**Hướng dẫn Cài đặt và Phát triển Zalo Mini App**

**1. Giới thiệu về Zalo Mini App**

Zalo Mini App là nền tảng cho phép các nhà phát triển xây dựng ứng dụng nhỏ gọn chạy trong ứng dụng Zalo, mang lại trải nghiệm liền mạch cho người dùng mà không cần cài đặt ứng dụng riêng biệt.

**2. Yêu cầu hệ thống**

**Phần mềm cần thiết:**

* **Node.js** (phiên bản 14.x trở lên)
* **npm** hoặc **yarn**
* **Zalo Mini App Studio** (IDE chính thức)
* **Git** (để quản lý source code)
* **Visual Studio Code** (khuyến nghị)

**Tài khoản cần thiết:**

* Tài khoản Zalo Developer
* Tài khoản Zalo để test

**3. Cài đặt môi trường phát triển**

**Bước 1: Cài đặt Node.js**

# Kiểm tra phiên bản Node.js

node --version

# Kiểm tra phiên bản npm

npm --version

**Bước 2: Tải và cài đặt Zalo Mini App Studio**

1. Truy cập: https://mini.zalo.me/studio
2. Tải xuống phiên bản phù hợp với hệ điều hành
3. Cài đặt theo hướng dẫn

**Bước 3: Đăng ký tài khoản Developer**

1. Truy cập: https://developers.zalo.me/
2. Đăng ký/Đăng nhập bằng tài khoản Zalo
3. Tạo ứng dụng mới và lấy App ID

**4. Tạo dự án mới**

**Sử dụng Zalo Mini App Studio:**

1. **Mở Zalo Mini App Studio**
2. **Chọn "Create New Project"**
3. **Điền thông tin dự án:**
   * Project Name: Tên dự án
   * App ID: ID từ Zalo Developer Console
   * Template: Chọn template phù hợp

**Sử dụng CLI (Command Line):**

# Cài đặt Zalo Mini App CLI

npm install -g zmp-cli

# Tạo dự án mới

zmp create my-mini-app

# Di chuyển vào thư mục dự án

cd my-mini-app

# Cài đặt dependencies

npm install

**5. Cấu trúc dự án**

my-mini-app/

├── src/

│ ├── components/ # Các component tái sử dụng

│ ├── pages/ # Các trang của app

│ │ ├── index/

│ │ │ ├── index.jsx

│ │ │ └── index.scss

│ │ └── profile/

│ ├── services/ # API services

│ ├── utils/ # Utility functions

│ └── app.jsx # Component gốc

├── public/

│ └── assets/ # Hình ảnh, icons

├── app.json # Cấu hình app

├── package.json

└── README.md

**6. Cấu hình cơ bản**

**File app.json:**

{

"app": {

"title": "My Mini App",

"headerTitle": "Ứng dụng của tôi",

"headerColor": "#1877F2",

"textColor": "white",

"statusBarColor": "#1877F2"

},

"pages": [

"pages/index/index",

"pages/profile/index"

],

"tabBar": {

"list": [

{

"pagePath": "pages/index/index",

"text": "Trang chủ",

"iconPath": "assets/home.png",

"selectedIconPath": "assets/home-active.png"

},

{

"pagePath": "pages/profile/index",

"text": "Hồ sơ",

"iconPath": "assets/profile.png",

"selectedIconPath": "assets/profile-active.png"

}

]

}

}

**7. Phát triển ứng dụng mẫu**

**Trang chủ (src/pages/index/index.jsx):**

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

import { Page, Text, Button, Box, Header } from 'zmp-ui';

import { getUserInfo, showToast } from 'zmp-sdk';

const HomePage = () => {

const [userInfo, setUserInfo] = useState(null);

useEffect(() => {

// Lấy thông tin người dùng

getUserInfo({

success: (data) => {

setUserInfo(data.userInfo);

},

fail: (error) => {

console.error('Lỗi lấy thông tin user:', error);

}

});

}, []);

const handleButtonClick = () => {

showToast({

message: 'Xin chào từ Zalo Mini App!',

type: 'success'

});

};

return (

<Page>

<Header title="Trang chủ" />

<Box p={4}>

<Text size="large" className="welcome-text">

Chào mừng đến với Mini App!

