



**FPT POLYTECHNIC**



**THỰC HỌC – THỰC NGHIỆP**



**Conceive Design Implement Operate**

## **LẬP TRÌNH FRONT-END FRAMEWORK 2**

---

**REDUX-FORM**

**REST-BASED APP REACT**

- ⦿ Redux-form
- ⦿ Rest-based app react





# PHẦN 1

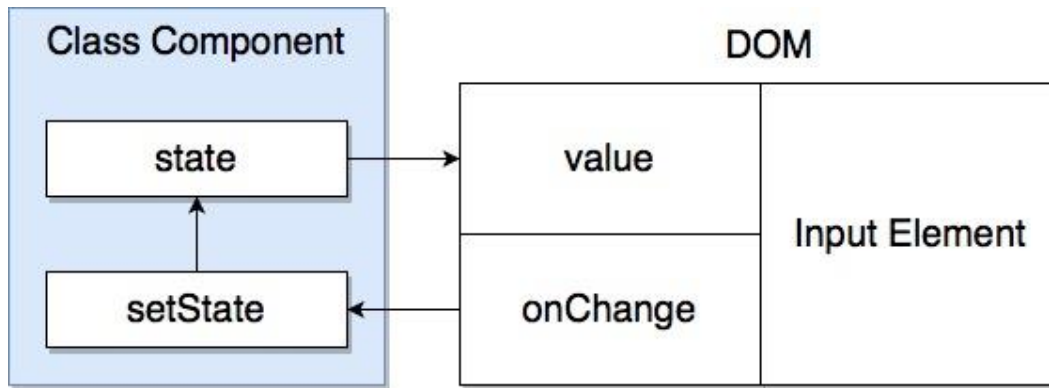
## REDUX FORM

- ❑ Redux-form là thư viện hỗ trợ quản lý react form state
- ❑ Cài đặt

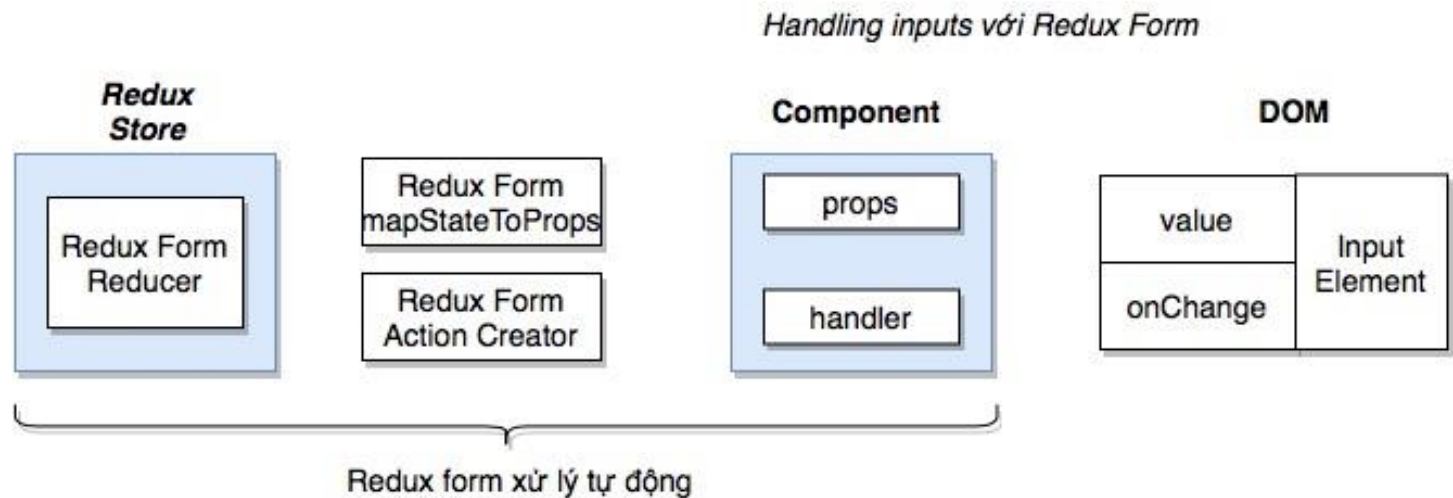
```
→ client git:(210-userid) x npm install --save redux-form
+ redux-form@7.4.2
added 3 packages from 4 contributors and audited 35724 packages in 10.155s
found 0 vulnerabilities

→ client git:(210-userid) x npm start

> client@0.1.0 start /Users/stephengrider/workspace/ReduxWorkspaceV2/prod/s
treameams/client
> react-scripts start
```

