5>

URL parsing in node-forge could lead to undesired behavior. -

<https://github.com/advisories/GHSA-gf8q-jrpm-jvxq>

Improper Verification of Cryptographic Signature in `node-forge` -

<https://github.com/advisories/GHSA-2r2c-g63r-vccr>

Open Redirect in node-forge - <https://github.com/advisories/GHSA-8fr3-hfg3-gpgp>

Improper Verification of Cryptographic Signature in node-forge -

<https://github.com/advisories/GHSA-cfm4-qjh2-4765>

Improper Verification of Cryptographic Signature in node-forge -

<https://github.com/advisories/GHSA-x4jg-mjrx-434g>

fix available via `npm audit fix --force`

Will install react-scripts@5.0.1, which is a breaking change

node_modules/node-forge

selfsigned 1.1.1 - 1.10.14

Depends on vulnerable versions of node-forge

node_modules/selfsigned

node-notifier <8.0.1

Severity: moderate

OS Command Injection in node-notifier - <https://github.com/advisories/GHSA-5fw9-fq32-ww5p>

fix available via `npm audit fix`

node_modules/node-notifier

nth-check <2.0.1
Severity: high
Inefficient Regular Expression Complexity in nth-check - <https://github.com/advisories/GHSA-rp65-9cf3-cjxr>
fix available via `npm audit fix --force`
Will install react-scripts@5.0.1, which is a breaking change
node_modules/svgo/node_modules/nth-check
css-select <=3.1.0
Depends on vulnerable versions of nth-check
node_modules/svgo/node_modules/css-select
svgo 1.0.0 - 1.3.2
Depends on vulnerable versions of css-select
node_modules/svgo
@svgr/plugin-svgo <=5.5.0
Depends on vulnerable versions of svgo
node_modules/@svgr/plugin-svgo
@svgr/webpack 4.0.0 - 5.5.0
Depends on vulnerable versions of @svgr/plugin-svgo
node_modules/@svgr/webpack
postcss-svgo <=5.0.0-rc.2
Depends on vulnerable versions of postcss
Depends on vulnerable versions of svgo
node_modules/postcss-svgo

postcss <8.4.31
Severity: moderate
PostCSS line return parsing error - <https://github.com/advisories/GHSA-7fh5-64p2-3v2j>
fix available via `npm audit fix --force`
Will install react-scripts@5.0.1, which is a breaking change
node_modules/postcss
autoprefixer 1.0.20131222 - 9.8.8
Depends on vulnerable versions of postcss
node_modules/autoprefixer
postcss-preset-env <=7.0.0
Depends on vulnerable versions of autoprefixer
Depends on vulnerable versions of css-blank-pseudo
Depends on vulnerable versions of css-has-pseudo
Depends on vulnerable versions of css-prefers-color-scheme
Depends on vulnerable versions of postcss
Depends on vulnerable versions of postcss-attribute-case-insensitive
Depends on vulnerable versions of postcss-color-functional-notation
Depends on vulnerable versions of postcss-color-gray
Depends on vulnerable versions of postcss-color-hex-alpha
Depends on vulnerable versions of postcss-color-mod-function
Depends on vulnerable versions of postcss-color-rebeccapurple
Depends on vulnerable versions of postcss-custom-media
Depends on vulnerable versions of postcss-custom-properties
Depends on vulnerable versions of postcss-custom-selectors
Depends on vulnerable versions of postcss-dir-pseudo-class
Depends on vulnerable versions of postcss-double-position-gradients

Depends on vulnerable versions of postcss-env-function
 Depends on vulnerable versions of postcss-focus-visible
 Depends on vulnerable versions of postcss-focus-within
 Depends on vulnerable versions of postcss-font-variant
 Depends on vulnerable versions of postcss-gap-properties
 Depends on vulnerable versions of postcss-image-set-function
 Depends on vulnerable versions of postcss-initial
 Depends on vulnerable versions of postcss-lab-function
 Depends on vulnerable versions of postcss-logical
 Depends on vulnerable versions of postcss-media-minmax
 Depends on vulnerable versions of postcss-nesting
 Depends on vulnerable versions of postcss-overflow-shorthand
 Depends on vulnerable versions of postcss-page-break
 Depends on vulnerable versions of postcss-place
 Depends on vulnerable versions of postcss-pseudo-class-any-link
 Depends on vulnerable versions of postcss-replace-overflow-wrap
 Depends on vulnerable versions of postcss-selector-matches
 Depends on vulnerable versions of postcss-selector-not
 node_modules/postcss-preset-env
 css-blank-pseudo <=1.0.0
 Depends on vulnerable versions of postcss
 node_modules/css-blank-pseudo
 css-declaration-sorter <=5.1.2
 Depends on vulnerable versions of postcss
 node_modules/css-declaration-sorter
 css-has-pseudo <=1.0.0
 Depends on vulnerable versions of postcss
 node_modules/css-has-pseudo
 css-prefers-color-scheme <=4.0.0
 Depends on vulnerable versions of postcss
 node_modules/css-prefers-color-scheme
 cssnano <=4.1.11
 Depends on vulnerable versions of cssnano-preset-default
 Depends on vulnerable versions of postcss
 node_modules/cssnano
 cssnano-preset-default <=4.0.8
 Depends on vulnerable versions of css-declaration-sorter
 Depends on vulnerable versions of cssnano-util-raw-cache
 Depends on vulnerable versions of postcss
 Depends on vulnerable versions of postcss-calc
 Depends on vulnerable versions of postcss-colormin
 Depends on vulnerable versions of postcss-convert-values
 Depends on vulnerable versions of postcss-discard-comments
 Depends on vulnerable versions of postcss-discard-duplicates
 Depends on vulnerable versions of postcss-discard-empty
 Depends on vulnerable versions of postcss-discard-overridden
 Depends on vulnerable versions of postcss-merge-longhand
 Depends on vulnerable versions of postcss-merge-rules
 Depends on vulnerable versions of postcss-minify-font-values
 Depends on vulnerable versions of postcss-minify-gradients
 Depends on vulnerable versions of postcss-minify-params

Depends on vulnerable versions of postcss-minify-selectors
 Depends on vulnerable versions of postcss-normalize-charset
 Depends on vulnerable versions of postcss-normalize-display-values
 Depends on vulnerable versions of postcss-normalize-positions
 Depends on vulnerable versions of postcss-normalize-repeat-style
 Depends on vulnerable versions of postcss-normalize-string
 Depends on vulnerable versions of postcss-normalize-timing-functions
 Depends on vulnerable versions of postcss-normalize-unicode
 Depends on vulnerable versions of postcss-normalize-url
 Depends on vulnerable versions of postcss-normalize-whitespace
 Depends on vulnerable versions of postcss-ordered-values
 Depends on vulnerable versions of postcss-reduce-initial
 Depends on vulnerable versions of postcss-reduce-transforms
 Depends on vulnerable versions of postcss-svg
 Depends on vulnerable versions of postcss-unique-selectors
 node_modules/cssnano-preset-default
 cssnano-util-raw-cache *
 Depends on vulnerable versions of postcss
 node_modules/cssnano-util-raw-cache
 icss-utils <=4.1.1
 Depends on vulnerable versions of postcss
 node_modules/icss-utils
 postcss-attribute-case-insensitive <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-attribute-case-insensitive
 postcss-browser-comments <=3.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-browser-comments
 postcss-normalize <=9.0.0
 Depends on vulnerable versions of postcss
 Depends on vulnerable versions of postcss-browser-comments
 node_modules/postcss-normalize
 postcss-calc 4.1.0 - 7.0.5
 Depends on vulnerable versions of postcss
 node_modules/postcss-calc
 postcss-color-functional-notation <=3.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-color-functional-notation
 postcss-color-gray >=3.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-color-gray
 postcss-color-hex-alpha 1.3.0 - 6.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-color-hex-alpha
 postcss-color-mod-function *
 Depends on vulnerable versions of postcss
 node_modules/postcss-color-mod-function
 postcss-color-rebeccapurple 1.2.0 - 6.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-color-rebeccapurple
 postcss-colormin <=4.0.3

