



FPT POLYTECHNIC



THỰC HỌC – THỰC NGHIỆP



Conceive Design Implement Operate

LẬP TRÌNH FRONT-END FRAMEWORK 2

REACT ROUTER

AUTHENTICATION VỚI REACT

- ⦿ React router
- ⦿ Authentication với react





PHẦN 1

REACT ROUTER

react-router	Core navigation lib (không cài đặt gói này)
react-router-dom	<i>Navigation cho dom-based app</i>
react-router-native	Navigation cho react-native app
react-router-redux	Ràng buộc giữa Redux và react router (không cần thiết)

Cài đặt

```

2. npm install --save react-router-dom (node)
→ client git:(182-chal) ✖ npm install --save react-router-dom
(( )) ∴ rollbackFailedOptional: verb npm-session 5df804e98c47
  
```

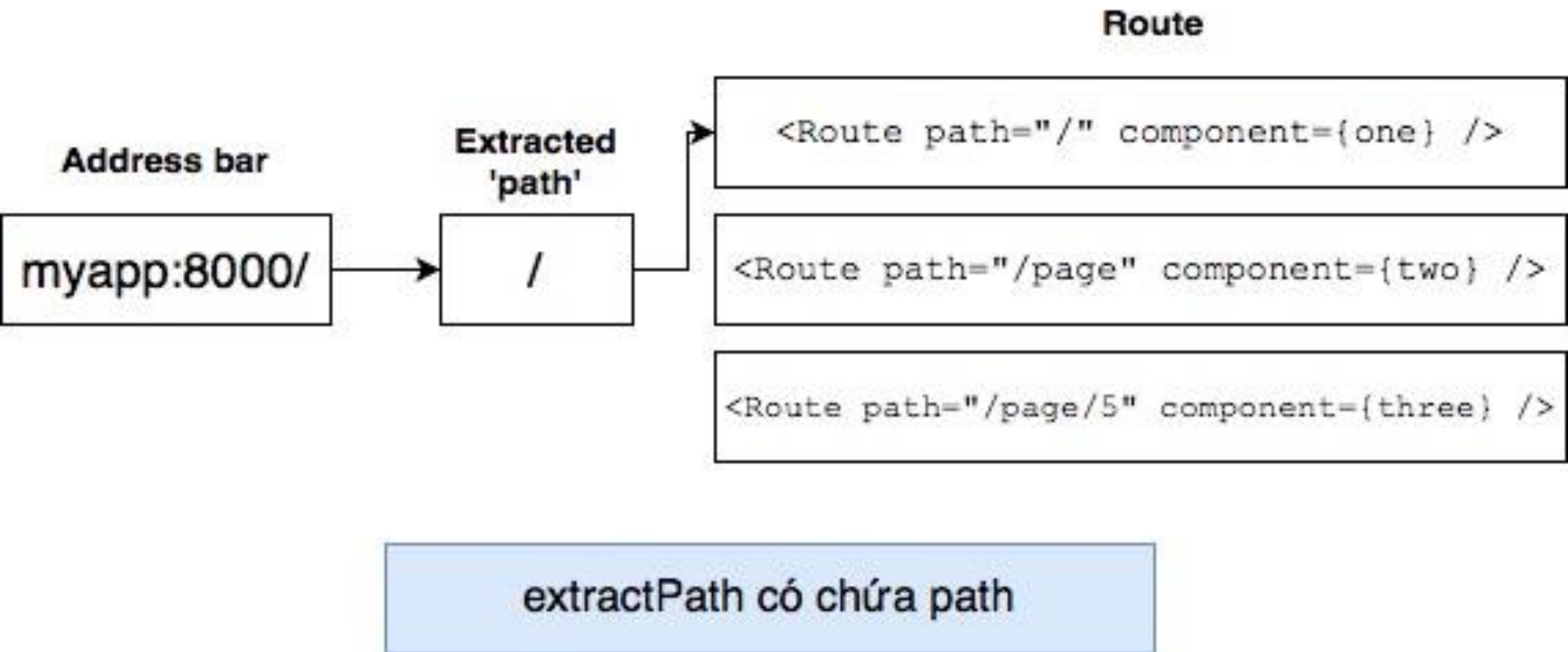
□ `<BrowserRouter basename>`

```
<BrowserRouter basename="/calendar">
  <Link to="/today"/> // renders <a href="/calendar/today">
  <Link to="/tomorrow"/> // renders <a href="/calendar/tomorrow">
  ...
</BrowserRouter>
```

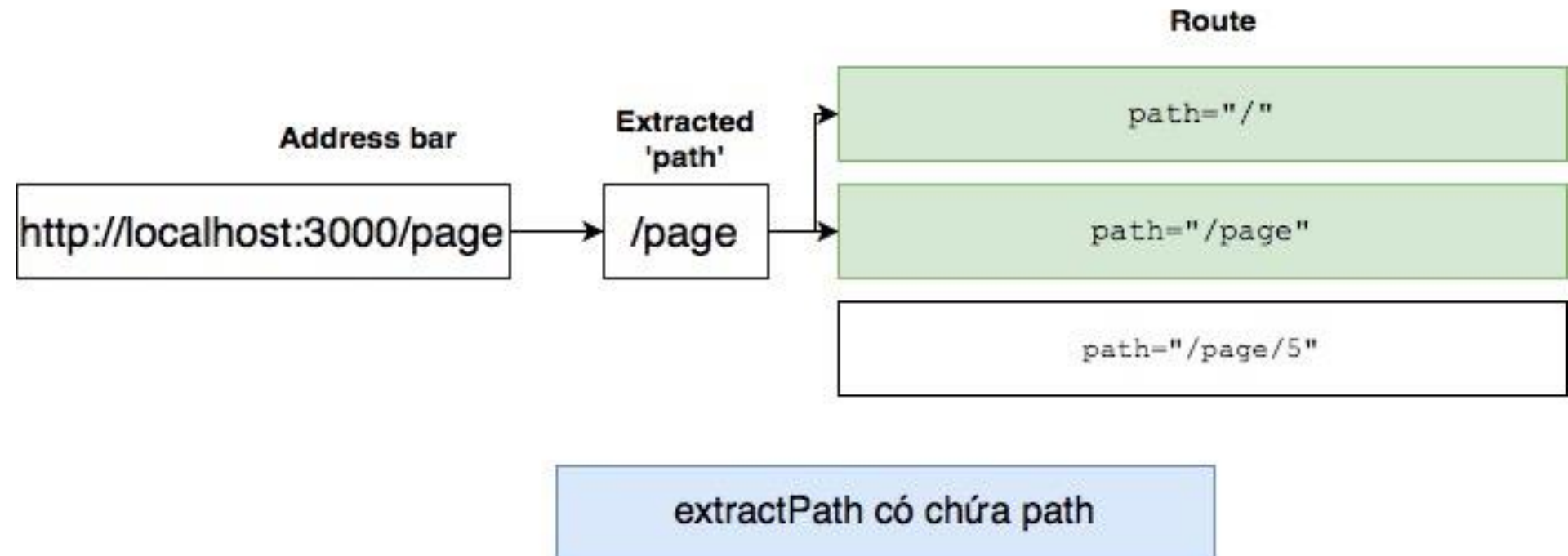
□ `<Route [exact] path component>`

- ❖ Path: string | string[]
- ❖ Component: react component được render khớp với path

