

Front End

Class 18

October 27, 2021

Agenda

- Kahoot – React concepts
- Review - Section 21 (Deploying React Apps) and
- Review - Section 22 (Adding Authentication to React Apps)
- Work on GA4

Deploying React Apps

From Development To Production

Deployment Steps & Pitfalls

Server-side Routing vs Client-side Routing

Deployment Steps



Test Code



Optimize Code



Build App for Production



Upload Production Code to Server



Configure Server

Lazy Loading

Load code only when it's needed

All Routes in App.js are evaluated and all the code for all the components for all the routes are Downloaded.

For example, we want to change the behavior to download the code for NewQuote when user visits that route.

Remove

```
import NotFound from './pages/NotFound';

const NewQuote = React.lazy(() => import('./pages/NewQuote'));
```

Doesn't work well as download takes time.
Need fallback ui or content if download takes longer.

Suspense Component.

```
<Suspense
  fallback={
    <div className='centered'>
      <LoadingSpinner />
    </div>
  }>
```

```
> react-complete-guide@0.1.0 build
> react-scripts build
```

Creating an optimized production build...

Compiled successfully.

File sizes after gzip:

49.7 KB	build/static/js/3.1c4baa01.chunk.js
4.06 KB	build/static/js/0.0c21642e.chunk.js
1.55 KB	build/static/js/runtime-main.db5d472c.js
1.28 KB	build/static/js/4.582350a3.chunk.js
914 B	build/static/js/5.dbcb604a.chunk.js
872 B	build/static/js/6.b5355a36.chunk.js
831 B	build/static/js/main.6630534a.chunk.js
744 B	build/static/css/main.0dfefc30.chunk.css
606 B	build/static/css/4.01c4d7ea.chunk.css
560 B	build/static/css/5.8a39e7ef.chunk.css
458 B	build/static/css/6.1c3d90f9.chunk.css
247 B	build/static/js/7.ee1d9215.chunk.js

Build folder has all the files needed to run application, translated to js and optimized.

A React SPA is a “Static Website”

Only HTML, CSS & JavaScript

Client-side JS code, runs in browser. No server-side code (No Node JS)

A Static Site Host Is Needed

Google, AWS - S3, Firebase is one possible option.
You are free to use any hosting provider.

```
react-complete-guide $ sudo npm install -g firebase-tools  
Password:  
((██████████)) : idealTree:zip-stream: sill fetch manifest fs-constants@^1.0.0
```

✗ Set up Firebase Hosting

1 Install Firebase CLI

2 Initialise your project

Open a terminal window and navigate to or create a root directory for your web app

Sign in to Google

\$ firebase login



Initiate your project

Run this command from your app's root directory:

\$ firebase init



Next

Might give you prompt in cli or browser to login

```
react-complete-guide $ firebase init
```

? Which Firebase CLI features do you want to set up for this folder? Press Space to select features, then Enter to confirm your choices.

- Database: Configure Firebase Realtime Database and deploy rules
- Firestore: Deploy rules and create indexes for Firestore
- Functions: Configure and deploy Cloud Functions
- Hosting: Configure and deploy Firebase Hosting sites
 - 7 people have written a note here. security rules
 - Emulators: Set up local emulators for Firebase features
 - Remote Config: Get, deploy, and rollback configurations for Remote Config