Depends on vulnerable versions of postcss
node_modules/postcss-colormin
postcss-convert-values <=4.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-convert-values
postcss-custom-media 4.0.0 - 7.0.8
Depends on vulnerable versions of postcss
node_modules/postcss-custom-media
postcss-custom-properties 3.3.0 - 10.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-custom-properties
postcss-custom-selectors 2.3.0 - 5.1.2
Depends on vulnerable versions of postcss
node_modules/postcss-custom-selectors
postcss-dir-pseudo-class <=5.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-dir-pseudo-class
postcss-discard-comments <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-discard-comments
postcss-discard-duplicates 1.1.0 - 4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-discard-duplicates
postcss-discard-empty 1.1.0 - 4.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-discard-empty
postcss-discard-overridden <=4.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-discard-overridden
postcss-double-position-gradients <=2.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-double-position-gradients
postcss-env-function <=3.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-env-function
postcss-flexbugs-fixes <=4.2.1
Depends on vulnerable versions of postcss
node_modules/postcss-flexbugs-fixes
postcss-focus-visible <=5.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-focus-visible
postcss-focus-within <=4.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-focus-within
postcss-font-variant 1.2.0 - 4.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-font-variant
postcss-gap-properties <=2.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-gap-properties
postcss-image-set-function <=3.0.1

Depends on vulnerable versions of postcss
node_modules/postcss-image-set-function
postcss-initial <=3.0.4
Depends on vulnerable versions of postcss
node_modules/postcss-initial
postcss-lab-function <=3.1.2
Depends on vulnerable versions of postcss
node_modules/postcss-lab-function
postcss-loader <=4.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-loader
postcss-logical <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-logical
postcss-media-minmax 1.2.0 - 4.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-media-minmax
postcss-merge-longhand <=4.0.11
Depends on vulnerable versions of postcss
Depends on vulnerable versions of stylehacks
node_modules/postcss-merge-longhand
postcss-merge-rules <=4.0.3
Depends on vulnerable versions of postcss
node_modules/postcss-merge-rules
postcss-minify-font-values <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-minify-font-values
postcss-minify-gradients <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-minify-gradients
postcss-minify-params <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-minify-params
postcss-minify-selectors <=4.0.2
Depends on vulnerable versions of postcss
node_modules/postcss-minify-selectors
postcss-modules-extract-imports <=2.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-modules-extract-imports
postcss-modules-local-by-default <=3.0.3
Depends on vulnerable versions of postcss
node_modules/postcss-modules-local-by-default
postcss-modules-scope <=2.2.0
Depends on vulnerable versions of postcss
node_modules/postcss-modules-scope
postcss-modules-values <=3.0.0
Depends on vulnerable versions of postcss
node_modules/postcss-modules-values
postcss-nesting <=7.0.1
Depends on vulnerable versions of postcss
node_modules/postcss-nesting

postcss-normalize-charset <=4.0.1
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-charset
 postcss-normalize-display-values <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-display-values
 postcss-normalize-positions <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-positions
 postcss-normalize-repeat-style <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-repeat-style
 postcss-normalize-string <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-string
 postcss-normalize-timing-functions <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-timing-functions
 postcss-normalize-unicode <=4.0.1
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-unicode
 postcss-normalize-url 1.1.0 - 4.0.1
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-url
 postcss-normalize-whitespace <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-normalize-whitespace
 postcss-ordered-values <=4.1.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-ordered-values
 postcss-overflow-shorthand <=2.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-overflow-shorthand
 postcss-page-break <=2.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-page-break
 postcss-place <=5.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-place
 postcss-pseudo-class-any-link <=6.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-pseudo-class-any-link
 postcss-reduce-initial <=4.0.3
 Depends on vulnerable versions of postcss
 node_modules/postcss-reduce-initial
 postcss-reduce-transforms <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-reduce-transforms
 postcss-replace-overflow-wrap <=3.0.0
 Depends on vulnerable versions of postcss
 node_modules/postcss-replace-overflow-wrap

postcss-safe-parser <=4.0.2
 Depends on vulnerable versions of postcss
 node_modules/postcss-safe-parser
 postcss-selector-matches *
 Depends on vulnerable versions of postcss
 node_modules/postcss-selector-matches
 postcss-selector-not <=4.0.1
 Depends on vulnerable versions of postcss
 node_modules/postcss-selector-not
 postcss-unique-selectors <=4.0.1
 Depends on vulnerable versions of postcss
 node_modules/postcss-unique-selectors
 stylehacks <=4.0.3
 Depends on vulnerable versions of postcss
 node_modules/stylehacks

request *
 Severity: moderate
 Server-Side Request Forgery in Request - <https://github.com/advisories/GHSA-p8p7-x288-28g6>
 Depends on vulnerable versions of tough-cookie
 fix available via `npm audit fix`
 node_modules/request
 request-promise-core *
 Depends on vulnerable versions of request
 node_modules/request-promise-core
 request-promise-native >=1.0.0
 Depends on vulnerable versions of request
 Depends on vulnerable versions of request-promise-core
 Depends on vulnerable versions of tough-cookie
 node_modules/request-promise-native

semver <5.7.2 || >=6.0.0 <6.3.1
 Severity: moderate
 semver vulnerable to Regular Expression Denial of Service - <https://github.com/advisories/GHSA-c2qf-rxjj-qqqw>
 semver vulnerable to Regular Expression Denial of Service - <https://github.com/advisories/GHSA-c2qf-rxjj-qqqw>
 fix available via `npm audit fix --force`
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/@typescript-eslint/typescript-estree/node_modules/semver
 node_modules/semver
 @typescript-eslint/typescript-estree <=2.0.0-alpha.6
 Depends on vulnerable versions of semver
 node_modules/@typescript-eslint/typescript-estree
 @typescript-eslint/eslint-plugin 0.2.1-alpha.1 - 2.0.0-alpha.6 || 7.15.1-alpha.0 - 8.0.0-alpha.11
 Depends on vulnerable versions of @typescript-eslint/parser
 Depends on vulnerable versions of @typescript-eslint/typescript-estree
 node_modules/@typescript-eslint/eslint-plugin
 @typescript-eslint/parser 0.2.1-alpha.1 - 2.0.0-alpha.6
 Depends on vulnerable versions of @typescript-eslint/typescript-estree

node_modules/@typescript-eslint/parser
 eslint-config-react-app 4.0.0-next.74 - 5.0.1
 Depends on vulnerable versions of @typescript-eslint/parser
 node_modules/eslint-config-react-app

serialize-javascript <=3.0.0
 Severity: high
 Cross-Site Scripting in serialize-javascript - <https://github.com/advisories/GHSA-h9rv-jmmf-4pgx>
 Insecure serialization leading to RCE in serialize-javascript - <https://github.com/advisories/GHSA-hxcc-f52p-wc94>
 fix available via ``npm audit fix --force``
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/serialize-javascript

shell-quote <=1.7.2
 Severity: critical
 Improper Neutralization of Special Elements used in a Command in Shell-quote - <https://github.com/advisories/GHSA-g4rg-993r-mgx7>
 fix available via ``npm audit fix --force``
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/shell-quote

sockjs <0.3.20
 Severity: moderate
 Improper Input Validation in SockJS-Node - <https://github.com/advisories/GHSA-c9g6-9335-x697>
 fix available via ``npm audit fix --force``
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/sockjs

terser <4.8.1
 Severity: high
 Terser insecure use of regular expressions leads to ReDoS - <https://github.com/advisories/GHSA-4wf5-vphf-c2xc>
 fix available via ``npm audit fix --force``
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/terser

tough-cookie <4.1.3
 Severity: moderate
 tough-cookie Prototype Pollution vulnerability - <https://github.com/advisories/GHSA-72xf-g2v4-qvf3>
 fix available via ``npm audit fix``
 node_modules/tough-cookie

webpack-dev-middleware <=5.3.3
 Severity: high
 Path traversal in webpack-dev-middleware - <https://github.com/advisories/GHSA-wr3j-pwj9-hqq6>
 fix available via ``npm audit fix --force``
 Will install react-scripts@5.0.1, which is a breaking change
 node_modules/webpack-dev-middleware

```
yargs-parser 6.0.0 - 13.1.1
Severity: moderate
yargs-parser Vulnerable to Prototype Pollution - https://github.com/advisories/GHSA-p9pc-299p-vxgp
fix available via `npm audit fix --force`
Will install react-scripts@5.0.1, which is a breaking change
node_modules/webpack-dev-server/node_modules/yargs-parser
yargs 8.0.0-candidate.0 - 12.0.5
Depends on vulnerable versions of yargs-parser
node_modules/webpack-dev-server/node_modules/yargs
```