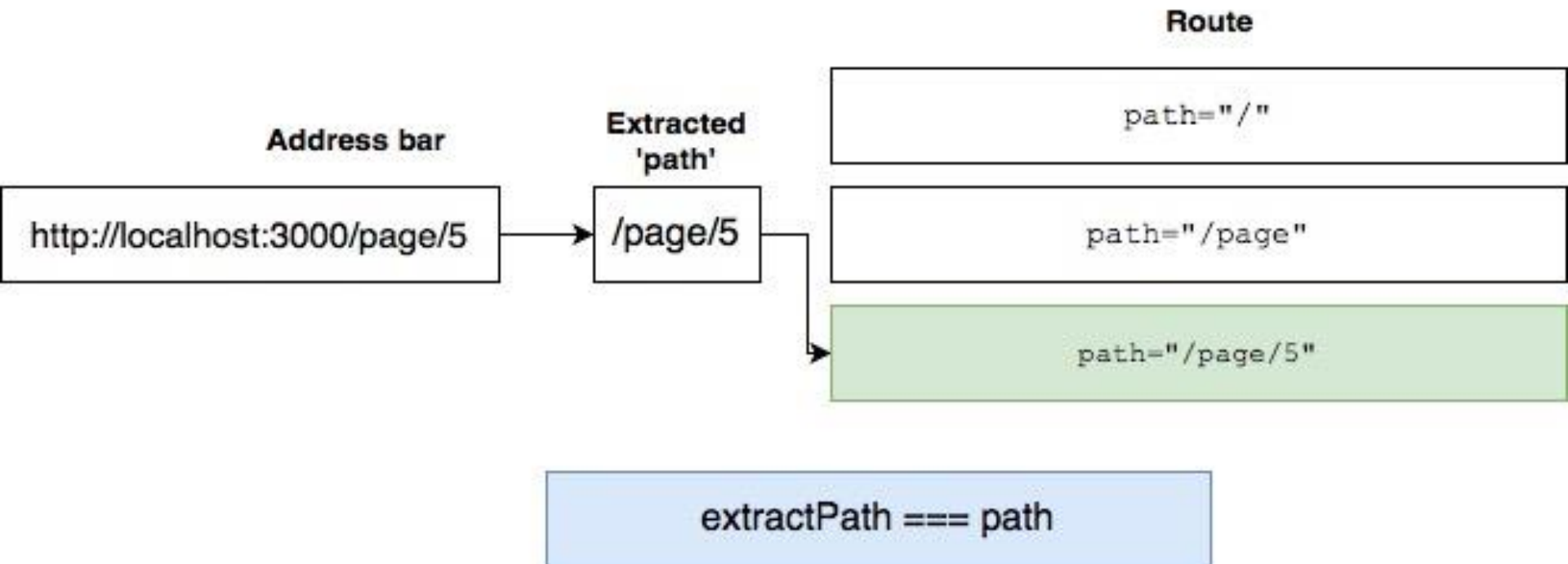
□ `exact={false}`



`exact={false}`



☐ `exact={true}`



□ `<Link to>`

Điều hướng (navigate).

Khai báo, truy xuất navigate trong ứng dụng

to: string: chuỗi liên kết, được tạo bằng cách nối tên đường dẫn (pathname), tìm kiếm và hash properties

```
<Link to="/about">About</Link>
```

```
<Link to="/courses?sort=name" />
```

mọi thứ sau TLD (.com,.net)
hoặc port là 'path'

BrowserRouter

localhost:3000/**pagetwo**

mọi thứ sau # là 'path'

HashRouter

localhost:3000/#/**pagetwo**

không dùng URL để theo dõi
điều hướng

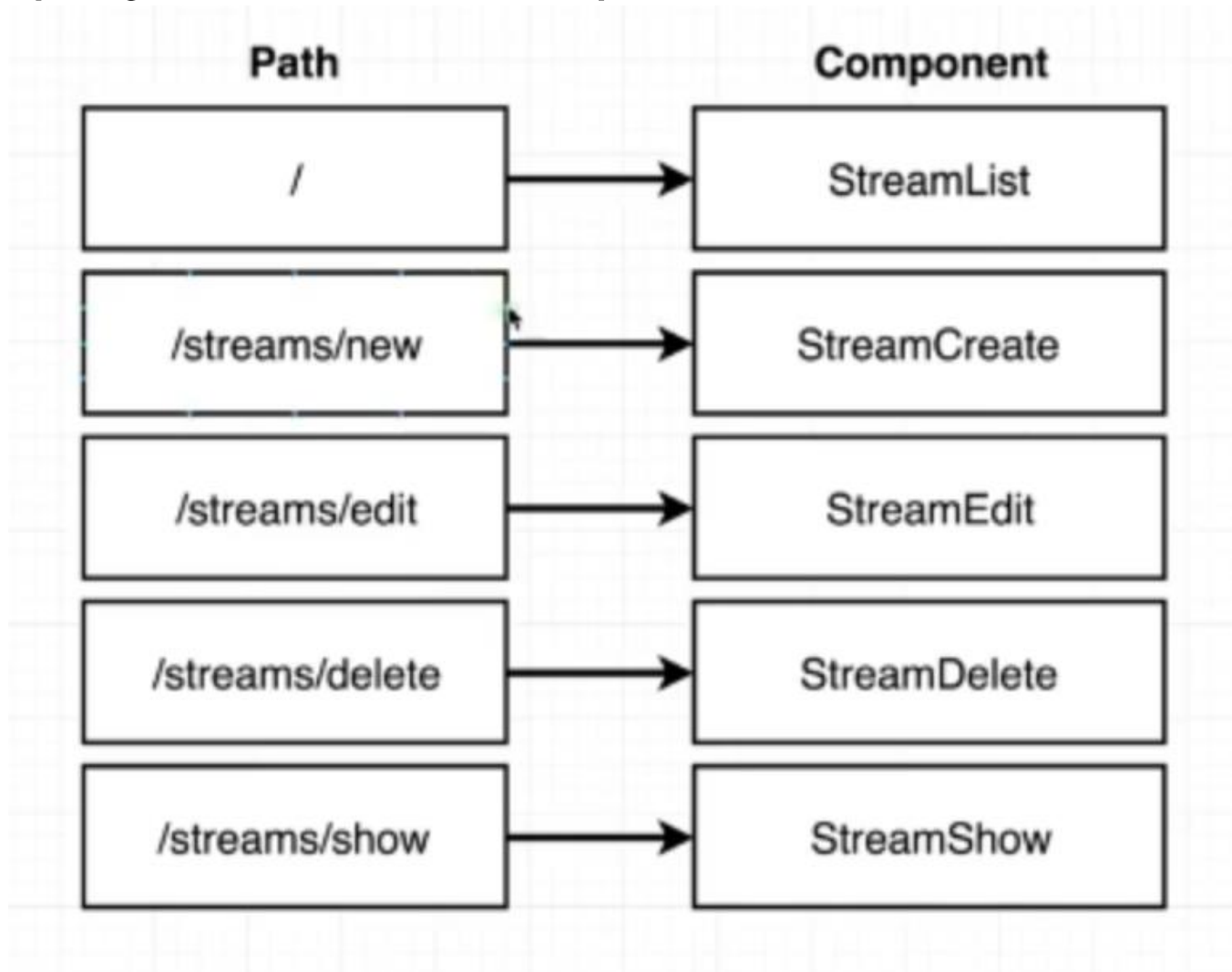
MemoryRouter

localhost:3000/

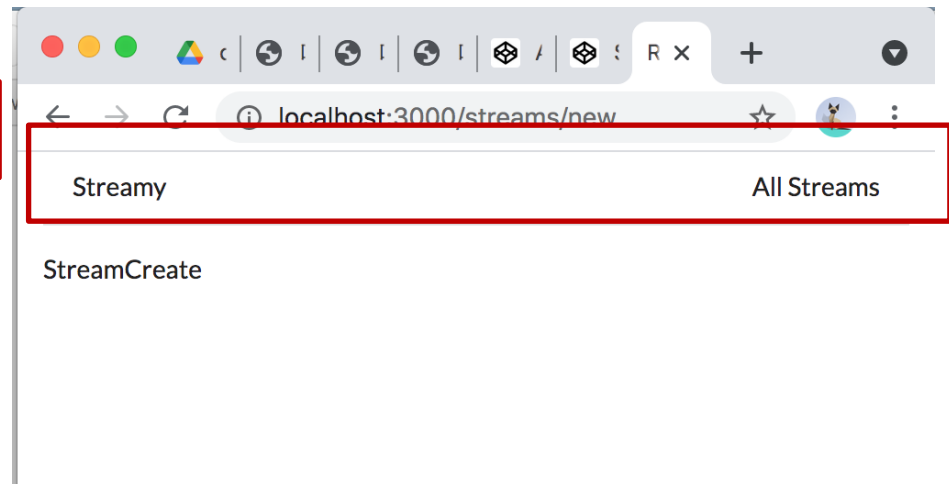
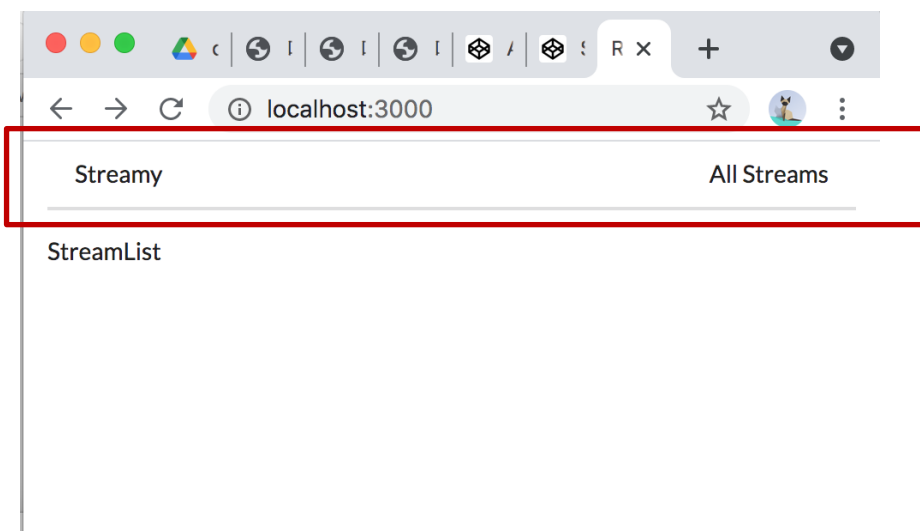