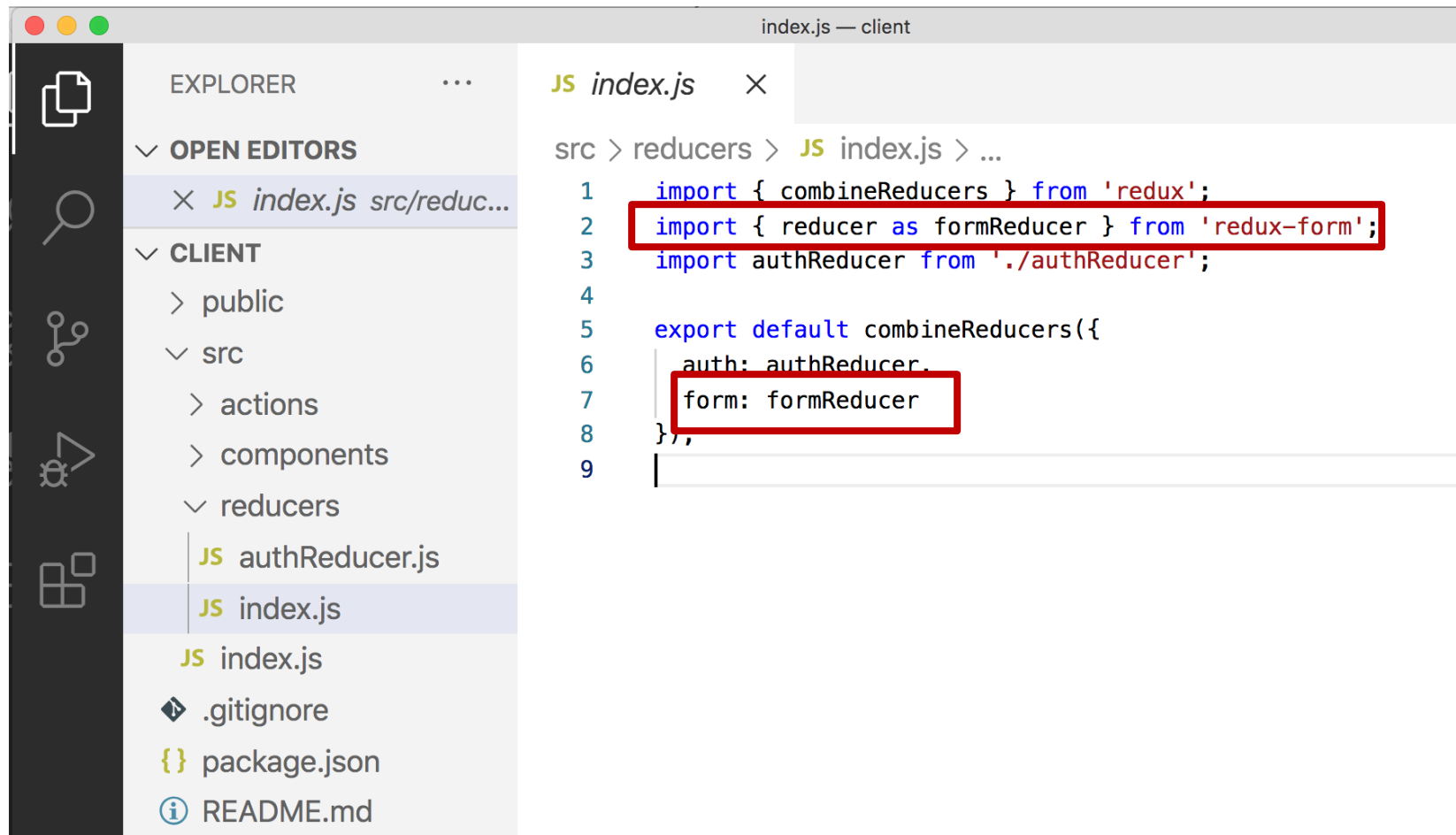


*Handling inputs không dùng Redux*



***formReducer***: function chịu trách nhiệm cập nhật redux store dựa trên những thay đổi từ app

❑ Import {reducer as formReducer } from 'redux-form'