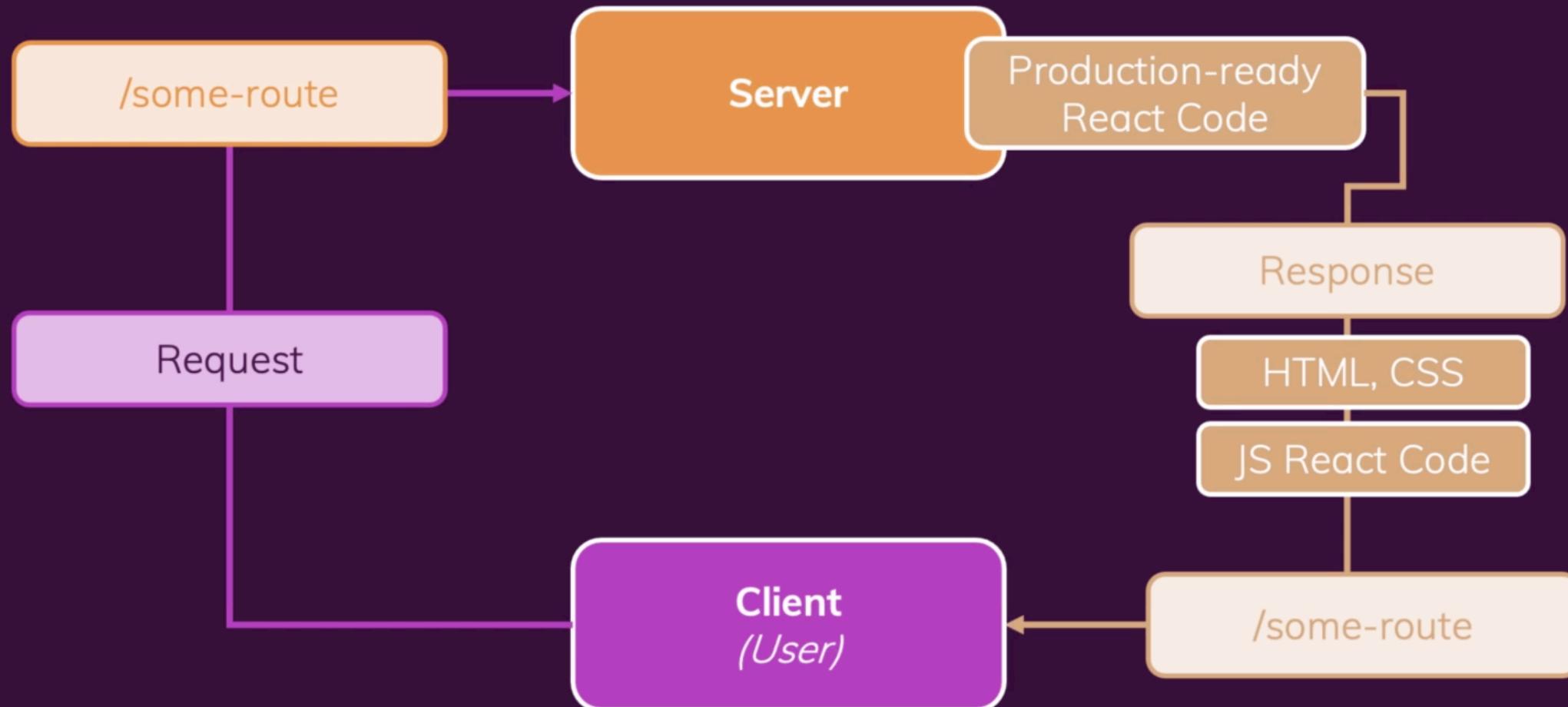
Can select hosting.

? Please select an option: Use an existing project

? Select a default Firebase project for this directory: react-prep (react-prep)

i Using project react-prep (react-prep)

Server-side Routing vs Client-side Routing



<https://www.pluralsight.com/guides/pros-and-cons-of-client-side-routing-with-react>

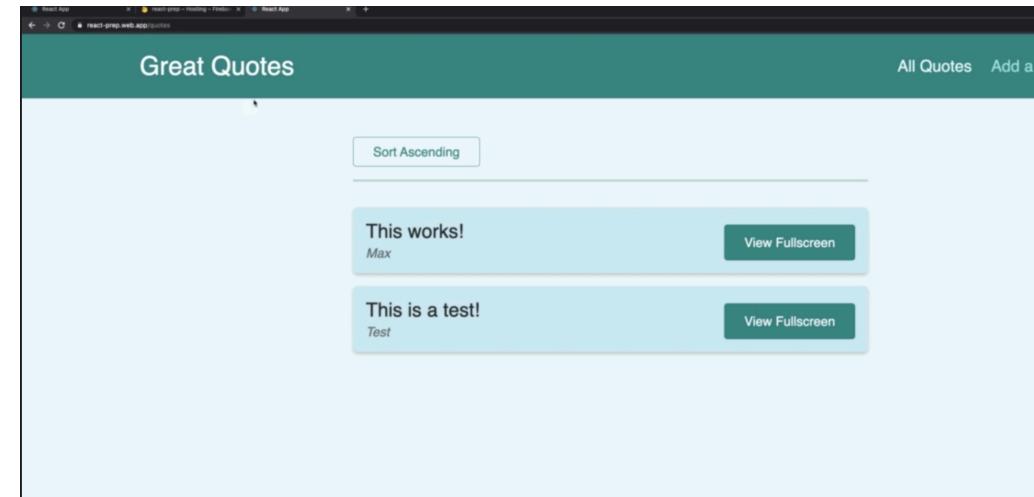
<https://medium.com/@wilbo/server-side-vs-client-side-routing-71d710e9227f>

```
? What do you want to use as your public directory? build  
? Configure as a single-page app (rewrite all urls to /index.html)? (y/N) ■
```