STREAMS PROJECT

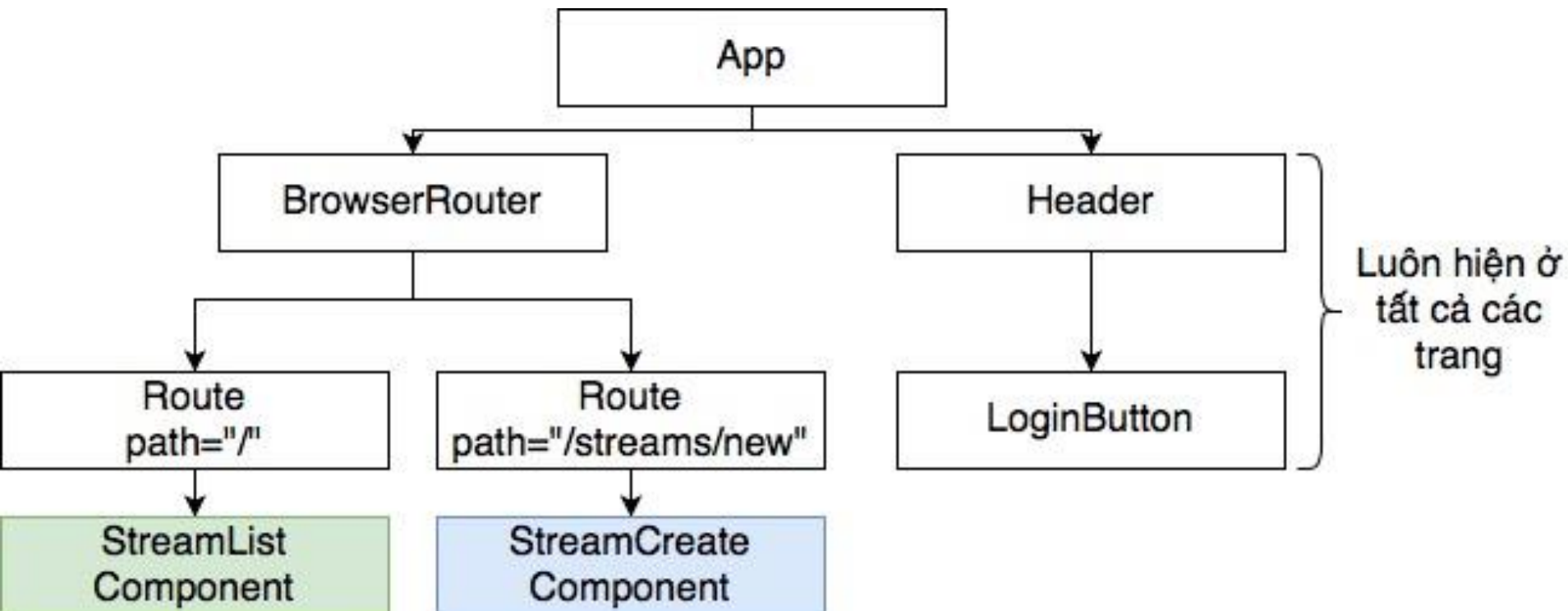
□ Tạo project có các component và route như sau



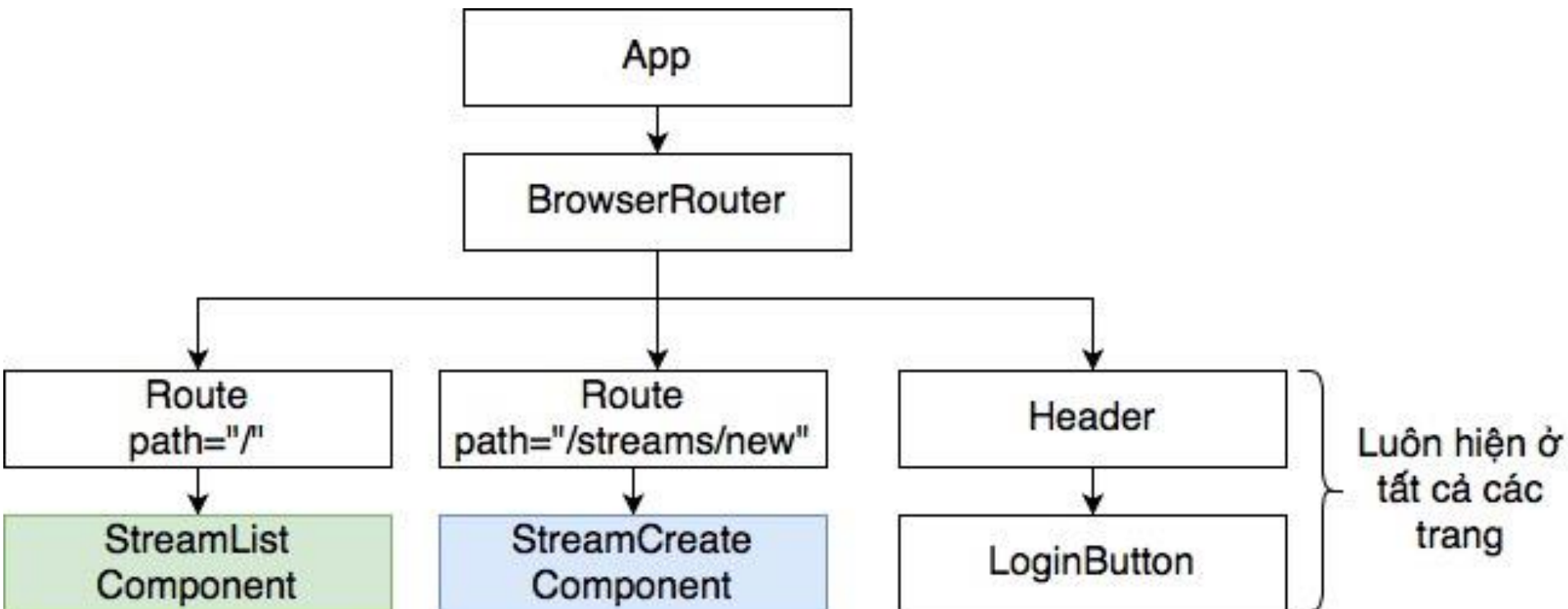
❑ Tạo header cho project



❑ Cấu trúc header component outside route



❑ Cấu trúc header component inside route





PHẦN 2

AUTHENCATION VỚI REACT

Các bước thiết lập OAuth

Tạo 1 new project tại console.developers.google.com

Thiết lập màn hình xác thực OAuth

Tạo ra 1 OAuth Client ID

Cài đặt thư viện Google's API, cài đặt với OAuth Client ID

Chắc chắn rằng thư viện được gọi bất cứ khi nào người dùng click vào nút "Login with Google"

https://console.developers.google.com/apis/dashboard?project=pagespeed-insights-221922&duration=PT1H