The screenshot shows a VS Code editor window titled 'index.js — client'. The Explorer sidebar on the left shows the project structure with 'src' expanded, and 'reducers' expanded, with 'index.js' selected. The main editor area shows the code in 'src > reducers > JS index.js > ...'. The code is as follows:

```
1 import { combineReducers } from 'redux';
2 import { reducer as formReducer } from 'redux-form';
3 import authReducer from './authReducer';
4
5 export default combineReducers({
6   auth: authReducer,
7   form: formReducer
8 },
9
```

Red boxes highlight the import statement on line 2 and the 'form: formReducer' entry in the combineReducers object on line 7.

JS StreamCreate.js •

```
1 import React from 'react';
2 import { Field, reduxForm } from 'redux-form';
3
4 class StreamCreate extends React.Component {
5   render() {
6     return (
7       <form>
8         <Field name="title" />
9         <Field name="description" />
10      </form>
11    );
12  }
13 }
14
15 export default reduxForm({
16   form: 'streamCreate'
17 })(StreamCreate);
```

□ Component `<Field />`

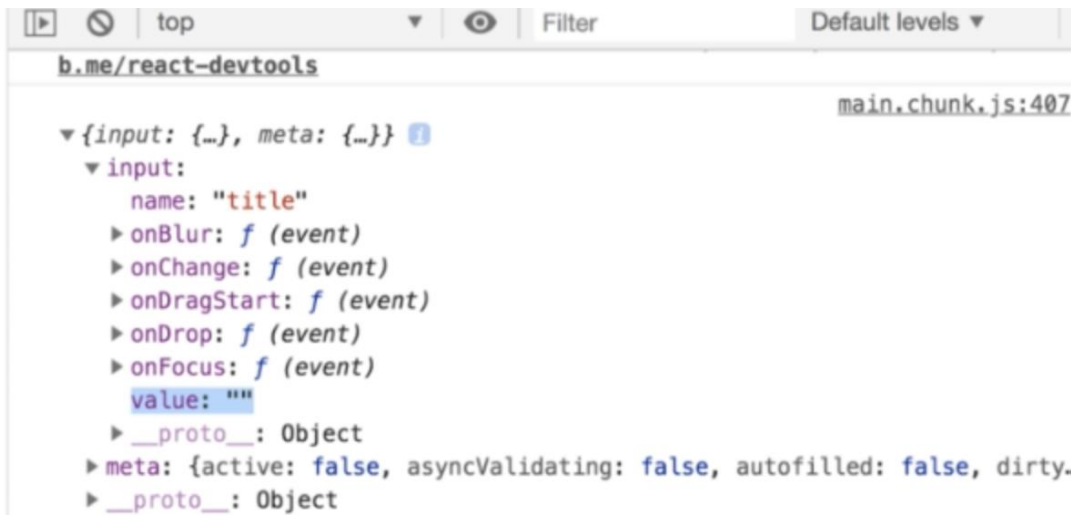
`<Field name="inputName"  
component="input" type="text" />`

