

Navigation

Chỉ có ở App

Không có URL

Dùng stack để quản lý

=====

Stack = Push (Sang màn hình mới)+ Pop(trở về màn hình cũ)

Ví dụ: Tạo màn hình mới:

```
navigation.push('Product');
```

```
navigation.pop() = navigation.goBack();
```

=====

Truyền dữ liệu giữa các màn hình:

```
navigation.navigate('Product',{
```

```
  id: 1,
```

```
  name: iphone 17,
```

```
});
```

Truyền Object

```
navigation.navigate('Product',{
```

```
  product:{
```

```
    id:1,
```

```
    name: iphone 17,
```

```
  },
```

```
});
```

=====

Ví dụ navigation trong bài toán bán hàng:

Splash

Login

Home

Product Detail

Cart

Checkout

=====

Sử dụng props trong navigation

function HomeScreen({navigation, route}){}
=====

Call API và luồng dữ liệu

Mobile <=> server

URL + Method+Request+Response

=> Mobile không truy cập trực tiếp vào Database mà
đoạn qua API (Nodejs, php, java, c#, python)

=> App -> API -> Database

=====

Đọc dữ liệu từ Server:

C1:

```
fetch('https://abc.com/products')  
.then(res=> res.json())  
.then(data => console.log(data))
```

Nhược điểm:

- Không timeout + không intercept

C2: axios

```
axios.get('/  
products').then(res=>console.log(res.data));
```

Ưu điểm:

Parse JSON + tự động reject lỗi http

=====

Luồng chuẩn: Call API -> Loading -> Success/error -> render

Chú ý: thời điểm mount dữ liệu

sử dụng useEffect

=>

try {

 setLoading(true);

 const res = await api.get('/product')

 setProducts(res.data)

}

catch(e){

 setError(e.message);

}

finally {

 setLoading(false);

}

=====

Status code:

200: ok

500: server error

400: Bad request

401: unauthorized

403: Forbidden

404: Notfound

=====