Y so firebase configure the server.
Single page application code.

```
react-complete-guide $ firebase deploy
```

```
Project Console: https://console.firebaseio.google  
Hosting URL: https://react-prep.web.app  
react-complete-guide $ ■
```



Authentication

Adding Authentication To React Apps

How Authentication Works In React Apps

Implementing User Authentication

Adding Auth Persistence & Auto-Logout

**Authentication is needed if
content should be protected**
(not accessible by everyone)

Your User Profile

New Password

Change Password

Authentication applies to website and API.

Two-step Process:

- 1) Get access / permission
- 2) Send request to protected resource

Login

Your Email

Your Password

Login

Create new account

How Does Authentication Work?

We can't just save and use the "yes"

We could send a fake "yes" to the server to request protected data

Server-side Sessions

Store unique identifier on server, send same identifier to client

Client sends identifier along with requests to protected resources

Authentication Tokens

Create (but not store) "permission" token on server, send token to client

Client sends token along with requests to protected resources

More on "Authentication Tokens"

When working with "Authentication Tokens", these tokens are typically created in the "JSON Web Token" Format (JWT).

As explained in the previous lecture, those "tokens" are really just long strings which are constructed by an algorithm that encodes data into a string (with help of a private key, only known by the server).

You can learn more about JSON Web Tokens (JWTs) here: <https://jwt.io/>

<https://jwt.io/>

Create Firebase account and Project

We will add these features to the demo App

Sign up

Login

Logout

Lock down access to pages without login

react prep ▾

Authentication

Authenticate and manage users from a variety of providers without server-side code

[Get started](#)



Authentication

Users **Sign-in method** Templates Usage

Sign-in providers

 Email/Password

Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives.

[Learn more ↗](#)

Email link (passwordless sign-in)

Enable

Cancel

Save

React Auth

Login Profile

Login

Your Email

Your Password

[Login](#)

[Create new account](#)

Create account using REST API

Check mode – login or sign-up mode.

```
const submitHandler = (event) => {
  event.preventDefault();

  const enteredEmail = emailInputRef.current.value;
  const enteredPassword = passwordInputRef.current.value;

  // optional: Add validation

  if (isLogin) {
  } else {
    fetch(
      'https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDex
    );
  }
};
```

```
if (isLogin) {
} else {
  fetch(
    'https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDex
  {
    method: 'POST',
    body: JSON.stringify({
      email: enteredEmail,
      password: enteredPassword,
      returnSecureToken: true
    }),
    headers: {
      'Content-Type': 'application/json'
    }
  );
}
};
```

```
2     if (isLoggedIn) {
3   } else {
4     fetch(
5       'https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDexE9BhcJbGxnZ4DiRlrCN9xe24',
6     {
7       method: 'POST',
8       body: JSON.stringify({
9         email: enteredEmail,
10        password: enteredPassword,
11        returnSecureToken: true,
12      }),
13      headers: {
14        'Content-Type': 'application/json',
15      },
16    }
17  ).then((res) => {
18    if (res.ok) {
19      // ...
20    } else {
21      return res.json().then((data) => {
22        // show an error modal
23        console.log(data);
24      });
25    }
26  });
27});
28};
```

React Auth

[Login](#)[Profile](#)[Logout](#)

Sign Up

Your Email

Your Password

Create Account

Login with existing account

Console top

▶ POST https://identitytool AUTHFORM.js:25
kit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDexE9BhcJbGxnZ4DiRlrCN9xe24
400

AuthForm.js:44

▼ {error: {...}} *i*

▼ error:

code: 400

► errors: [{...}]

message: "WEAK_PASSWORD : Password ...