- Kết nối input đến store
- Tạo phần tử HTML `<input />` với type text
- Truyền vào props: **value**, **onChange**, **onBlur**,...giúp theo dõi và cập nhật state
- Các loại input khác trong `<Field />`:  
<https://redux-form.com/7.3.0/docs/api/field.md/#usage>



```
renderInput(formProps) {
  return (
    <input
      onChange={formProps.input.onChange}
      value={formProps.input.value}
    />
  );
}

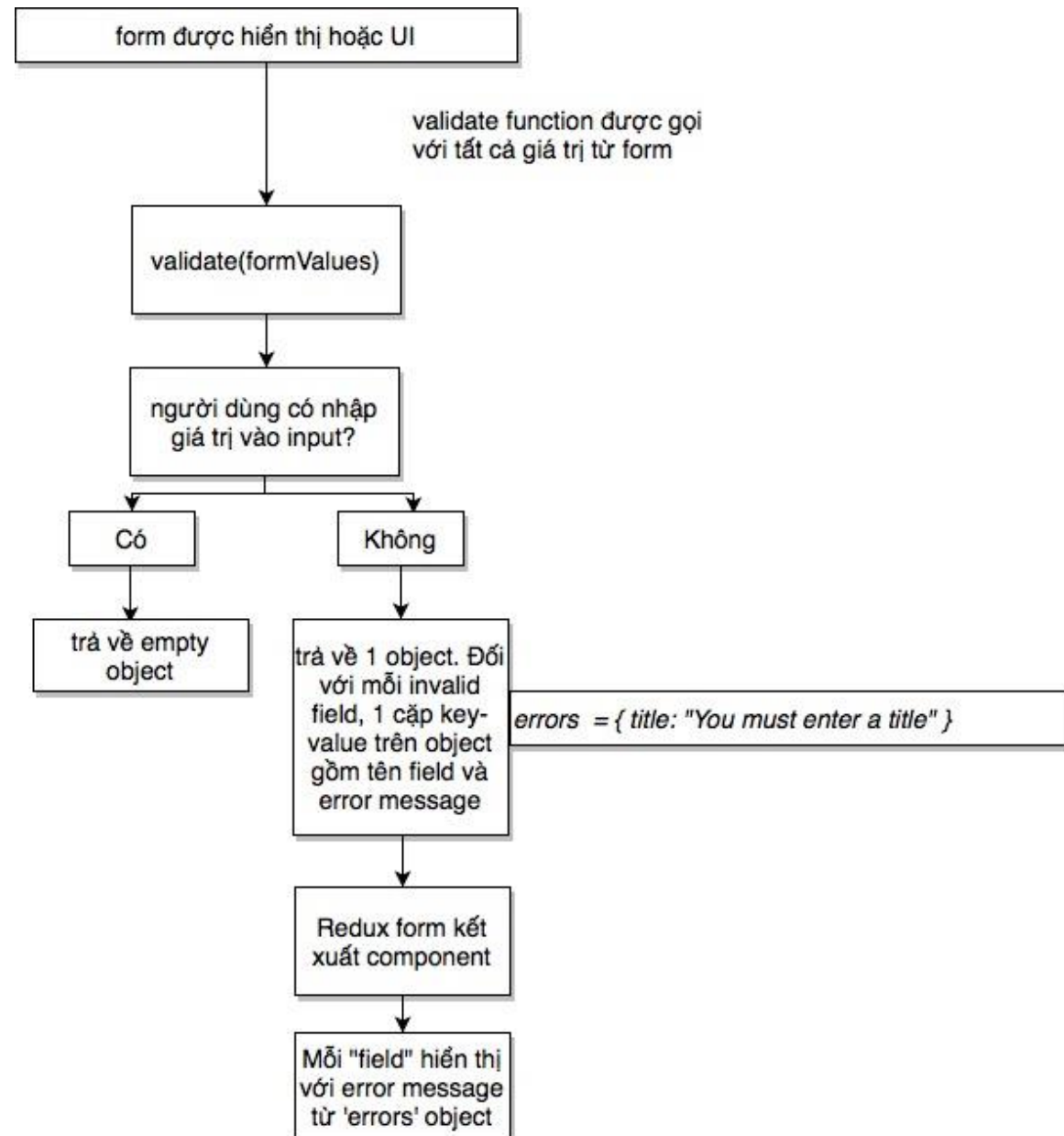
render() {
  return (
    <form>
      <Field name="title" component={this.renderInput} />
      <Field name="description" component={this.renderInput} />
    </form>
  );
}
```



```
onSubmit(formValues) {  
  console.log(formValues);  
}
```

