

Getting Started with Create React App

This project was bootstrapped with [Create React App](#).

Available Scripts

In the project directory, you can run:

`npm start`

Runs the app in the development mode.

Open <http://localhost:3000> to view it in the browser.

The page will reload if you make edits.

You will also see any lint errors in the console.

`npm test`

Launches the test runner in the interactive watch mode.

See the section about [running tests](#) for more information.

`npm run build`

Builds the app for production to the `build` folder.

It correctly bundles React in production mode and optimizes the build for the best performance.

The build is minified and the filenames include the hashes.

Your app is ready to be deployed!

See the section about [deployment](#) for more information.

`npm run eject`

Note: this is a one-way operation. Once you `eject`, you can't go back!

If you aren't satisfied with the build tool and configuration choices, you can `eject` at any time. This command will remove the single build dependency from your project.

Instead, it will copy all the configuration files and the transitive dependencies (webpack, Babel, ESLint, etc) right into your project so you have full control over them. All of the commands except `eject` will still work, but they will point to the copied scripts so you can tweak them. At this point you're on your own.

You don't have to ever use `eject`. The curated feature set is suitable for small and middle deployments, and you shouldn't feel obligated to use this feature. However we understand that this tool wouldn't be useful if you couldn't customize it when you are ready for it.

Learn More

You can learn more in the [Create React App documentation](#).

To learn React, check out the [React documentation](#).

Overview

- <https://beautyx.vn> is a web app using reactjs 17x library. Runs on multiple platforms: web, momo miniapp, tiki miniapp

1. Momo mini app

- After, you should register an account business, developer at <https://developers.momoapp.vn/>
- Manually Config WebApp Links (For Web App): When you want to test any URLs of a Webapp but do not want to deploy them on the MoMo App Platform, you can input the Webapp URLs into the Devtool to open the test Mini App. For more details, follow the Webapp Config section in the <https://developers.momo.vn/v3/docs/app-center/development-guideline/development-tools/mini-dev-tools/mini-app-web/> document
- Mini app API (API Specifications)
 - Use this API to get user information from Momo App to login BeautyX mini app (required: Beautyx should be run on Momo mini app)
 - Installation & Packages dependencies:
 - Installation: To be able to install Mini API, please follow the document to set the [NPM configuration](#)
 - After install package:
 - `momo-miniapp update-packages`
 - `npm install @momo-miniapp/api`
- USAGE
 - Example

```
import MiniApi from "@momo-miniapp/api";
```