162 vulnerabilities (1 low, 91 moderate, 67 high, 3 critical)

To address issues that do not require attention, run:
`npm audit fix`

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



The output indicates that you have fixed some vulnerabilities but there are still many remaining, including some that might require breaking changes to fix. Here's a step-by-step approach to address the remaining issues:

Step 1: Understand the Impact

Before applying fixes that might introduce breaking changes, it's important to understand the potential impact on your project. The `npm audit fix --force` command suggests upgrading `react-scripts` to a major version, which can introduce breaking changes.

Step 2: Review Vulnerabilities

Review the audit report carefully. Identify which vulnerabilities are critical or high priority and need immediate attention.

Step 3: Apply Fixes Selectively

Apply fixes for the highest severity issues first. Here's how you can approach this:

1. Fix Known Vulnerabilities:

```
bash
```

```
npm audit fix
```

2. Fix Critical Issues:

For remaining critical issues, you may need to apply fixes forcefully or update specific packages

manually.

3. Manual Updates:

Manually update dependencies that are causing vulnerabilities. For example, if `react-scripts` needs to be updated:

```
bash

npm install react-scripts@latest
```

Step 4: Test Your Application

After updating dependencies, thoroughly test your application to ensure it still functions correctly. Run your development server and check for any issues:

```
bash

npm start
```

Step 5: Address Remaining Issues

For any remaining vulnerabilities, consider if they are critical and whether they can be deferred. Not all vulnerabilities may need immediate fixes if they are not exploitable in your specific use case.

Example Commands

1. Run Audit Fix:

```
bash

npm audit fix
```

2. Force Fix (if necessary):

```
bash

npm audit fix --force
```

3. Manual Updates:

Update specific packages manually if needed. For example:

```
bash

npm install react-scripts@latest
```

Summary

1. **Review:** Check the audit report for critical and high-severity vulnerabilities.
2. **Fix:** Apply fixes using ``npm audit fix`` and selectively use ``npm audit fix --force`` if necessary.
3. **Update:** Manually update packages causing vulnerabilities.
4. **Test:** Ensure your application still works correctly after applying updates.

Final Note

It's crucial to maintain a balance between fixing vulnerabilities and ensuring the stability of your application. Always test thoroughly after applying any fixes or updates.

Would you like further assistance with a specific vulnerability or dependency?



```
python manage.py runserver is running on Django version 5.0.6, using settings
'testproject.settings'
```

```
Starting development server at http://127.0.0.1:8000/
```

```
then go for npm start it not working and showing this
```

```
Starting the development server...
```

```
Error: error:0308010C:digital envelope routines::unsupported
    at new Hash (node:internal/crypto/hash:69:19)
```

```
Starting the development server...
```

```
Error: error:0308010C:digital envelope routines::unsupported
    at new Hash (node:internal/crypto/hash:69:19)
```

```
    at Object.createHash (node:crypto:138:10)
```

```
    at module.exports
```

```
(C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\util\createHash.js:135:53)
```

```
    at NormalModule._initBuildHash
```

```
(C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:417:16)
```

```
    at handleParseError
```

```
(C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:471:10)
```

```
    at
```

```
C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:503:5
```

```
    at
```

```
C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:358:12
```

```
    at C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
```

```
runner\lib\LoaderRunner.js:373:3
```

```
    at iterateNormalLoaders (C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
```

```
runner\lib\LoaderRunner.js:214:10)
```

```
    at iterateNormalLoaders (C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
```

```
runner\lib\LoaderRunner.js:221:10)
```

```
C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\react-scripts\scripts\start.js:19
```

```
    throw err;
```

```
    ^
```

```
Error: error:0308010C:digital envelope routines::unsupported
```

```
    at new Hash (node:internal/crypto/hash:69:19)
```

```
    at Object.createHash (node:crypto:138:10)
```

```
    at module.exports
(C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\util\createHash.js:135:53)
    at NormalModule._initBuildHash
(C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:417:16)
    at
C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:452:10
    at
C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\webpack\lib\NormalModule.js:323:13
    at C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
runner\lib\LoaderRunner.js:367:11
    at C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
runner\lib\LoaderRunner.js:233:18
    at context.callback (C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\loader-
runner\lib\LoaderRunner.js:111:13)
    at C:\Users\nandu\Desktop\Dev\test\myfrontend\node_modules\babel-
loader\lib\index.js:59:103 {
  opensslErrorStack: [ 'error:03000086:digital envelope routines::initialization error' ],
  library: 'digital envelope routines',
  reason: 'unsupported',
  code: 'ERR_OSSL_EVP_UNSUPPORTED'
}
```

Node.js v20.0.0



The error you're encountering is related to changes in the OpenSSL support in Node.js v20.0.0. To resolve this issue, you can try one of the following solutions:

Solution 1: Downgrade Node.js

Downgrade Node.js to a version that is compatible with your project's dependencies, such as v14.x or v16.x. You can use a version manager like `nvm` (Node Version Manager) to easily switch between Node.js versions.

Install `nvm`:

1. Install `nvm` on Windows:

```
bash
```

```
npm install -g nvm
```

2. Install Node.js v16.x:

```
bash
```

```
nvm install 16
```



```
nvm use 16
```

3. Verify Node.js Version:

```
bash
```

```
node -v
```

Solution 2: Set Environment Variable

Set the `NODE_OPTIONS` environment variable to use legacy OpenSSL providers.

On Windows:

1. Open Command Prompt or PowerShell.
2. Set the environment variable:

```
bash
```

```
set NODE_OPTIONS=--openssl-legacy-provider
```

On macOS/Linux:

1. Open your terminal.
2. Set the environment variable:

```
bash
```

```
export NODE_OPTIONS=--openssl-legacy-provider
```

Solution 3: Update Dependencies

Update your project's dependencies to ensure they are compatible with the latest version of Node.js.

1. Update `react-scripts`:

```
bash
```

```
npm install react-scripts@latest
```

2. Update Other Dependencies:

```
bash
```

```
npm outdated
```

```
npm update
```

Solution 4: Modify `package.json`

You can also add the `NODE_OPTIONS` configuration directly in your `package.json`.

1. Open `package.json` and add the `start` script:

```
json
```

```
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject",
  "start-legacy": "set NODE_OPTIONS=--openssl-legacy-provider && react-scripts start"
}
```

2. Start your development server using the new script:

```
bash
```

```
npm run start-legacy
```

Summary

1. **Downgrade Node.js** to a version that is compatible with your project's dependencies using `nvm`.
2. **Set Environment Variable** to use legacy OpenSSL providers.
3. **Update Dependencies** to ensure compatibility with the latest Node.js version.
4. **Modify `package.json`** to include the `NODE_OPTIONS` setting.