*Dùng arrow function*

```
render() {  
  return (  
    <form  
      onSubmit={this.props.handleSubmit(this.onSubmit)}  
      className="ui form">  
      <  
        <Field name="title" component={this.renderInput} label="Enter Title" />  
        <Field  
          name="description"  
          component={this.renderInput}  
          label="Enter Description"  
        />  
        <button className="ui button primary">Submit</button>  
      </form>  
    );  
  }  
}
```



```
const validate = formValues => {  
  const errors = {};  
  
  if (!formValues.title) {  
    errors.title = 'You must enter a title';  
  }  
  
  if (!formValues.description) {  
    errors.description = 'You must enter a description';  
  }  
  
  return errors;  
};  
  
export default reduxForm({  
  form: 'streamCreate',  
  validate  
})(StreamCreate);
```

```
renderInput({ input, label, meta }) {  
  return (  
    <div className="field">  
      <label>{label}</label>  
      <input {...input} />  
      <div>{meta.error}</div>  
    </div>  
  );  
}
```



**STREAMS APP**

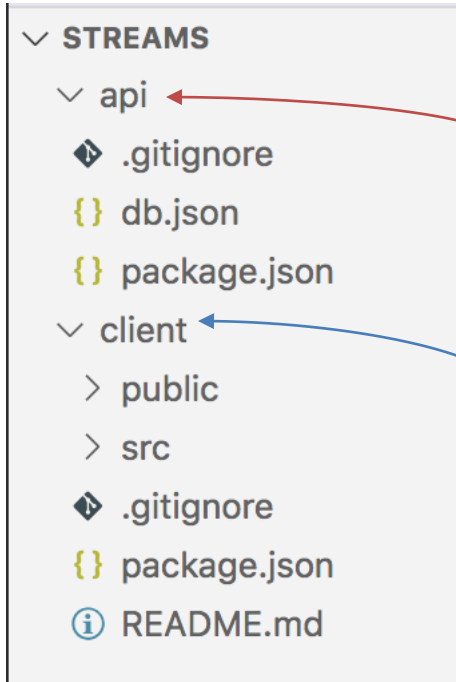


## PHẦN 2

# REST-BASED APP REACT

## □ Structure

### ❖ Streams project



Server

React app




```
→ api git:(223-highlighting) ✕ ls
package.json
→ api git:(223-highlighting) ✕ npm install --save json-server
npm notice created a lockfile as package-lock.json. You should commit this
file.
npm WARN api@1.0.0 No description
npm WARN api@1.0.0 No repository field.

+ json-server@0.14.0
added 223 packages from 121 contributors and audited 444 packages in 12.831
s
found 0 vulnerabilities

→ api git:(223-highlighting) ✕
```

## □ Db.json



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the file structure with 'db.json' selected. The main editor area shows the content of 'db.json' with line numbers 1 through 4. The JSON content is as follows:

```
1 {  
2   "streams": []  
3 }  
4
```

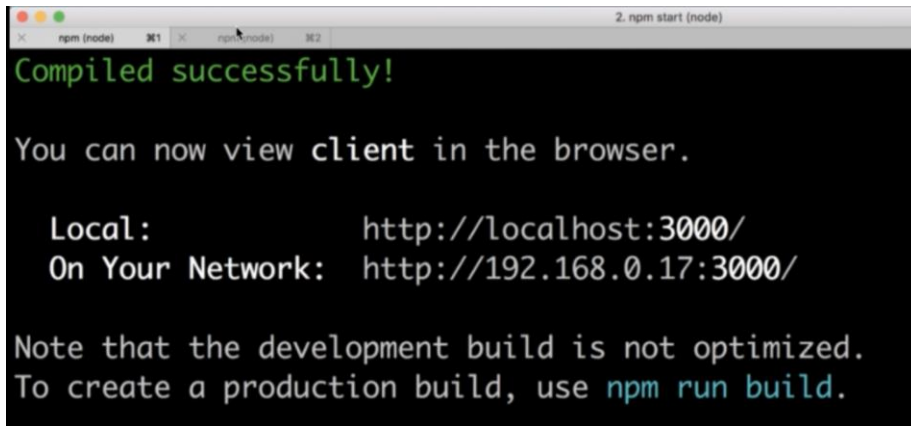
## □ Package.json



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a folder named 'API' containing files 'db.json', 'package-lock.json', and 'package.json'. The code editor displays the content of 'package.json' with the following JSON structure:

```
1 {
2   "name": "api",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "json-server -p 3001 -w db.json"
8   },
9   "author": "",
10  "license": "ISC",
11  "dependencies": {
12    "json-server": "^0.14.0"
13  }
14 }
15
```