Select a project NEW PROJECT

Search projects and folders

RECENT ALL

Name	ID
✓ Pagespeed Insights ?	pagespeed-insights-221922
insights ?	insights-221902
dartdev ?	dartdev-206217
docker-react ?	docker-react-211322
REAL-gmaps-RSS ?	real-gmaps-rss
emaily-prod ?	emaily-prod-173022
emaily-dev ?	emaily-dev-172901
runner ?	runner-195104
PROD-REACT-SSR ?	prod-react-ssr
emaily ?	emaily-171322

[←](#)
[→](#)
[↻](#)
<https://console.developers.google.com/apis/credentials/consent?project=ringed-arcana-223119>

Google APIs

Streamy

API
APIs & Services

Dashboard

Library

Credentials

Credentials

Verification status
 Not published

Application name

The name of the app asking for consent

Streamy

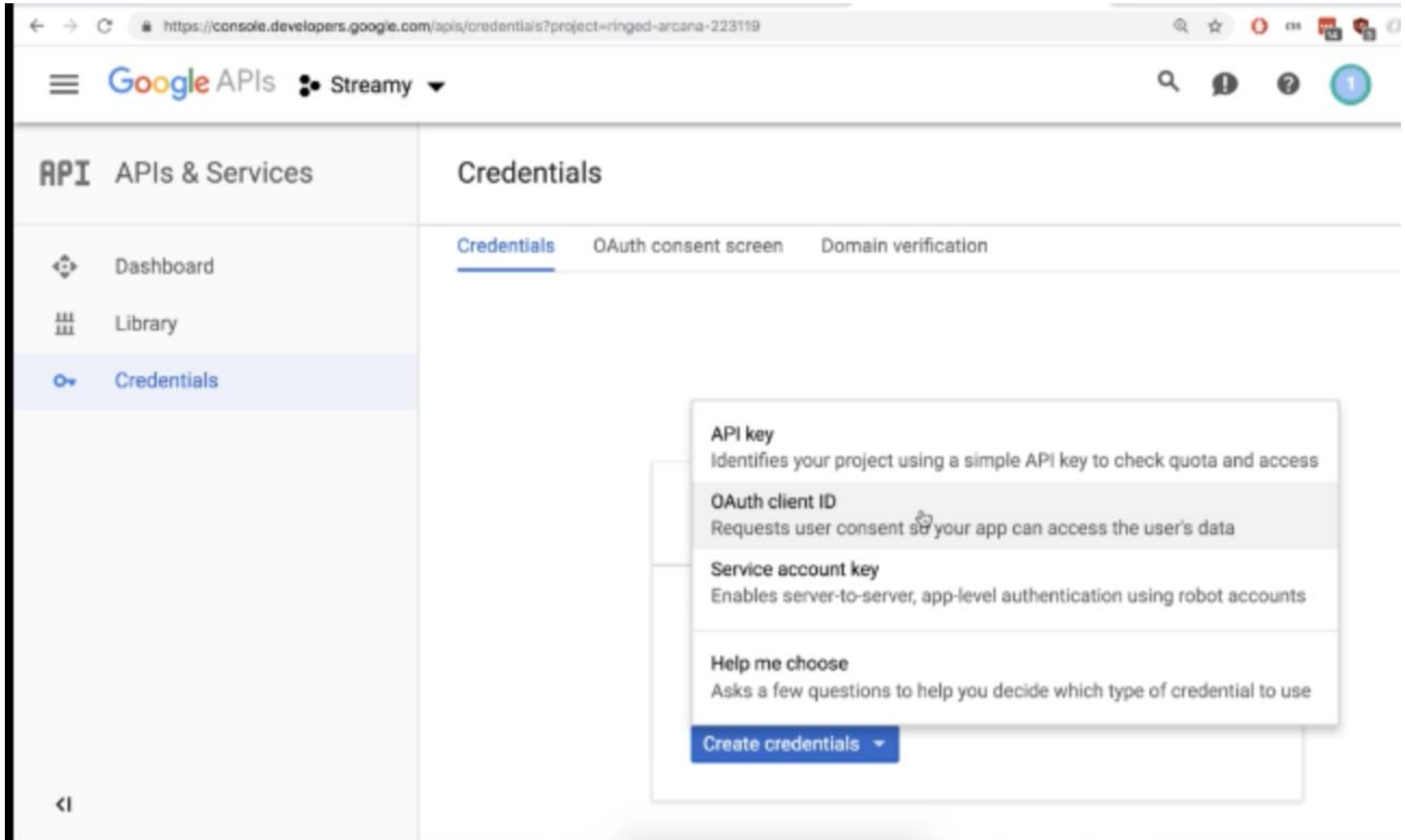
go

An image on the consent screen that will help users recognize your app

Support email

Shown on the consent screen for user support

Scopes for Google APIs
 Scopes allow your application to access your user's private data. [Learn more](#)



<https://console.developers.google.com/apis/credentials?project=ringed-arcana-223119>

Google APIs Streamy

API APIs & Services

Dashboard
 Library
Credentials

Credentials

[Credentials](#)
[OAuth consent screen](#)
[Domain verification](#)

API key
 Identifies your project using a simple API key to check quota and access



OAuth client ID
 Requests user consent so your app can access the user's data

Service account key
 Enables server-to-server, app-level authentication using robot accounts

Help me choose
 Asks a few questions to help you decide which type of credential to use

Create credentials

[←](#) [→](#) [↻](#) <https://console.developers.google.com/apis/credentials/oauthclient?project=ringed-arcana-223119>





[←](#) Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

☒ Web application [Learn more](#)
☐ Android [Learn more](#)
☐ Chrome App [Learn more](#)
☐ iOS [Learn more](#)
☐ Other