Try one of these solutions to resolve the issue with starting your React development server.



```
npm install -g nvm
npm WARN deprecated nvm@0.0.4: This is NOT the correct nvm. Visit https://nvm.sh and use the
curl command to install it.
```

```
added 1 package in 2s
```



The warning indicates that you have installed an incorrect version of `nvm`. The correct way to install `nvm` on Windows is to use the `nvm-windows` package. Here's how you can do it:

Install `nvm` on Windows (Correct Way)

1. Uninstall the Incorrect `nvm`:

```
bash
```

```
npm uninstall -g nvm
```

2. Download `nvm-windows`:

Visit the [nvm-windows GitHub releases page](#) and download the `nvm-setup.zip` file.

3. Install `nvm-windows`:

- Extract the `nvm-setup.zip` file.
- Run the `nvm-setup.exe` installer and follow the installation instructions.

Use `nvm-windows` to Install Node.js

1. Open Command Prompt or PowerShell as an administrator.

2. Install Node.js Version 16:

```
bash
```

```
nvm install 16
```

3. Use Node.js Version 16:

```
bash
```

```
nvm use 16
```

4. Verify Node.js Version:

```
bash
```

```
node -v
```

Setting Environment Variable for OpenSSL

If you still want to try setting the `NODE_OPTIONS` environment variable, you can do so as follows:

1. Set Environment Variable in Command Prompt or PowerShell:

```
bash
```

```
set NODE_OPTIONS=--openssl-legacy-provider
```

2. Run your React app:

```
bash
```

```
npm start
```

Summary of Steps

1. Uninstall incorrect `nvm`:

```
bash
```

```
npm uninstall -g nvm
```

2. Download and install `nvm-windows` from [GitHub releases page](#).

3. Install and use Node.js version 16:

```
bash
```

```
nvm install 16  
nvm use 16
```

4. Verify Node.js version:

```
bash
```

```
node -v
```

5. Set environment variable for OpenSSL (if needed):

```
bash
```

```
set NODE_OPTIONS=--openssl-legacy-provider
```

6. Run your React app:

```
bash
```

```
npm start
```

These steps should help you resolve the issue with starting your React development server.



Compiled with warnings.

./src/App.js

Line 13:5: React Hook useEffect has a missing dependency: 'setData'. Either include it or remove the dependency array react-hooks/exhaustive-deps

Search for the keywords to learn more about each warning.

To ignore, add `// eslint-disable-next-line` to the line before.



The warning you are seeing is related to React's rules of hooks, specifically about how dependencies are managed in the `useEffect` hook. React wants you to include all variables and functions that are used inside `useEffect` as dependencies in its dependency array.

In your case, `setData` is a function returned by `useState`, which React guarantees will not change between re-renders. Therefore, it's generally safe to exclude `setData` from the dependency array, but to silence the warning, you can include it.

Here's how you can update your code:

Updated `src/App.js`

javascript

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, [setData]); // Include setData in the dependency array

  return (
    <div className="App">
      <h1>My Models</h1>
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

Alternatively, Disable the ESLint Warning

If you are certain that the dependency array is correct and you want to disable the ESLint warning, you can add a comment to ignore the line:

javascript

```
useEffect(() => {
  api.get('mymodels/')
    .then((response) => {
      setData(response.data);
    })
    .catch((error) => {
      console.error('There was an error fetching the data!', error);
    });
// eslint-disable-next-line react-hooks/exhaustive-deps
}, []); // You can leave the dependency array empty
```

Explanation

- **Including `setData` in the dependency array:** This is the recommended approach as it satisfies the linter rule. Even though `setData` is stable, including it makes the dependencies explicit and prevents potential issues if the linter rule changes or if you refactor your code.
- **Disabling the warning:** If you are sure that the warning is unnecessary, you can disable it for the specific line. However, use this with caution as it might lead to unexpected behavior in other cases.

By following either of these approaches, you can address the warning and ensure your code adheres to React's rules of hooks.



Compiled successfully!

You can now view myfrontend in the browser.

Local: <http://localhost:3000>
On Your Network: <http://192.168.39.46:3000>

Note that the development build is not optimized.
To create a production build, use `npm run build`.
i got this . now how i excute App function in App.js.



Great! Your React application is running successfully. The `App` function in `App.js` is already being executed because it is the main component of your React application.
Here's a quick guide on how the `App` component works and how you can see it in action:

Understanding `App.js`

1. **App Component:** The `App` component is a functional React component that is defined in `App.js`.
2. **useEffect Hook:** Inside `App`, the `useEffect` hook fetches data from your Django backend when the component mounts.

3. **Rendering:** The ``App`` component renders a list of items fetched from the backend.

Steps to See ``App`` Component in Action

1. Start the Development Server:

Since you have already started the development server, your app should be running at ``http://localhost:3000``.

2. Open the Browser:

Open a web browser and navigate to ``http://localhost:3000``. You should see the content rendered by your ``App`` component.

3. Check the Output:

If the data fetch from your Django backend is successful, you should see the list of items rendered in the browser.

Example ``App.js`` for Clarity

Here's the ``App.js`` file you should have:

javascript

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, [setData]); // Include setData in the dependency array

  return (
    <div className="App">
      <h1>My Models</h1>
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

Ensuring Data Fetch is Successful

- **Backend is Running:** Make sure your Django backend is running and accessible at ``http://127.0.0.1:8000/api/mymodels/``.
- **Correct API Endpoint:** Ensure that the API endpoint in your ``api.js`` file matches the endpoint provided by your Django backend.

Example ``api.js``

Here's what your `api.js` should look like:

```
javascript
```

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

Viewing the Application

1. **Visit `http://localhost:3000`:** Open this URL in your browser to see your React application in action.
2. **Inspect Network Requests:** Use the browser's developer tools (F12) to inspect network requests and ensure that your React app is successfully fetching data from your Django backend.

By following these steps, you can see the `App` function in `App.js` executed and rendering data from your Django backend. If everything is set up correctly, you should see a list of items fetched from your API displayed in your browser.



http://localhost:3000 does have any content



If `http://localhost:3000` doesn't show any content, there are a few potential issues to investigate:

Step-by-Step Debugging

1. **Check Console for Errors:**
Open the browser's developer tools (F12 or right-click and select "Inspect") and go to the "Console" tab to see if there are any errors logged.
2. **Check Network Requests:**
Go to the "Network" tab in the developer tools and refresh the page. Look for the request to `http://127.0.0.1:8000/api/mymodels/` and see if it was successful or if there were any errors.
3. **Verify API Response:**
Manually access the API endpoint in the browser or using a tool like `curl` or Postman:

```
bash
```

```
curl http://127.0.0.1:8000/api/mymodels/
```

Make sure that the API is returning the expected data.

Sample Code Verification

`App.js`

Ensure your `App.js` file looks like this:

javascript

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        setData(response.data);
      })
      .catch((error) => {
        console.error('There was an error fetching the data!', error);
      });
  }, [setData]);

  return (
    <div className="App">
      <h1>My Models</h1>
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

`api.js`

Ensure your `api.js` file is correctly set up to point to your Django backend:

javascript

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

Additional Checks

1. Check Django Server:

Make sure your Django server is running without errors. You should see log entries indicating that it is serving requests:

```
bash
```

```
python manage.py runserver
```

2. Check React App Configuration:

Ensure that your React app is correctly configured and running:

```
bash
```

```
npm start
```

3. Default Content in `App.js`:

Temporarily replace the content of your `App.js` to ensure that the basic React rendering works:

```
javascript
```

```
function App() {
  return (
    <div className="App">
      <h1>Hello, World!</h1>
    </div>
  );
}

export default App;
```

If you see "Hello, World!" when you visit `http://localhost:3000`, the issue is likely with the API call.