## ❖ React app



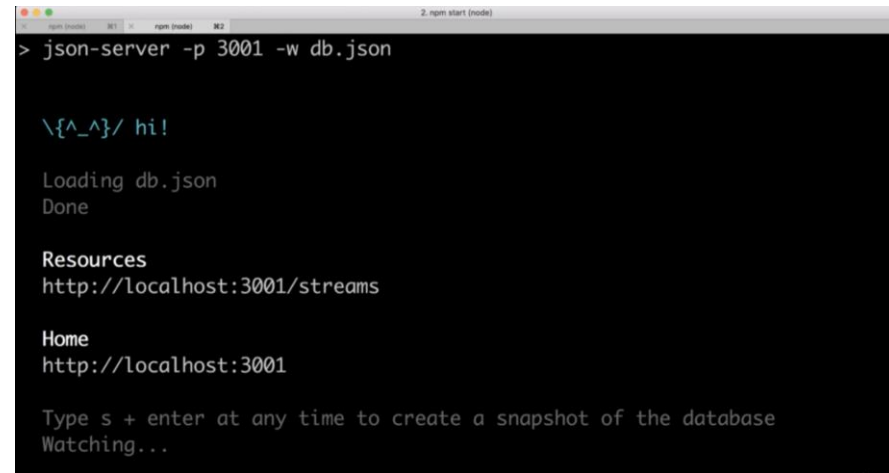
```
2. npm start (node)
Compiled successfully!

You can now view client in the browser.

Local:      http://localhost:3000/
On Your Network:  http://192.168.0.17:3000/

Note that the development build is not optimized.
To create a production build, use npm run build.
```

## ❖ Server



```
2. npm start (node)
> json-server -p 3001 -w db.json

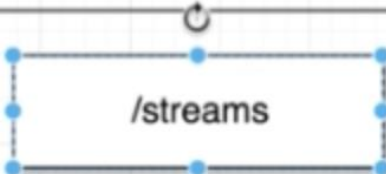
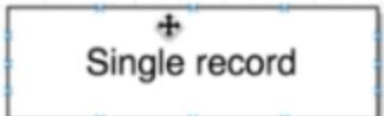
\{^_^\}/ hi!

Loading db.json
Done

Resources
http://localhost:3001/streams

Home
http://localhost:3001

Type s + enter at any time to create a snapshot of the database
Watching...
```

Action	Method	Route	Response
List all records	GET		Array of records
Get one particular record	GET	/streams/:id	Single record
Create record	POST	/streams	 Single record
Update a record	PUT	/streams/:id	Single record
Delete a record	DELETE	/streams/:id	Nothing

## ❑ Tạo record

client > src > actions > JS index.js > ...

```
16
17   export const createStream = (formValues) => async (dispatch) => {
18     const response = await streams.post('/streams', formValues);
19
20     dispatch({ type: CREATE_STREAM, payload: response.data });
21   };
22
```

## ❑ Tạo record với id

client > src > actions > JS index.js > ...

```
24
25   export const createStream = formValues => async (dispatch, getState) => {
26     const { userId } = getState().auth;
27     const response = await streams.post('/streams', { ...formValues, userId });
28
29     dispatch({ type: CREATE_STREAM, payload: response.data });
30   };
```

## □ Records

```
export const fetchStreams = () => async dispatch => {  
  const response = await streams.get('/streams');  
  
  dispatch({ type: FETCH_STREAMS, payload: response.data });  
};
```

## □ Record với id

```
export const fetchStream = id => async dispatch => {  
  const response = await streams.get(`/streams/${id}`);  
  
  dispatch({ type: FETCH_STREAM, payload: response.data });  
};
```