</Text>

{userInfo && (

<Box mt={4}>

<Text>Xin chào, {userInfo.name}!</Text>

</Box>

)}

<Button

variant="primary"

fullWidth

onClick={handleButtonClick}

className="mt-4"

>

Nhấn để chào hỏi

</Button>

</Box>

</Page>

);

};

export default HomePage;

**Component App chính (src/app.jsx):**

import React from 'react';

import { App, ZMPRouter, AnimationRoutes, SnackbarProvider } from 'zmp-ui';

import { RecoilRoot } from 'recoil';

import HomePage from './pages/index';

import ProfilePage from './pages/profile';

const MyApp = () => {

return (

<RecoilRoot>

<App>

<SnackbarProvider>

<ZMPRouter>

<AnimationRoutes>

<Route path="/" component={HomePage} />

<Route path="/profile" component={ProfilePage} />

</AnimationRoutes>

</ZMPRouter>

</SnackbarProvider>

</App>

</RecoilRoot>

);

};

export default MyApp;

**8. Sử dụng Zalo APIs**

**Lấy thông tin người dùng:**

import { getUserInfo } from 'zmp-sdk';

const fetchUserInfo = () => {

getUserInfo({

success: (data) => {

console.log('User info:', data.userInfo);

},

fail: (error) => {

console.error('Error:', error);

}

});

};

**Chia sẻ nội dung:**

import { shareMessage } from 'zmp-sdk';

const shareContent = () => {

shareMessage({

message: 'Hãy thử Mini App này!',

url: 'https://zalo.me/s/123456789',

success: () => {

console.log('Chia sẻ thành công');

},

fail: (error) => {

console.error('Lỗi chia sẻ:', error);

}

});

};

**Gọi API backend:**

import { request } from 'zmp-sdk';

const callAPI = async () => {

try {

const response = await request({

url: 'https://api.example.com/data',

method: 'GET',

header: {

'Content-Type': 'application/json'

}

});

console.log('API Response:', response.data);

} catch (error) {

console.error('API Error:', error);

}

};

**9. Quản lý trạng thái với Recoil**

**Tạo atom state (src/state/user.js):**

import { atom } from 'recoil';

export const userState = atom({

key: 'userState',

default: {

id: null,

name: '',

avatar: '',

isLoggedIn: false

}

});

**Sử dụng trong component:**

import { useRecoilState } from 'recoil';

import { userState } from '../state/user';

const MyComponent = () => {

const [user, setUser] = useRecoilState(userState);

// Cập nhật user state

const updateUser = (userData) => {

setUser({

...user,

...userData,

isLoggedIn: true

});

};

return (

<div>

{user.isLoggedIn ? (

<p>Xin chào, {user.name}</p>

) : (

<p>Vui lòng đăng nhập</p>

)}

</div>

);

};

**10. Chạy và Debug ứng dụng**

**Chạy trong môi trường development:**

# Chạy dev server

npm start

# Hoặc

yarn start

**Debug trên điện thoại:**

1. Mở Zalo trên điện thoại
2. Vào **Cá nhân** > **Tiện ích** > **Mini App Studio**
3. Scan QR code từ dev server
4. Test ứng dụng trực tiếp

**Debug trên Simulator:**

1. Mở Zalo Mini App Studio
2. Chọn **Preview**
3. Test các tính năng trong simulator

**11. Build và Deploy**

**Build production:**

npm run build

**Upload lên Zalo:**

1. Mở Zalo Mini App Studio
2. Chọn **Upload**
3. Điền thông tin phiên bản
4. Chờ review từ Zalo

**12. Best Practices**

**Performance:**

* Sử dụng lazy loading cho images
* Minimize bundle size
* Cache API responses
* Optimize component rendering

**UI/UX:**

* Tuân thủ Zalo Design System
* Responsive design
* Loading states
* Error handling

**Security:**

* Validate user input
* Secure API calls
* Handle sensitive data properly
* Implement proper authentication

**13. Tài nguyên hữu ích**

**Documentation:**

* [Zalo Mini App Documentation](https://mini.zalo.me/docs)
* [Zalo Mini App UI Components](https://zmp-ui.vercel.app/)
* [Zalo Developer Portal](https://developers.zalo.me/)

**Community:**

* [Zalo Developer Facebook Group](https://claude.ai/chat/fbaa58ec-5128-44b5-9b7d-e83dc70cd1e6)
* [GitHub Examples](https://claude.ai/chat/fbaa58ec-5128-44b5-9b7d-e83dc70cd1e6)
* [Stack Overflow Tags](https://claude.ai/chat/fbaa58ec-5128-44b5-9b7d-e83dc70cd1e6)

**14. Troubleshooting**

**Lỗi thường gặp:**

**1. App ID không hợp lệ:**

Solution: Kiểm tra App ID trong app.json và đảm bảo đã tạo app trên Zalo Developer

**2. API calls bị chặn:**

Solution: Kiểm tra domain whitelist trong Zalo Developer Console

**3. Build lỗi:**

# Clear cache và reinstall

rm -rf node\_modules package-lock.json

npm install

**4. Preview không hoạt động:**

Solution: Đảm bảo Zalo app trên điện thoại đã cập nhật phiên bản mới nhất

**15. Kết luận**

Zalo Mini App cung cấp một nền tảng mạnh mẽ để phát triển ứng dụng tích hợp sâu với hệ sinh thái Zalo. Với hướng dẫn này, bạn đã có thể bắt đầu xây dựng ứng dụng Mini App đầu tiên của mình.

Hãy tiếp tục khám phá các tính năng nâng cao và tham gia cộng đồng developer để học hỏi thêm kinh nghiệm từ các nhà phát triển khác.