Name 

Web client 1

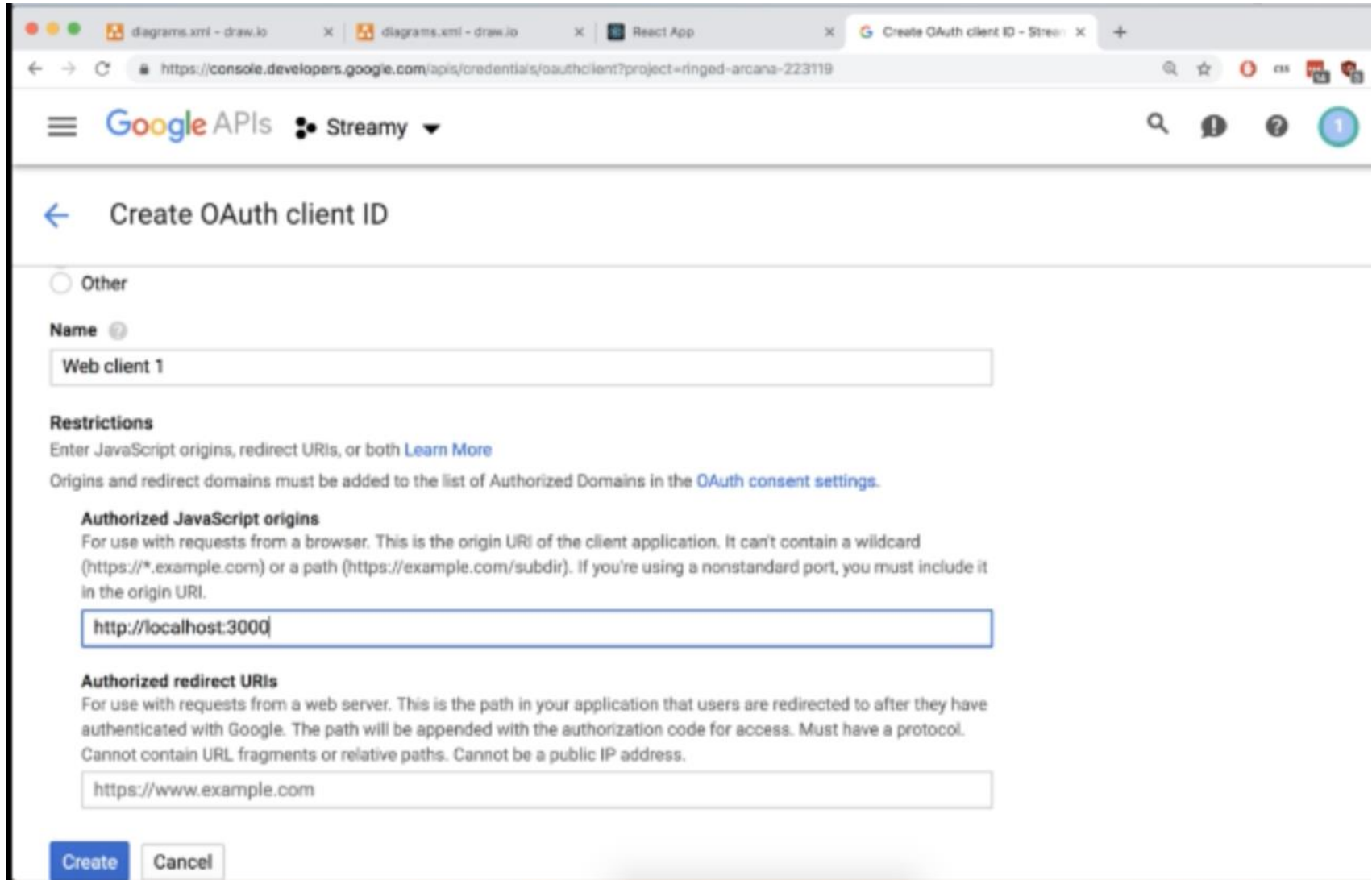
Restrictions

Enter JavaScript origins, redirect URIs, or both [Learn More](#)

Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.



diagrams.xml - draw.io X diagrams.xml - draw.io X React App X Create OAuth client ID - Stream X

https://console.developers.google.com/apis/credentials/oauthclient?project=ringed-arcana-223119

Google APIs Streamy

Create OAuth client ID

☐ Other

Name ?

Web client 1

Restrictions

Enter JavaScript origins, redirect URIs, or both [Learn More](#)

Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (https://*.example.com) or a path (https://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.

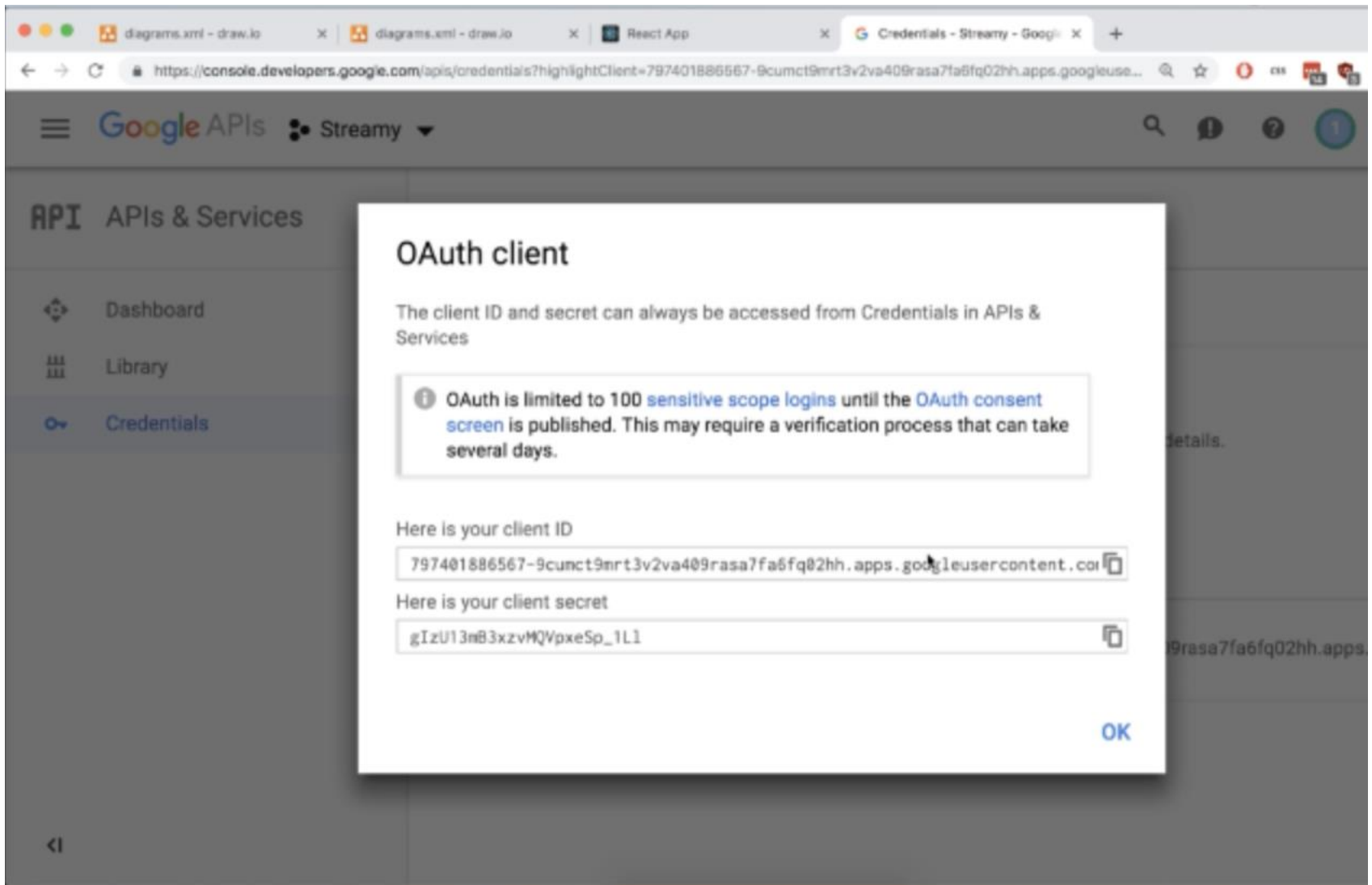
http://localhost:3000

Authorized redirect URIs

For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.

https://www.example.com

Create Cancel



The screenshot shows the Google Cloud Platform console with the 'Credentials' section selected. A modal window titled 'OAuth client' is displayed, providing the client ID and client secret for an OAuth client.

OAuth client

The client ID and secret can always be accessed from Credentials in APIs & Services

Information: OAuth is limited to 100 [sensitive scope logins](#) until the [OAuth consent screen](#) is published. This may require a verification process that can take several days.