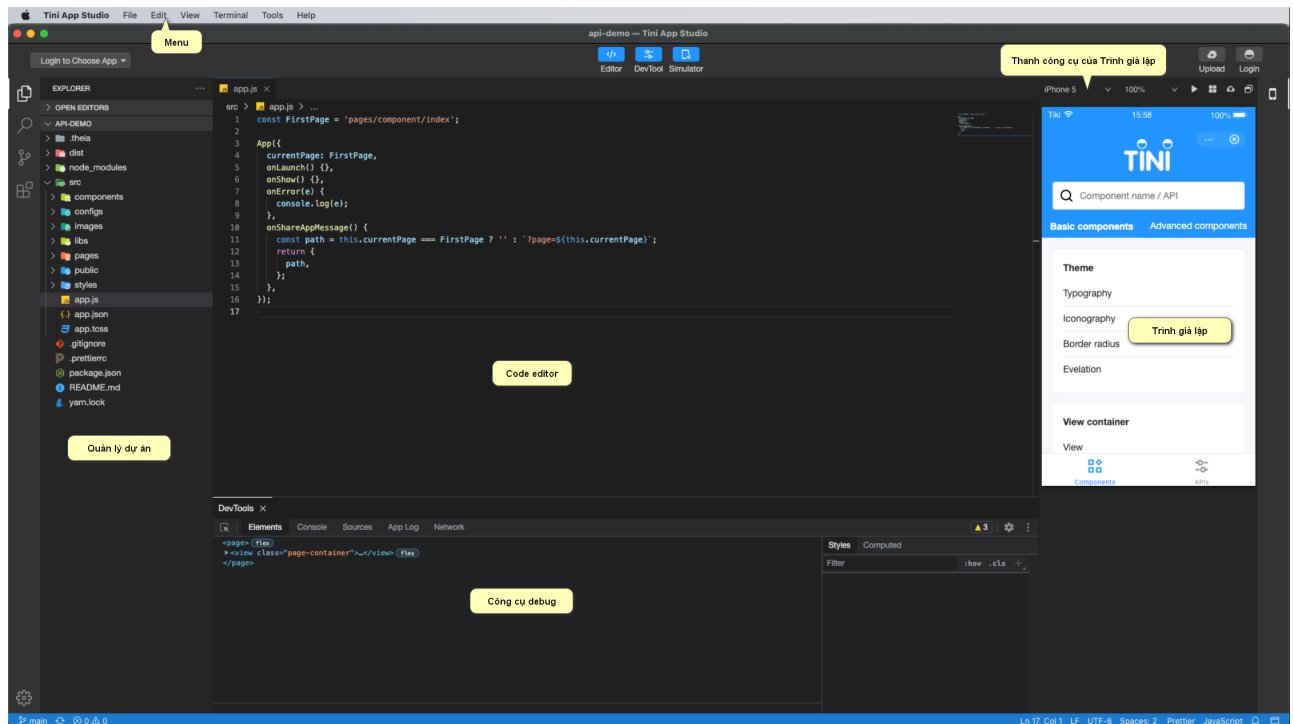
```
const onLoginFlatFormMomo = async () => {
  try {
    momoApi.initApp();
    MiniApi.showLoading([""]);
    MiniApi.getUserConsents({
      "permissions": [
        {
          "role": "name",
        },
      ],
    })
  }
}
```

```
        "role": "phone"
      },
      {
        "role": "email",
      },
    ]
  }, async ({ data, status }: any) => {
    const data0b: IUserConsentsData = {
      email: data?.email,
      name: data?.name,
      phone: data?.phone
    }
    if (data0b.phone) {
      //handle login api beautyx web app
    }
    else {
      MiniApi.showToast({
        description: "có lỗi khi nhận thông tin từ momo",
        type: "failure",
        duration: 3000,
      });
      MiniApi.hideLoading();
    }
    return { data: data }
  })
} catch (err) {
  console.log(err);
}
};
```

- View detail at [API mini app docs](#)

2.Tiki mini app

- Get user information (fullname, email, telephone) from Tiki app to login Beautyx web app
- Web app is run on the Tini framework platform. View doc Tini app at [Tini](#)
- DevTools Tini studio, download at [Tini studio](#)



- Tiki mini app APIs overview [link](#) (Tini App provides a number of APIs that when you want to use it, you will need to ask for permission [Tini console](#))
- List of roles rights in beautyx miniapp
 - `getAuthCode`: This permission is only used for internal Tiki apps. With this permission, the app does not need to ask users again when calling the APIs `my.getAuthCode`, `my.getUserInfo`. Example usage on tini framework
 -

```
onAuthen() {
  my.getAuthCode({
    scope: "user_profile",
    success: (res) => {
      //
    },
    fail: (res) => {
      //
    },
  });
},
```

- `getUserInfo`: This permission is used to get user information from Tiki app
- Setup domain [BeautyX Web app](#) to run on Tini app
 - Use component webview of Tini framework. Example:
 -

```
<web-view src={{`yourdomain`}} id="web-view1"
onMessage="handleOnMessage" />
```

- Communication between web view and Tini app
 - On Tini app: post message to Web view. Example
 -

```
handleOnMessage(e){
  const { requestId, params } = e.detail;
  const webview = my.createWebViewContext("web-view1");
  webview.postMessage({
    requestId,
    result: {
      //data send to webview
    },
  });
}
```

- On your Web app: get message to Tini app
-

```
function useGetMessageTiki() {
  const [response, setResponse] = useState<any>();
  const callbacks: any = {};
  useEffect(() => {
    let mount = true
    if (mount) {
      window.addEventListener('message', (e) => {
        const { requestId, result } = e.data;
        if (requestId === undefined) {
          return setResponse(null);
        }
        delete callbacks[requestId];
        setResponse({
          requestId: requestId,
          result: result
        });
      })
    }
    return () => { mount = false }
  }, [])
  return response;
}
```