4. Verify CORS Configuration:

Ensure that your Django backend allows requests from your React frontend. Add or verify the CORS configuration in your Django settings:

```
python
```

```
INSTALLED_APPS = [
    ...
    'corsheaders',
    ...
]

MIDDLEWARE = [
    ...
    'corsheaders.middleware.CorsMiddleware',
    ...
]

CORS_ALLOW_ALL_ORIGINS = True
```

Conclusion

By following these steps, you should be able to identify and resolve the issue preventing your React application from displaying content. If you have checked all of the above and still face issues, please

provide more details about any errors or logs you encounter for further assistance.



i have add value at `http://127.0.0.1:8000/api/mymodels/`
then do npm start, got this on browser consol

```
App.js:4 Uncaught TypeError: Object is not a function or its return value is not iterable
    at App (App.js:4:1)
    at renderWithHooks (react-dom.development.js:15486:1)
    at mountIndeterminateComponent (react-dom.development.js:20103:1)
    at beginWork (react-dom.development.js:21626:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
    at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
    at invokeGuardedCallback (react-dom.development.js:4277:1)
    at beginWork$1 (react-dom.development.js:27490:1)
    at performUnitOfWork (react-dom.development.js:26599:1)
    at workLoopSync (react-dom.development.js:26505:1)
    at renderRootSync (react-dom.development.js:26473:1)
    at performConcurrentWorkOnRoot (react-dom.development.js:25777:1)
    at workLoop (scheduler.development.js:266:1)
    at flushWork (scheduler.development.js:239:1)
    at performWorkUntilDeadline (scheduler.development.js:533:1)
    at run (setImmediate.js:40:1)
    at runIfPresent (setImmediate.js:69:1)
    at onGlobalMessage (setImmediate.js:109:1)
App.js:4 Uncaught TypeError: Object is not a function or its return value is not iterable
    at App (App.js:4:1)
    at renderWithHooks (react-dom.development.js:15486:1)
    at mountIndeterminateComponent (react-dom.development.js:20103:1)
    at beginWork (react-dom.development.js:21626:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
    at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
    at invokeGuardedCallback (react-dom.development.js:4277:1)
    at beginWork$1 (react-dom.development.js:27490:1)
    at performUnitOfWork (react-dom.development.js:26599:1)
    at workLoopSync (react-dom.development.js:26505:1)
    at renderRootSync (react-dom.development.js:26473:1)
    at recoverFromConcurrentError (react-dom.development.js:25889:1)
    at performConcurrentWorkOnRoot (react-dom.development.js:25789:1)
    at workLoop (scheduler.development.js:266:1)
    at flushWork (scheduler.development.js:239:1)
    at performWorkUntilDeadline (scheduler.development.js:533:1)
    at run (setImmediate.js:40:1)
    at runIfPresent (setImmediate.js:69:1)
    at onGlobalMessage (setImmediate.js:109:1)
index.js:1 The above error occurred in the <App> component:

    at App (http://localhost:3000/static/js/main.chunk.js:37:81)
```

Consider adding an error boundary to your tree to customize error handling behavior. Visit <https://reactjs.org/link/error-boundaries> to learn more about error boundaries.

```

console.<computed> @ index.js:1
Show 1 more frame
Show less
react-dom.development.js:26962 Uncaught TypeError: Object is not a function or its return value
is not iterable
    at App (App.js:4:1)
    at renderWithHooks (react-dom.development.js:15486:1)
    at mountIndeterminateComponent (react-dom.development.js:20103:1)
    at beginWork (react-dom.development.js:21626:1)
    at beginWork$1 (react-dom.development.js:27465:1)
    at performUnitOfWork (react-dom.development.js:26599:1)
    at workLoopSync (react-dom.development.js:26505:1)
    at renderRootSync (react-dom.development.js:26473:1)
    at recoverFromConcurrentError (react-dom.development.js:25889:1)
    at performConcurrentWorkOnRoot (react-dom.development.js:25789:1)
    at workLoop (scheduler.development.js:266:1)
    at flushWork (scheduler.development.js:239:1)
    at performWorkUntilDeadline (scheduler.development.js:533:1)
    at run (setImmediate.js:40:1)
    at runIfPresent (setImmediate.js:69:1)
    at onGlobalMessage (setImmediate.js:109:1)
VM8:2 Uncaught ReferenceError: process is not defined
    at 4043 (<anonymous>:2:13168)
    at r (<anonymous>:2:306599)
    at 8048 (<anonymous>:2:9496)
    at r (<anonymous>:2:306599)
    at 8641 (<anonymous>:2:1379)
    at r (<anonymous>:2:306599)
    at <anonymous>:2:315627
    at <anonymous>:2:324225
    at <anonymous>:2:324229
    at e.onload (index.js:1:1)

```



The error you're seeing, `TypeError: Object is not a function or its return value is not iterable`, typically indicates an issue with how the data is being handled or the API response. Let's troubleshoot and ensure everything is correctly set up.

Debugging Steps

1. Verify API Response:

Ensure that the API endpoint returns a valid JSON response. You can use tools like Postman or curl to check the response.

2. Check the API Call in React:

Ensure that the API call is correct and the response is properly handled.

3. Add Error Handling:

Add more detailed error handling to understand what is going wrong.

Updated `App.js` with Debugging and Error Handling

javascript

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);
  const [error, setError] = useState(null);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        if (Array.isArray(response.data)) {
          setData(response.data);
        } else {
          throw new Error('Data is not an array');
        }
      })
      .catch((error) => {
        console.error('There was an error fetching the data:', error);
        setError(error.message);
      });
  }, []);

  return (
    <div className="App">
      <h1>My Models</h1>
      {error && <p>Error: {error}</p>}
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

Explanation

- 1. State for Error Handling:** Added an `error` state to track any errors that occur during the data fetch.
- 2. Error Handling in `useEffect`:** Improved error handling to catch and log errors, and update the `error` state.
- 3. Condition for Array Check:** Ensure that `response.data` is an array before setting it in state.

Steps to Test

1. Check API Response:

bash

```
curl http://127.0.0.1:8000/api/mymodels/
```

Ensure the response is a valid JSON array.

2. Run the React App:

```
bash
```

```
npm start
```

3. Check Browser Console:

Look at the browser console for any new error messages or logs that might help identify the issue.

Verify `api.js`

Ensure that `api.js` is correctly set up to point to your Django backend.

```
javascript
```

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

By following these steps and adding better error handling, you should be able to identify and resolve the issue preventing your React app from displaying the data. If you encounter specific error messages or need further assistance, please provide the details for more targeted help.



Compiled with warnings.

./src/App.js

Line 18:5: React Hook `useEffect` has missing dependencies: `'setData'` and `'setError'`. Either include them or remove the dependency array `react-hooks/exhaustive-deps`

Search for the keywords to learn more about each warning.

To ignore, add `// eslint-disable-next-line` to the line before.



To address the warning about missing dependencies in the `useEffect` hook, you should include all state setter functions (`setData` and `setError`) in the dependency array. Although these functions are stable and do not change between renders, including them will satisfy the linter.

Here's the updated `App.js`:

Updated `App.js`

javascript

```
import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);
  const [error, setError] = useState(null);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        if (Array.isArray(response.data)) {
          setData(response.data);
        } else {
          throw new Error('Data is not an array');
        }
      })
      .catch((error) => {
        console.error('There was an error fetching the data:', error);
        setError(error.message);
      });
  }, [setData, setError]); // Include setData and setError in the dependency array

  return (
    <div className="App">
      <h1>My Models</h1>
      {error && <p>Error: {error}</p>}
      <ul>
        {data.map((item) => (
          <li key={item.id}>{item.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default App;
```

Explanation

- **Dependencies Array:** Adding `setData` and `setError` to the dependencies array to satisfy the linter.

Steps to Test

1. Start the React App:

```
bash
```

```
npm start
```

2. Check Browser Console:

Open the browser console and look for any error messages or logs.