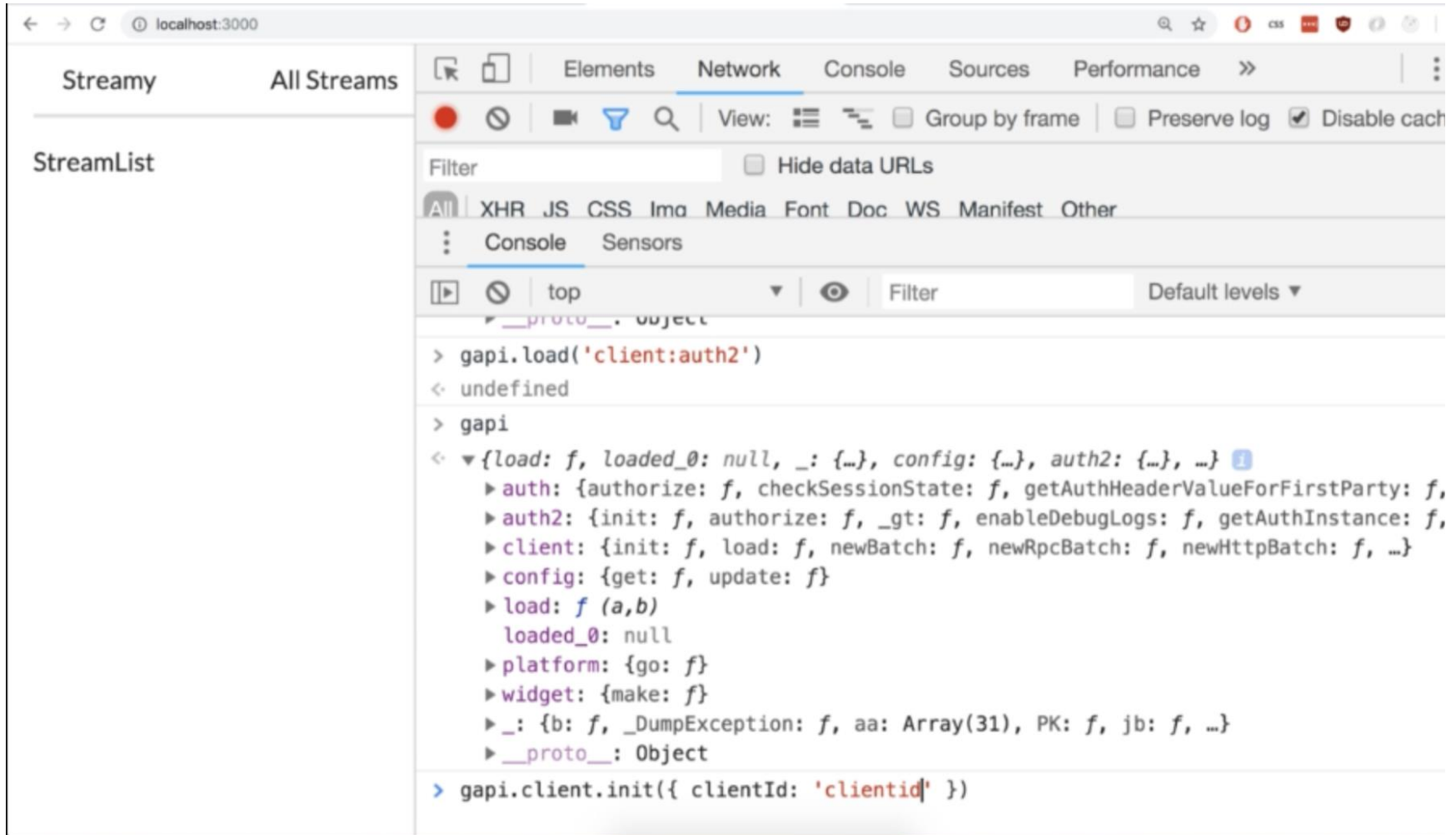
Here is your client ID

797401886567-9cunct9mrt3v2va409rasa7fa6fq02hh.apps.googleusercontent.com

Here is your client secret

gIzU13nB3xzvMQVpxeSp_1L1

OK



The screenshot shows a web browser at `localhost:3000` with the Google API Library loaded. The Network tab is active, displaying a successful XHR request to `gapi.load('client:auth2')`. The console shows the `gapi` object structure, which includes methods for authentication and client management.

```

> gapi.load('client:auth2')
< undefined
> gapi
< {load: f, loaded_0: null, _: {...}, config: {...}, auth2: {...}, ...}
  ▶ auth: {authorize: f, checkSessionState: f, getAuthHeaderValueForFirstParty: f, ...}
  ▶ auth2: {init: f, authorize: f, _gt: f, enableDebugLogs: f, getAuthInstance: f, ...}
  ▶ client: {init: f, load: f, newBatch: f, newRpcBatch: f, newHttpBatch: f, ...}
  ▶ config: {get: f, update: f}
  ▶ load: f(a,b)
    loaded_0: null
  ▶ platform: {go: f}
  ▶ widget: {make: f}
  ▶ _: {b: f, _DumpException: f, aa: Array(31), PK: f, jb: f, ...}
  ▶ __proto__: Object
> gapi.client.init({ clientId: 'clientid' })
  
```


GoogleAuth.js — streams

JS GoogleAuth.js X JS Header.js

EXPLORER

OPEN EDITORS

JS GoogleAuth.js cl...

JS Header.js client/...

STREAMS

client

public

favicon.ico

index.html

logo192.png

logo512.png

manifest.json

robots.txt

src

components

streams

JS StreamCreate.js

JS StreamDelete.js

JS StreamEdit.js

JS StreamList.js

JS StreamShow.js

JS App.js

JS GoogleAuth.js

JS Header.js

JS index.js

client > src > components > JS GoogleAuth.js > ...

1 import React from 'react';

2

3 class GoogleAuth extends React.Component {

4 componentDidMount() {

5 window.gapi.load('client:auth2', () => {

6 window.gapi.client.init({

7 clientId:

8 '797401886567-9cumct9mrt3v2va409rasa7fa6fq02hh.apps.googleusercontent.com',

9 scope: 'email'

10 });

11 });

12 }

13

14 render() {

15 return <div>Google Auth</div>;

16 }

17 }

18

19 export default GoogleAuth;

20

Header.js — streams

EXPLORER

OPEN EDITORS

JS GoogleAuth.js cl...

JS Header.js client/...

STREAMS

client

public

★ favicon.ico

<> index.html

logo192.png

logo512.png

{ } manifest.json

≡ robots.txt

src

components

streams

JS StreamCreate.js

JS StreamDelete.js

JS StreamEdit.js

JS StreamList.js

JS StreamShow.js

JS App.js

JS GoogleAuth.js

JS Header.js

JS index.js

🔍 .gitignore

{ } package.json

📖 README.md

client > src > components > JS Header.js > ...

1 import React from 'react';

2 import { Link } from 'react-router-dom';

3 import GoogleAuth from './GoogleAuth';

4

5 const Header = () => {

6 return (

7 <div className="ui secondary pointing menu">

8 <Link to="/" className="item">

9 Streamy

10 </Link>

11 <div className="right menu">

12 <Link to="/" className="item">

13 All Streams

14 </Link>

15 <GoogleAuth />

16 </div>

17 </div>

18);

19 };

20

21 export default Header;

22

Tham khảo hướng dẫn

The screenshot shows the Google API Client Libraries JavaScript Reference page for Authentication. The browser address bar displays the URL: `https://developers.google.com/api-client-library/javascript/reference/referencedocs#authentication`. The page title is "Google API Client Libraries > JavaScript". The navigation menu includes "HOME", "GUIDES", "REFERENCE", "SAMPLES", and "SUPPORT". The "Authentication" section is highlighted. The text describes the `GoogleAuth` singleton class and its methods. The `gapi.auth2.getAuthInstance()` method is shown, returning the `GoogleAuth` object. The `GoogleAuth.isSignedIn.get()` method is also shown, returning a Boolean value.

Authentication

`GoogleAuth` is a singleton class that provides methods to allow the user to sign in with a Google account, get the user's current sign-in status, get specific data from the user's Google profile, request additional scopes, and sign out from the current account.

`gapi.auth2.getAuthInstance()`

Returns the `GoogleAuth` object. You must initialize the `GoogleAuth` object with `gapi.auth2.init()` before calling this method.

Returns

<code>gapi.auth2.GoogleAuth</code>	The <code>gapi.auth2.GoogleAuth</code> object. Use this object to call <code>gapi.auth2.GoogleAuth</code> 's methods.
------------------------------------	---

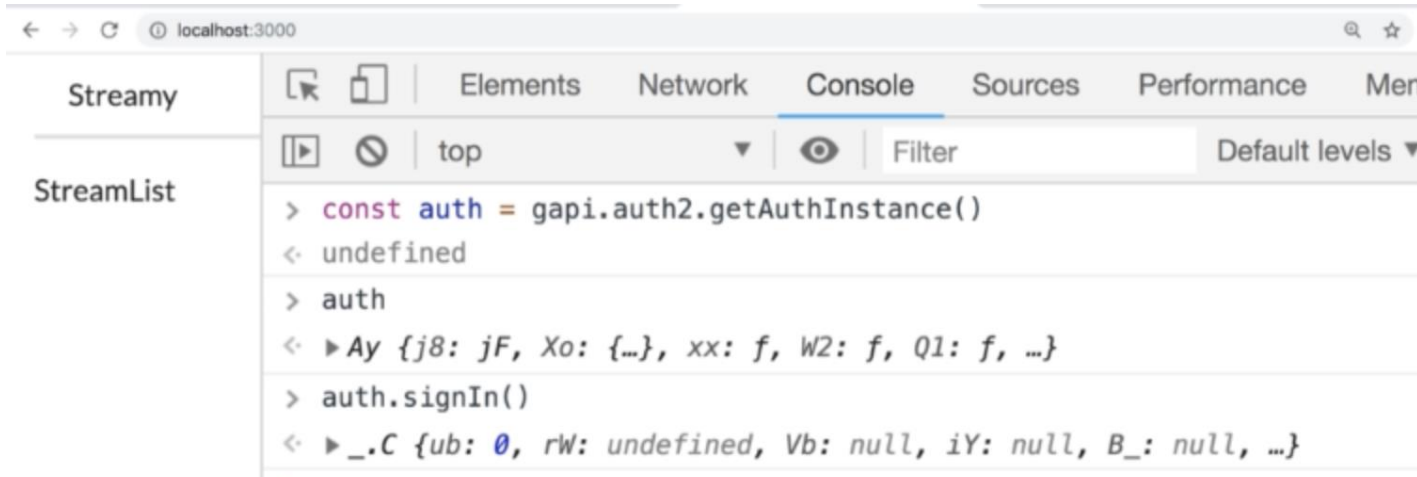
`GoogleAuth.isSignedIn.get()`

Returns whether the current user is currently signed in.