WEAK_PASSWORD : Password should be at least 6 characters

>

The screenshot shows the Firebase Authentication console for a project named "react-prep". The left sidebar lists various services: Build (Authentication, Firestore, Realtime Database, Storage, Hosting, Functions, Machine Learning), Release and monitor (Crashlytics, Performance, Test Lab...), Analytics (Dashboard, Realtime, Extensions), and Spark (Free \$0/month, Upgrade). The main area displays the Authentication screen with tabs for Users, Sign-in method, Templates, and Usage. A search bar at the top allows searching by email address, phone number, or user UID. Below it is a table with columns: Identifier, Providers, Created, Signed In, and User UID (sorted by User UID). One row is shown: test@test.com, dropdown, 6 Apr 2021, 6 Apr 2021, 5cCxqgTt41cOe0znYojmDZBBeuq2. The table includes pagination (Rows per page: 50, 1 - 1 of 1) and a three-dot menu icon. At the bottom of the main area is a control bar with icons for volume, file, CC, settings, and a double arrow. The right side of the screen features a "Course content" sidebar with a list of 13 items, each with a play button and duration: 302. Adding User Signup (16min), 303. Showing Feedback To The User (6min), 304. Adding User Login (7min), 305. Managing The Auth State With Context (12min), 306. Using The Token For Requests To Protected Resources (11min), 307. Redirecting The User (3min), 308. Adding Logout (3min), 309. Protecting Frontend Pages (7min), 310. Persisting The User Authentication Status (7min), 311. Adding Auto-Logout (8min), 312. Finishing Steps (11min), and 313. Module Resources (1min). A "Resources" button is located next to item 310.

[Overview](#) [Q&A](#) [Notes](#) [Announcements](#) [Learning tools](#)

About this course

Dive in and learn React.js from scratch! Learn Reactjs, Hooks, Redux, React Routing, Animations, Next.js and way more!

By the numbers

Skill level: All Levels
Students: 462470

Lectures: 488
Video: 48 total hours



1635077000717.jpeg

Show All

- [Course content](#)
 - [302. Adding User Signup](#) 16min
 - [303. Showing Feedback To The User](#) 6min
 - [304. Adding User Login](#) 7min
 - [305. Managing The Auth State With Context](#) 12min
 - [306. Using The Token For Requests To Protected Resources](#) 11min
 - [307. Redirecting The User](#) 3min
 - [308. Adding Logout](#) 3min
 - [309. Protecting Frontend Pages](#) 7min
 - [310. Persisting The User Authentication Status](#) 7min
 - [311. Adding Auto-Logout](#) 8min
 - [312. Finishing Steps](#) 11min
 - [313. Module Resources](#) 1min
- [Section 23: A \(Pretty Deep Dive\) Introduction to Next.js](#) 1 / 35 | 3hr 16min
- [Section 24: Animating React Apps](#) 0 / 16 | 1hr 8min
- [Section 25: Replacing Redux with React Hooks](#) 0 / 14 | 54min
- [Section 26: Testing React Apps \(Unit Tests\)](#) 14 / 14 | 1hr 12min

Show feedback to user

```
  .then((res) => {
    if (res.ok) {
      // ...
    } else {
      return res.json().then((data) => {
        let errorMessage = 'Authentication failed!';
        if (data && data.error && data.error.message) {
          errorMessage = data.error.message;
        }
        alert(errorMessage);
      });
    }
  })
```

localhost:3000 says
WEAK_PASSWORD : Password should be at least 6 characters

OK

Console ➞ top

Filter

✖ ▶ POST https://identitytool AuthForm.js:25
kit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDexE9BhcJbGxnZ4DiRlrCN9xe24
400

AuthForm.js:44

▼ {error: {...}} ⓘ
 ▼ error:
 code: 400
 ► errors: [...]
 message: "WEAK_PASSWORD : Password ...
 ► __proto__: Object
 ► __proto__: Object

✖ ▶ POST https://identitytool AuthForm.js:25
kit.googleapis.com/v1/accounts:signUp?key=AIzaSyBZhsabDexE9BhcJbGxnZ4DiRlrCN9xe24
400

React Auth Login Profile Logout

Sign Up

Your Email

test2@test.com

Your Password

...

Create Account

Login with existing account

Can change to generic message and also show loading state.

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/03-showing-feedback/src/components/AuthForm.js>

```
        )
      .then((res) => {
        setIsLoading(false);
        if (res.ok) {
          // ...
        } else {
          return res.json().then((data) => {
            let errorMessage = 'Authentication failed!';
            // if (data && data.error && data.error.message) {
            //   errorMessage = data.error.message;
            // }
            alert(errorMessage);
          });
        }
      });
    }
};
```

Adding User Login

Auth token that is used in following requests.