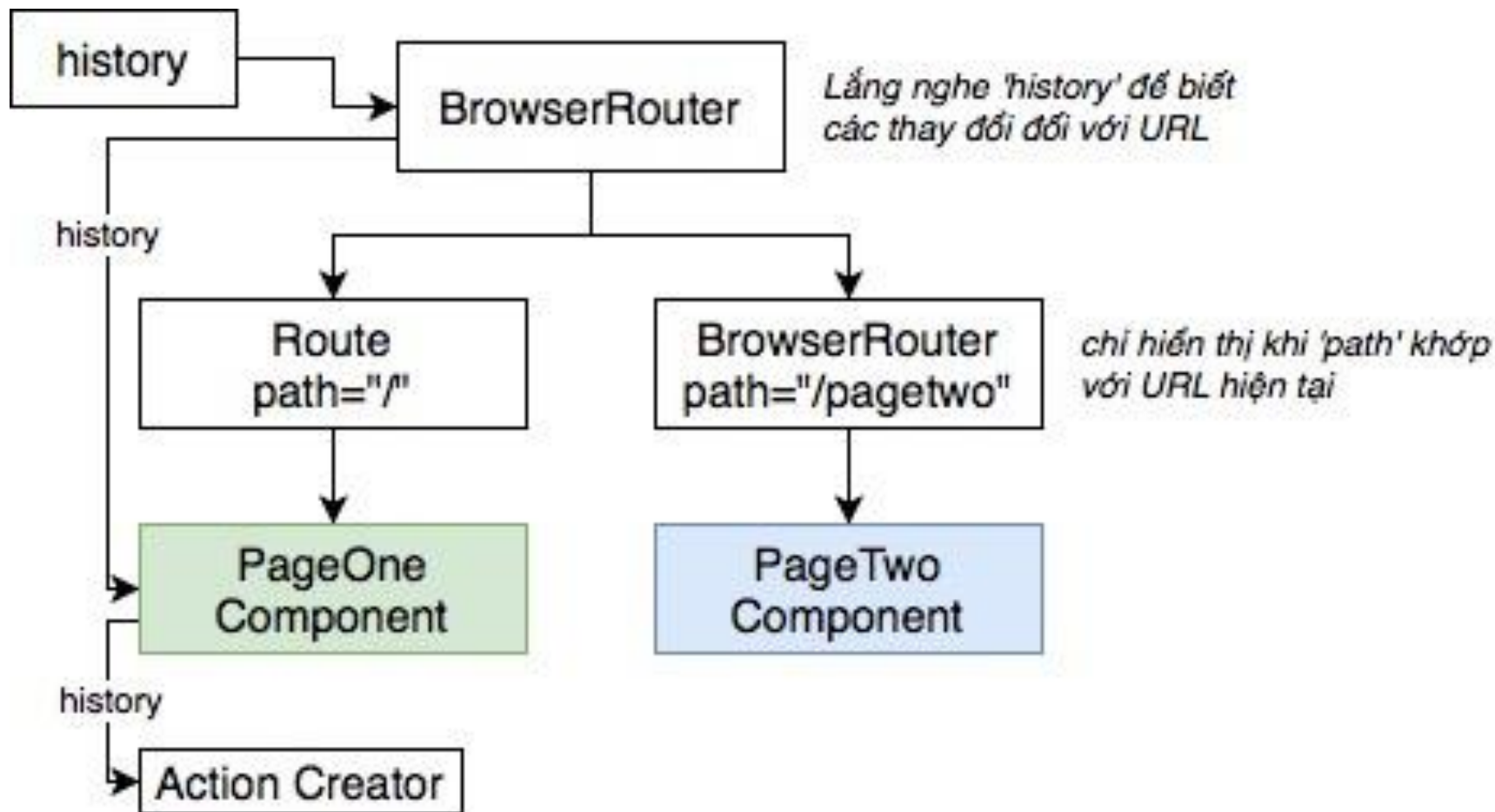
## □ Edit

```
export const editStream = (id, formValues) => async dispatch => {  
  const response = await streams.put(`/streams/${id}`, formValues);  
  
  dispatch({ type: EDIT_STREAM, payload: response.data });  
};
```

## □ Delete

```
export const deleteStream = id => async dispatch => {  
  await streams.delete(`/streams/${id}`);  
  
  dispatch({ type: DELETE_STREAM, payload: id });  
};
```