Returns

Boolean	<code>true</code> if the user is signed in, or <code>false</code> if the user is signed out or the <code>GoogleAuth</code> object isn't initialized.
---------	--

- ❑ `const auth = gapi.auth2.getAuthInstance()`
- ❑ `auth.signIn()`
- ❑ `auth.isSignedIn.get()`



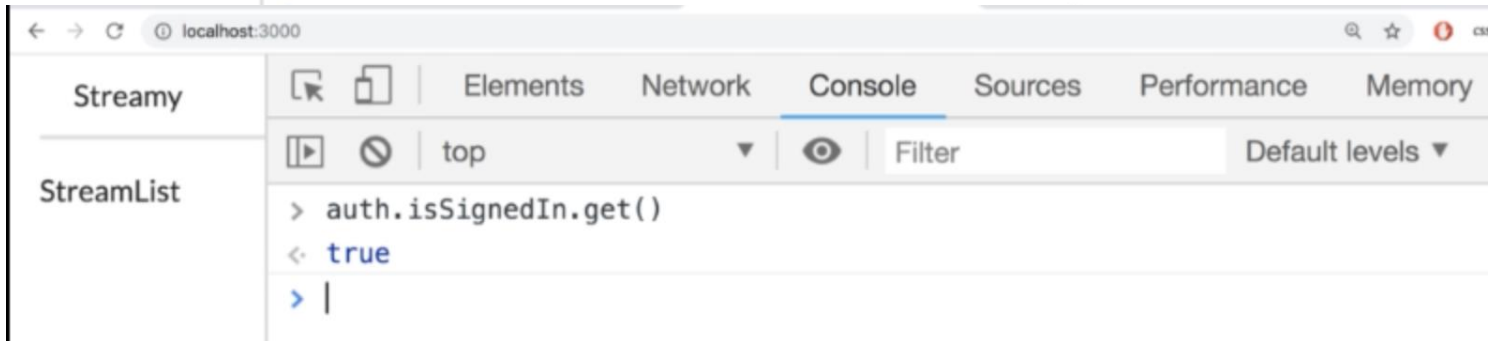
The screenshot shows a web browser at localhost:3000 with the developer console open. The console displays the following sequence of commands and results:

```

> const auth = gapi.auth2.getAuthInstance()
< undefined

> auth
< ▶ Ay {j8: jF, Xo: {...}, xx: f, W2: f, Q1: f, ...}

> auth.signIn()
< ▶ _.C {ub: 0, rW: undefined, Vb: null, iY: null, B_: null, ...}
  
```



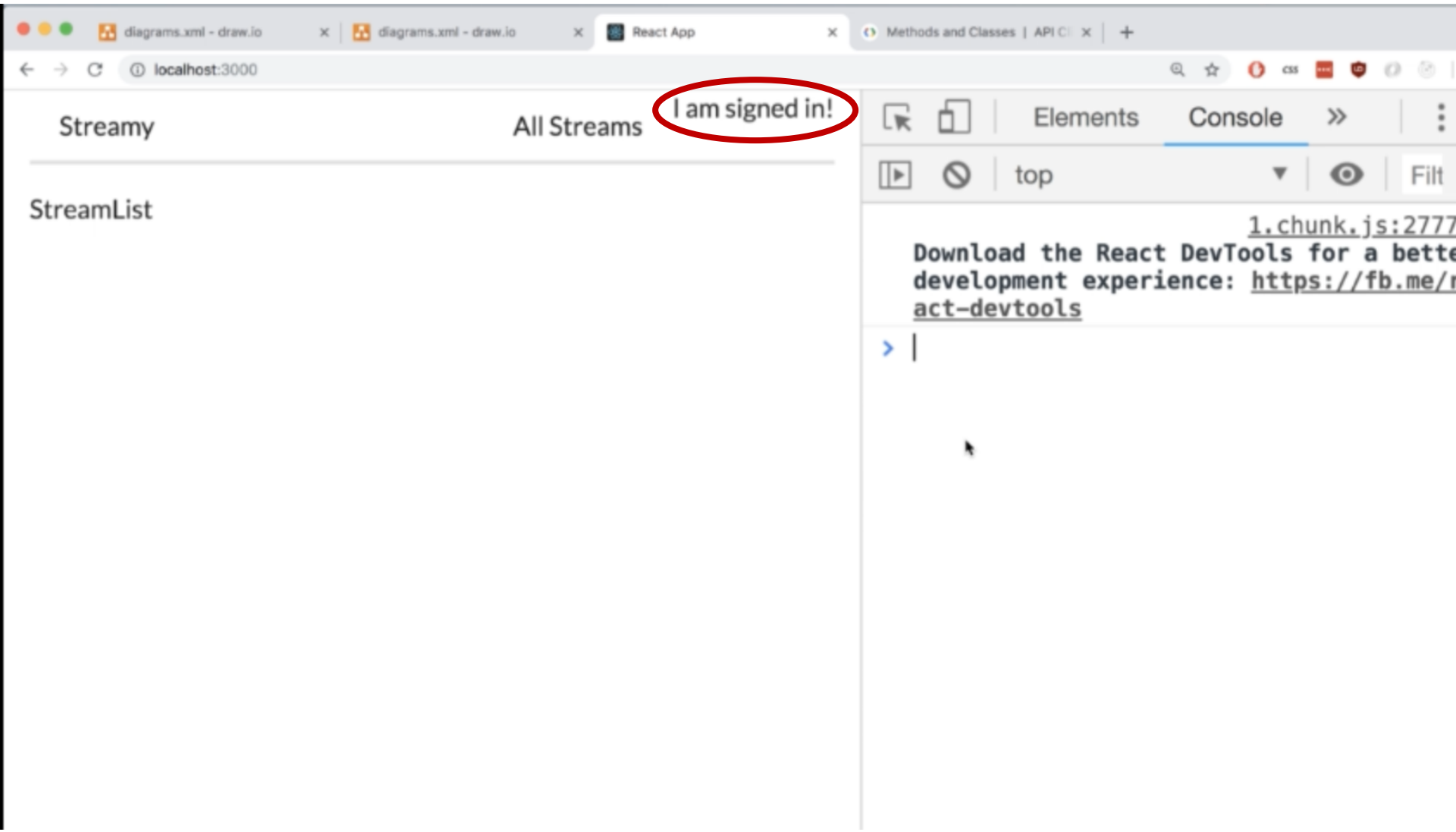
The screenshot shows the same web browser at localhost:3000 with the developer console open. The console displays the following command and result:

```

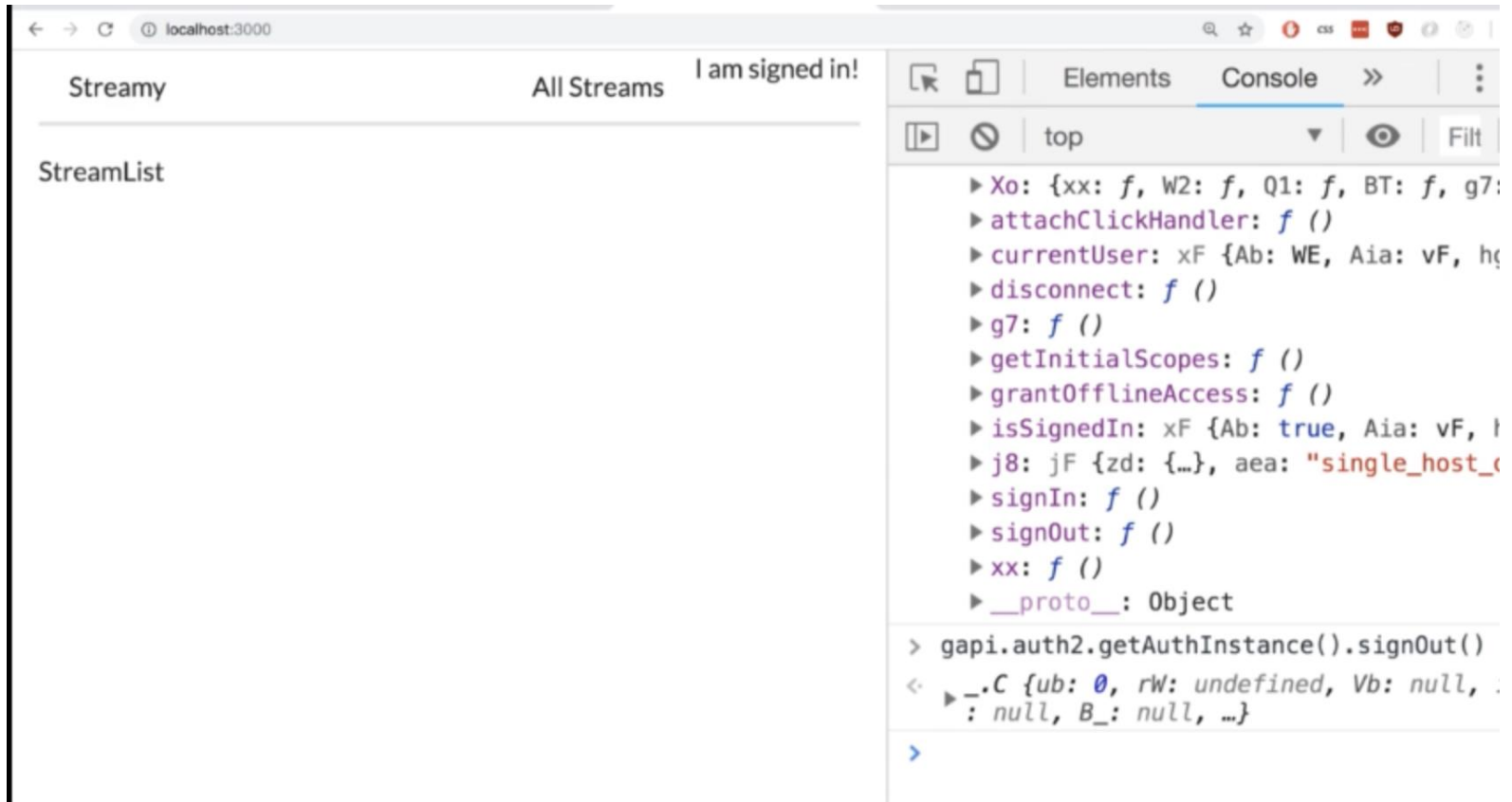
> auth.isSignedIn.get()
< true

> |
  