3. Verify Output:

Visit `http://localhost:3000` and ensure the data is being fetched and displayed correctly.

Verify API Response

Ensure your API response is an array of objects with the expected structure. You can test this using curl or Postman:

```
bash
```

```
curl http://127.0.0.1:8000/api/mymodels/
```

The response should be a JSON array, such as:

```
json
```

```
[
  { "id": 1, "name": "Item 1" },
  { "id": 2, "name": "Item 2" }
]
```

By including ``setData`` and ``setError`` in the dependency array and ensuring the API response is correctly formatted, you should be able to resolve the warning and display the data correctly in your React application.



```
App.js:4 Uncaught TypeError: Object is not a function or its return value is not iterable
    at App (App.js:4:1)
    at renderWithHooks (react-dom.development.js:15486:1)
    at mountIndeterminateComponent (react-dom.development.js:20103:1)
    at beginWork (react-dom.development.js:21626:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
    at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
    at invokeGuardedCallback (react-dom.development.js:4277:1)
    at beginWork$1 (react-dom.development.js:27490:1)
    at performUnitOfWork (react-dom.development.js:26599:1)
    at workLoopSync (react-dom.development.js:26505:1)
    at renderRootSync (react-dom.development.js:26473:1)
    at performConcurrentWorkOnRoot (react-dom.development.js:25777:1)
    at workLoop (scheduler.development.js:266:1)
    at flushWork (scheduler.development.js:239:1)
    at performWorkUntilDeadline (scheduler.development.js:533:1)
    at run (setImmediate.js:40:1)
    at runIfPresent (setImmediate.js:69:1)
    at onGlobalMessage (setImmediate.js:109:1)
App.js:4 Uncaught TypeError: Object is not a function or its return value is not iterable
    at App (App.js:4:1)
    at renderWithHooks (react-dom.development.js:15486:1)
    at mountIndeterminateComponent (react-dom.development.js:20103:1)
    at beginWork (react-dom.development.js:21626:1)
    at HTMLUnknownElement.callCallback (react-dom.development.js:4164:1)
```



```

at Object.invokeGuardedCallbackDev (react-dom.development.js:4213:1)
at invokeGuardedCallback (react-dom.development.js:4277:1)
at beginWork$1 (react-dom.development.js:27490:1)
at performUnitOfWork (react-dom.development.js:26599:1)
at workLoopSync (react-dom.development.js:26505:1)
at renderRootSync (react-dom.development.js:26473:1)
at recoverFromConcurrentError (react-dom.development.js:25889:1)
at performConcurrentWorkOnRoot (react-dom.development.js:25789:1)
at workLoop (scheduler.development.js:266:1)
at flushWork (scheduler.development.js:239:1)
at performWorkUntilDeadline (scheduler.development.js:533:1)
at run (setImmediate.js:40:1)
at runIfPresent (setImmediate.js:69:1)
at onGlobalMessage (setImmediate.js:109:1)
index.js:1 The above error occurred in the <App> component:

```

```

  at App (http://localhost:3000/static/js/main.chunk.js:37:81)

```

Consider adding an error boundary to your tree to customize error handling behavior.
Visit <https://reactjs.org/link/error-boundaries> to learn more about error boundaries.

console.<computed> @ index.js:1

Show 1 more frame

Show less

react-dom.development.js:26962 Uncaught TypeError: Object is not a function or its return value is not iterable

```

  at App (App.js:4:1)
  at renderWithHooks (react-dom.development.js:15486:1)
  at mountIndeterminateComponent (react-dom.development.js:20103:1)
  at beginWork (react-dom.development.js:21626:1)
  at beginWork$1 (react-dom.development.js:27465:1)
  at performUnitOfWork (react-dom.development.js:26599:1)
  at workLoopSync (react-dom.development.js:26505:1)
  at renderRootSync (react-dom.development.js:26473:1)
  at recoverFromConcurrentError (react-dom.development.js:25889:1)
  at performConcurrentWorkOnRoot (react-dom.development.js:25789:1)
  at workLoop (scheduler.development.js:266:1)
  at flushWork (scheduler.development.js:239:1)
  at performWorkUntilDeadline (scheduler.development.js:533:1)
  at run (setImmediate.js:40:1)
  at runIfPresent (setImmediate.js:69:1)
  at onGlobalMessage (setImmediate.js:109:1)

```

VM9:2 Uncaught ReferenceError: process is not defined

```

  at 4043 (<anonymous>:2:13168)
  at r (<anonymous>:2:306599)
  at 8048 (<anonymous>:2:9496)
  at r (<anonymous>:2:306599)
  at 8641 (<anonymous>:2:1379)
  at r (<anonymous>:2:306599)
  at <anonymous>:2:315627
  at <anonymous>:2:324225

```

```
at <anonymous>:2:324229
at e.onload (index.js:1:1)
```



The error ``TypeError: Object is not a function or its return value is not iterable`` indicates that there might be an issue with how you're handling the response from the API call. Let's walk through a few steps to diagnose and fix the issue.

Step-by-Step Debugging and Fix

1. Ensure API Response is Correct:

Check that the API returns the correct format. The response should be a JSON array of objects.

```
json

[
  { "id": 1, "name": "Item 1" },
  { "id": 2, "name": "Item 2" }
]
```

2. Inspect ``api.js``:

Verify that the ``api.js`` file is correctly set up to use Axios and point to your Django backend.

```
javascript

import axios from 'axios';

const api = axios.create({
  baseURL: 'http://127.0.0.1:8000/api/', // Django server URL
});

export default api;
```

3. Check ``App.js``:

Ensure your ``App.js`` file handles the response correctly. Add console logs to inspect the response.

```
javascript

import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]);
  const [error, setError] = useState(null);

  useEffect(() => {
    api.get('mymodels/')
      .then((response) => {
        console.log(response.data); // Log the response data
        if (Array.isArray(response.data)) {
          setData(response.data);
        } else {
          throw new Error('Data is not an array');
        }
      })
      .catch((error) => {
        setError(error);
      });
  });
}
```

```

    }
  })
  .catch((error) => {
    console.error('There was an error fetching the data:', error);
    setError(error.message);
  });
}, [setData, setError]); // Include setData and setError in the dependency array

return (
  <div className="App">
    <h1>My Models</h1>
    {error && <p>Error: {error}</p>}
    <ul>
      {data.map((item) => (
        <li key={item.id}>{item.name}</li>
      ))}
    </ul>
  </div>
);
}

export default App;

```

4. Check for Errors in Browser Console:

Open the browser console (F12 or right-click and select "Inspect" -> "Console") and look for any logged errors or logs from the above code.

5. Check Your Backend:

Ensure your Django backend is correctly configured to return the expected JSON response. You can test the endpoint using a tool like Postman or curl:

```
bash
```

```
curl http://127.0.0.1:8000/api/mymodels/
```

6. Verify Your Backend API:

Ensure your Django view is returning a proper JSON response. Here's an example of a Django view that returns a list of models:

```
python
```

```

from rest_framework import viewsets
from .models import MyModel
from .serializers import MyModelSerializer

class MyModelViewSet(viewsets.ModelViewSet):
    queryset = MyModel.objects.all()
    serializer_class = MyModelSerializer

```

Further Debugging

If the issue persists, you can add more detailed debugging statements:

1. Log Full Response Object:

javascript

```

useEffect(() => {
  api.get('mymodels/')
    .then((response) => {
      console.log('Full response:', response); // Log the full response
      if (Array.isArray(response.data)) {
        setData(response.data);
      } else {
        throw new Error('Data is not an array');
      }
    })
    .catch((error) => {
      console.error('There was an error fetching the data:', error);
      setError(error.message);
    });
}, [setData, setError]);

```

2. Check for Typos:

Ensure there are no typos in your component or API setup.