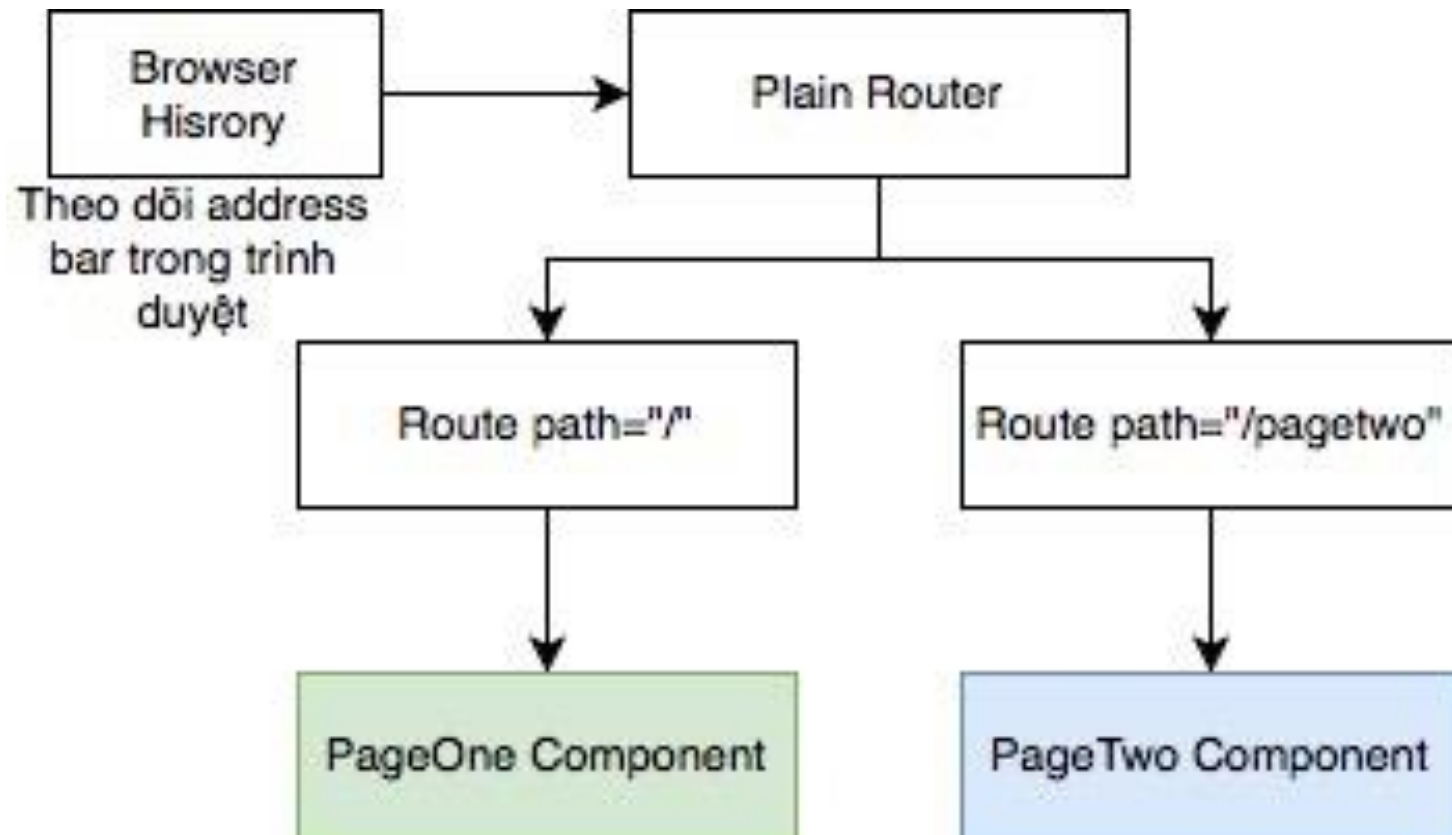


❑ Tham khảo:

<https://reactrouter.com/web/api/history>

❑ History object có những thuộc tính và phương thức quan trọng sau

- ❖ Location
- ❖ Push()
- ❖ Replace()
- ❖ Go()
- ❖ goBac()



demo

- ☑ Redux-form
- ☑ Rest-based app react



thank  
you!