```

❑ Kết quả hiển thị



auth.signOut()



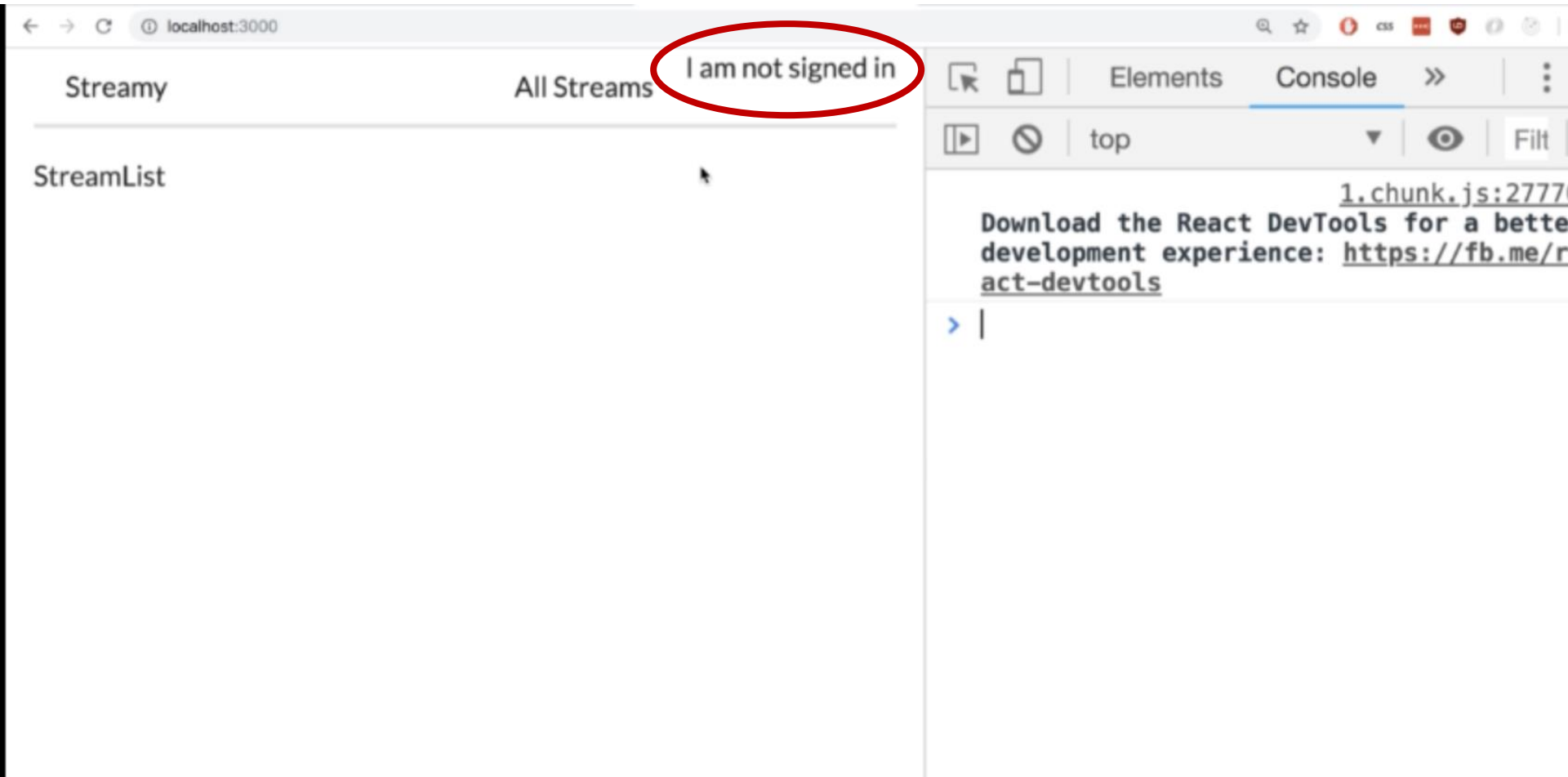
The screenshot shows a web application running on localhost:3000. The application has a header with "Streamy" and "All Streams" links, and a message "I am signed in!". Below the header is a "StreamList" section. The developer console is open, showing the "Console" tab. The console displays the following log:

```

> Xo: {xx: f, W2: f, Q1: f, BT: f, g7: f, attachClickHandler: f ()}
> attachClickHandler: f ()
> currentUser: xF {Ab: WE, Aia: vF, hg: f, disconnect: f ()}
> disconnect: f ()
> g7: f ()
> getInitialScopes: f ()
> grantOfflineAccess: f ()
> isSignedIn: xF {Ab: true, Aia: vF, hg: f, j8: jF {zd: {...}, aea: "single_host_c", signIn: f (), signOut: f (), xx: f (), __proto__: Object}
> signIn: f ()
> signOut: f ()
> xx: f ()
> __proto__: Object
> gapi.auth2.getAuthInstance().signOut()
< >_.C {ub: 0, rw: undefined, Vb: null, B_: null, ...}
>

```

❑ Kết quả hiển thị



demo

thank
you!