Final Notes

- **Correct API URL:** Double-check the URL used in `api.js` to ensure it matches your Django API endpoint.
- **Valid JSON Response:** Ensure your Django API is returning a valid JSON response. Use `return JsonResponse(data, safe=False)` if needed.

By following these steps and ensuring your API and React components are correctly set up, you should be able to resolve the issue and display the data in your React application. If you continue to experience issues, please provide the details of the logs or errors you encounter for further assistance.



```

import React, { useEffect, useState } from 'react';
import api from './api';

function App(){
  const [data, setData] = useState([]);
  const [error, setError] = useState(null);

  useEffect(()=> {
    //from router in the app urls.py
    api.get('mymodels/')
      .then((response)=>{
        console.log(response.data); // Log the response data
        if (Array.isArray(response.data)){
          setData(response.data);
        }
        else{
          throw new Error('Data is not a array');
        }
      })
  })
}

```

```

})
.catc((error)=>{
  console.error('There was an error fetching the data!',error);
  setError(error.message);
});
],[setData,setError]); // Include setData in the dependency array
return(
<div className='App'>
  <h1>My Models</h1>
  {error && <p>Error : {error}</p>}
  <ul>
    {data.map((item)=>(
      <li key={item.id}> {item.name} </li>
    ))}
  </ul>
</div>
);
}
export default App;
what is the difference and eroor here

```



There are a few issues and typos in your code that need to be corrected:

1. **Typo in `useState`**: The correct spelling is `useState`, not `usestate`.
2. **Typo in `catch`**: The correct spelling is `catch`, not `catc`.

Here's the corrected version of your code:

Corrected `App.js`

javascript

```

import React, { useEffect, useState } from 'react';
import api from './api';

function App() {
  const [data, setData] = useState([]); // Corrected to useState
  const [error, setError] = useState(null); // Corrected to useState

  useEffect(() => {
    // From router in the app urls.py
    api.get('mymodels/')
      .then((response) => {
        console.log(response.data); // Log the response data
        if (Array.isArray(response.data)) {
          setData(response.data);
        } else {
          throw new Error('Data is not an array');
        }
      })
      .catch((error) => { // Corrected to catch
        console.error('There was an error fetching the data!', error);
        setError(error.message);
      });
  }, [setData, setError]); // Include setData and setError in the dependency array

```

```
return (
  <div className='App'>
    <h1>My Models</h1>
    {error && <p>Error: {error}</p>}
    <ul>
      {data.map((item) => (
        <li key={item.id}>{item.name}</li>
      ))}
    </ul>
  </div>
);
}

export default App;
```

Explanation of Corrections

1. Correct `useState` Import:

javascript

```
import React, { useEffect, useState } from 'react';
```

2. Correct `useState` Usage:

javascript

```
const [data, setData] = useState([]); // Corrected to useState
const [error, setError] = useState(null); // Corrected to useState
```

3. Correct `catch` Block:

javascript

```
.catch((error) => { // Corrected to catch
  console.error('There was an error fetching the data!', error);
  setError(error.message);
});
```

4. Ensure Correct Dependency Array:

javascript

```
useEffect(() => {
  // API call and state setting
}, [setData, setError]); // Include setData and setError in the dependency array
```

Testing

1. Run the React App:

```
bash
```

```
npm start
```

2. Check Browser Console:

Open the browser console and look for any logged errors or logs.

3. Verify Output:

Visit `http://localhost:3000`` and ensure the data is being fetched and displayed correctly.

By fixing these typos, your React component should work correctly and fetch data from your Django backend. If you encounter any further issues, please provide additional details for further assistance.



output from `http://localhost:3000` is

My Models

Error : Network Error

and console shows

Access to XMLHttpRequest at 'http://127.0.0.1:8000/api/mymodels/' from origin 'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

App.js:22 There was an error fetching the data! AxiosError {message: 'Network Error', name: 'AxiosError', code: 'ERR_NETWORK', config: {...}, request: XMLHttpRequest, ...}

console.<computed> @ index.js:1

(anonymous) @ App.js:22

Promise.catch (async)

(anonymous) @ App.js:20

commitHookEffectListMount @ react-dom.development.js:23189

commitPassiveMountOnFiber @ react-dom.development.js:24970

commitPassiveMountEffects_complete @ react-dom.development.js:24930

commitPassiveMountEffects_begin @ react-dom.development.js:24917

commitPassiveMountEffects @ react-dom.development.js:24905

flushPassiveEffectsImpl @ react-dom.development.js:27078

flushPassiveEffects @ react-dom.development.js:27023

(anonymous) @ react-dom.development.js:26808

workLoop @ scheduler.development.js:266

flushWork @ scheduler.development.js:239

performWorkUntilDeadline @ scheduler.development.js:533

run @ setImmediate.js:40

runIfPresent @ setImmediate.js:69

onGlobalMessage @ setImmediate.js:109

postMessage (async)

registerImmediate @ setImmediate.js:120

setImmediate @ setImmediate.js:27

schedulePerformWorkUntilDeadline @ scheduler.development.js:564

performWorkUntilDeadline @ scheduler.development.js:538

run @ setImmediate.js:40

runIfPresent @ setImmediate.js:69

```

onGlobalMessage @ setImmediate.js:109
postMessage (async)
registerImmediate @ setImmediate.js:120
setImmediate @ setImmediate.js:27
schedulePerformWorkUntilDeadline @ scheduler.development.js:564
requestHostCallback @ scheduler.development.js:588
unstable_scheduleCallback @ scheduler.development.js:441
scheduleCallback$1 @ react-dom.development.js:27576
ensureRootIsScheduled @ react-dom.development.js:25722
scheduleUpdateOnFiber @ react-dom.development.js:25570
updateContainer @ react-dom.development.js:28897
push../node_modules/react-dom/cjs/react-
dom.development.js.ReactDOMHydrationRoot.render.ReactDOMRoot.render @ react-
dom.development.js:29353
./src/index.js @ index.js:8
__webpack_require__ @ bootstrap:789
fn @ bootstrap:150
1 @ reportWebVitals.js:13
__webpack_require__ @ bootstrap:789
checkDeferredModules @ bootstrap:45
webpackJsonpCallback @ bootstrap:32
(anonymous) @ main.chunk.js:1
Show 32 more frames
Show less
App.js:10

```

```

GET http://127.0.0.1:8000/api/mymodels/ net::ERR_FAILED 200 (OK)
dispatchXhrRequest @ xhr.js:188
xhr @ xhr.js:15
dispatchRequest @ dispatchRequest.js:51
_request @ Axios.js:173
request @ Axios.js:40
Axios.<computed> @ Axios.js:199
wrap @ bind.js:5
(anonymous) @ App.js:10
commitHookEffectListMount @ react-dom.development.js:23189
commitPassiveMountOnFiber @ react-dom.development.js:24970
commitPassiveMountEffects_complete @ react-dom.development.js:24930
commitPassiveMountEffects_begin @ react-dom.development.js:24917
commitPassiveMountEffects @ react-dom.development.js:24905
flushPassiveEffectsImpl @ react-dom.development.js:27078
flushPassiveEffects @ react-dom.development.js:27023
(anonymous) @ react-dom.development.js:26808
workLoop @ scheduler.development.js:266
flushWork @ scheduler.development.js:239
performWorkUntilDeadline @ scheduler.development.js:533
run @ setImmediate.js:40
runIfPresent @ setImmediate.js:69
onGlobalMessage @ setImmediate.js:109
postMessage (async)