<https://firebase.google.com/docs/reference/rest/auth>

```
    return res.json();
} else {
  return res.json().then((data) => {
    let errorMessage = 'Authentication failed!';
    // if (data && data.error && data.error.message) {
    //   errorMessage = data.error.message;
    // }

    throw new Error(errorMessage);
});
}
})
.then((data) => {})
.catch((err) => [
  alert(err.message),
]);
};
```

localhost:3000/auth

Incognito (2) Update

React Auth

Login Profile Logout

Login

Your Email

Your Password

Login

Create new account

Console >

top

Filter

```
AuthForm.js:60
{kind: "identitytoolkit#VerifyPasswordRe
sponse", localId: "5cCxqgTt41c0e0znYojmQ
ZZBeuq2", email: "test@test.com", displa
▼yName: "", idToken: "eyJhbGciOiJSUzI1NiI
sImtpZCI6IjhkOGM3OTd1MDQ5YWFKZW...tmFIwf
p6
GJw0THD5sAwz6u8Bs3Gw7cCMH60aHH6IqVJ29be0
A", ...} ⓘ
displayName: ""
email: "test@test.com"
expiresIn: "3600"
idToken: "eyJhbGciOiJSUzI1NiIsImtpZCI...
kind: "identitytoolkit#VerifyPassword...
localId: "5cCxqgTt41c0e0znYojmQZZBeuq...
refreshToken: "A0vuKvRuASZs0acL10PZhQ...
registered: true
► __proto__: Object
```

Managing the Auth State with Context API

Add store folder to hold context code. auth-context.js

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/05-using-context-for-managing-auth-state/src/store/auth-context.js>

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/05-using-context-for-managing-auth-state/src/store/auth-context.js>

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/05-using-context-for-managing-auth-state/src/components/Auth/AuthForm.js>

Using the token for requests to protected resource

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/06-attaching-tokens-to-requests/src/components/Profile/ProfileForm.js>

Redirecting user for login

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/07-adding-logout/src/components/Auth/AuthForm.js>

Logout

Update state and set context to null.

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/07-adding-logout/src/components/Layout/MainNavigation.js>

Protected Frontend Pages

Navigation guards

Only visit routes defined in App.js

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/08-protecting-pages/src/App.js>

Persisting the User Auth Status

When we reload, we lose the auth status, for user to stay logged in, at least for a given amount of time.
Token expires in some time usually.
Expiration time is 1 hour default. Security mechanism.

Need to store token outside of JS state.

Browser has cookies. Even easier to use is localStorage.

<https://academind.com/tutorials/localstorage-vs-cookies-xss>

Persisted login status

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/09-finished/src/store/auth-context.js>

Login

Your Email

Your Password

[Create new account](#)



Adding Auto Logout

<https://github.com/academind/react-complete-guide-code/blob/22-authentication/code/09-finished/src/store/auth-context.js>

```
2 const calculateRemainingTime = (expirationTime) => {
3   const currentTime = new Date().getTime();
4   const adjExpirationTime = new Date(expirationTime).getTime();
5
6   const remainingDuration = adjExpirationTime - currentTime;
7
8   return remainingDuration;
9 };
0
```

Adding Auto Logout



A screenshot of a React application interface showing a login form. The top navigation bar is dark purple with the text "React Auth" on the left and a "Login" button on the right. The main content area has a dark purple background. It contains a "Login" heading, two input fields labeled "Your Email" and "Your Password", a "Login" button, and a "Create new account" link at the bottom.

Logout after expiration time passes. Need to set new timer when user logs in.
Also remove after logout.

Great job!

Practice, Practice, Practice

Build Demo & Dummy Projects

Clone popular websites & web apps

Re-build the course projects

Come up with dummy business ideas & websites

Search for “React examples”

Next.js – react framework for production. Large scale production ready apps. Full stack framework.
Gatsby.js produce static websites pre-rendered for search engines optimization. Better user experience.
Preact – React with smaller footprint. Strips out internal functionality with build in optimization.
React Native framework built by facebook for mobile apps. Don't need to learn android or swift.

Explore the React Ecosystem



Gatsby.js



Preact



React Native