```



```

registerImmediate @ setImmediate.js:120
setImmediate @ setImmediate.js:27
schedulePerformWorkUntilDeadline @ scheduler.development.js:564
performWorkUntilDeadline @ scheduler.development.js:538
run @ setImmediate.js:40
runIfPresent @ setImmediate.js:69
onGlobalMessage @ setImmediate.js:109
postMessage (async)
registerImmediate @ setImmediate.js:120
setImmediate @ setImmediate.js:27
schedulePerformWorkUntilDeadline @ scheduler.development.js:564
requestHostCallback @ scheduler.development.js:588
unstable_scheduleCallback @ scheduler.development.js:441
scheduleCallback$1 @ react-dom.development.js:27576
ensureRootIsScheduled @ react-dom.development.js:25722
scheduleUpdateOnFiber @ react-dom.development.js:25570
updateContainer @ react-dom.development.js:28897
push../node_modules/react-dom/cjs/react-
dom.development.js.ReactDOMHydrationRoot.render.ReactDOMRoot.render @ react-
dom.development.js:29353
./src/index.js @ index.js:8
__webpack_require__ @ bootstrap:789
fn @ bootstrap:150
1 @ reportWebVitals.js:13
__webpack_require__ @ bootstrap:789
checkDeferredModules @ bootstrap:45
webpackJsonpCallback @ bootstrap:32
(anonymous) @ main.chunk.js:1
Show 38 more frames
Show less
localhost/:1 Access to XMLHttpRequest at 'http://127.0.0.1:8000/api/mymodels/' from origin
'http://localhost:3000' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is
present on the requested resource.
App.js:22 There was an error fetching the data! AxiosError {message: 'Network Error', name:
'AxiosError', code: 'ERR_NETWORK', config: {...}, request: XMLHttpRequest, ...}
console.<computed> @ index.js:1
(anonymous) @ App.js:22
Promise.catch (async)
(anonymous) @ App.js:20
commitHookEffectListMount @ react-dom.development.js:23189
invokePassiveEffectMountInDEV @ react-dom.development.js:25193
invokeEffectsInDev @ react-dom.development.js:27390
commitDoubleInvokeEffectsInDEV @ react-dom.development.js:27369
flushPassiveEffectsImpl @ react-dom.development.js:27095
flushPassiveEffects @ react-dom.development.js:27023
(anonymous) @ react-dom.development.js:26808
workLoop @ scheduler.development.js:266
flushWork @ scheduler.development.js:239
performWorkUntilDeadline @ scheduler.development.js:533
run @ setImmediate.js:40
runIfPresent @ setImmediate.js:69

```

```

onGlobalMessage @ setImmediate.js:109
postMessage (async)
registerImmediate @ setImmediate.js:120
setImmediate @ setImmediate.js:27
schedulePerformWorkUntilDeadline @ scheduler.development.js:564
performWorkUntilDeadline @ scheduler.development.js:538
run @ setImmediate.js:40
runIfPresent @ setImmediate.js:69
onGlobalMessage @ setImmediate.js:109
postMessage (async)
registerImmediate @ setImmediate.js:120
setImmediate @ setImmediate.js:27
schedulePerformWorkUntilDeadline @ scheduler.development.js:564
requestHostCallback @ scheduler.development.js:588
unstable_scheduleCallback @ scheduler.development.js:441
scheduleCallback$1 @ react-dom.development.js:27576
ensureRootIsScheduled @ react-dom.development.js:25722
scheduleUpdateOnFiber @ react-dom.development.js:25570
updateContainer @ react-dom.development.js:28897
push../node_modules/react-dom/cjs/react-
dom.development.js.ReactDOMHydrationRoot.render.ReactDOMRoot.render @ react-
dom.development.js:29353
./src/index.js @ index.js:8
__webpack_require__ @ bootstrap:789
fn @ bootstrap:150
1 @ reportWebVitals.js:13
__webpack_require__ @ bootstrap:789
checkDeferredModules @ bootstrap:45
webpackJsonpCallback @ bootstrap:32
(anonymous) @ main.chunk.js:1
Show 31 more frames
Show less
App.js:10

```

GET http://127.0.0.1:8000/api/mymodels/ net::ERR_FAILED 200 (OK)



The error you are encountering is due to the Cross-Origin Resource Sharing (CORS) policy. The CORS policy on your Django backend needs to be configured to allow requests from your React frontend. Here's how you can resolve this issue:

Step 1: Install `django-cors-headers`

First, install the `django-cors-headers` package:

```
bash
```

```
pip install django-cors-headers
```

Step 2: Configure `django-cors-headers` in Django

1. Add `corsheaders` to `INSTALLED_APPS`:

Add `corsheaders` to the `INSTALLED_APPS` list in your Django `settings.py` file:

```
python
```

```
INSTALLED_APPS = [
    ...
    'corsheaders',
    ...
]
```

2. Add `CorsMiddleware` to Middleware:

Add `CorsMiddleware` to the top of the `MIDDLEWARE` list in `settings.py`:

```
python
```

```
MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    ...
]
```

3. Configure CORS:

Add the following configuration to `settings.py` to allow requests from your React app:

```
python
```

```
CORS_ALLOWED_ORIGINS = [
    "http://localhost:3000",
]
```

Alternatively, if you want to allow all origins (useful for development but not recommended for production), you can use:

```
python
```

```
CORS_ALLOW_ALL_ORIGINS = True
```

Full Example of `settings.py` Changes

Here is how your `settings.py` might look after making the necessary changes:

```
python
```

```
# settings.py

INSTALLED_APPS = [
    ...
    'corsheaders',
    'rest_framework',
    'myapp', # your app
    ...
]

MIDDLEWARE = [
    'corsheaders.middleware.CorsMiddleware',
    ...
    'django.middleware.common.CommonMiddleware',
    ...
]

# Allow only localhost:3000 for development
CORS_ALLOWED_ORIGINS = [
    "http://localhost:3000",
]

# Alternatively, allow all origins (for development only)
# CORS_ALLOW_ALL_ORIGINS = True
```

Step 3: Restart Django Server

After making these changes, restart your Django server:

```
bash
```

```
python manage.py runserver
```

Step 4: Verify the React App

Now, run your React app:

```
bash
```

```
npm start
```

Verify CORS Configuration

Open your browser and navigate to `http://localhost:3000`. The network request should now be allowed, and you should see the data fetched from your Django backend.

By configuring `django-cors-headers`, you enable your Django backend to accept requests from your React frontend, resolving the CORS policy issue. If you encounter any further issues, please provide additional details for further assistance.



```
CORS_ALLOWED_ORIGINS=[  
    "http://localhost:3000",  
]
```



There is a typo in your configuration. It should be `CORS_ALLOWED_ORIGINS` instead of `CORS_ALLOWED_ORIGINS`. Here is the corrected configuration:

Correct `settings.py` Configuration

python

```
# settings.py  
  
INSTALLED_APPS = [  
    ...  
    'corsheaders',  
    'rest_framework',  
    'myapp', # your app  
    ...  
]  
  
MIDDLEWARE = [  
    'corsheaders.middleware.CorsMiddleware',  
    ...  
    'django.middleware.common.CommonMiddleware',  
    ...  
]  
  
# Allow only localhost:3000 for development  
CORS_ALLOWED_ORIGINS = [  
    "http://localhost:3000",  
]  
  
# Alternatively, allow all origins (for development only)  
# CORS_ALLOW_ALL_ORIGINS = True
```

Steps to Verify

1. Update `settings.py` with Correct Configuration:

Ensure your `settings.py` file includes the correct CORS settings as shown above.

2. Restart Django Server:

Restart your Django server to apply the changes:

bash

```
python manage.py runserver
```

3. Run React App:

Start your React app:

```
bash
```

```
npm start
```

4. Open Browser and Verify:

Open your browser and navigate to `http://localhost:3000`. The network request should now be allowed, and you should see the data fetched from your Django backend.

By ensuring the correct spelling of `CORS_ALLOWED_ORIGINS`, you will enable your Django backend to accept requests from your React frontend, resolving the CORS policy issue. If you encounter any further issues, please provide additional details for further assistance.

ChatGPT can make mistakes. Check important info.