

Micro Frontends With a Hands-On Example Using React, Webpack 5, and Module Federation

Micro Frontends Step by Step Using React, Webpack 5, and Module Federation



Rany ElHousieny , PhD^{ABD} | Follow
SENIOR MANAGER SOFTWARE E...



14



0



0

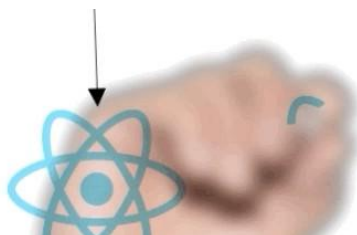
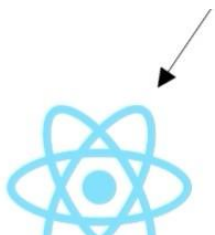
In this article, I will go step by step in creating two Micro Frontend React Components and render a Button component from one into the other. This hands-on example is a continuation to the following two articles where I explained how Micro Frontends work with Webpack5 and Module Federation. I wanted to give clean steps on React components without all the explanations. Please, refer to those articles for explanations



Micro-Front-Ends: Hands-On Project

Medium

Prepare your machine and install nodejs, if needed follow the article/video, below:



Micro Frontends Hands-On Example Using React, Webpack 5, and Module Federation: Adding a third...

Medium

This article is a continuation of the previous article

The final project can be found at <https://github.com/ranyelhousieny/react-microfrontends>

Video for the project at

Create Micro Frontends: Webpack configuration



=====

What is a Micro-Frontend? #shorts



Create Micro Frontends Step By Step: Introduction #shorts #microfrontends #reac...



1. Create two React apps and install dependencies

```
npx create-react-app mfe1
```

```
cd mfe1
```

```
yarn add webpack webpack-cli webpack-server html-webpack-plugin babel-loader webpack-dev-server
```

```

Rany>

```

```
npx create-react-app mfe2
```

```
cd mfe2
```

```
yarn add webpack webpack-cli webpack-server html-webpack-plugin babel-loader webpack-dev-server
```

Create Micro Frontends Step By Step: Step 1 #shorts #microfrontends #react #we...



2. In both Apps, rename index.js to bootstrap.js

in **bootstrap.js**, Remove the imports to index.css and reportWebVitals and keep the file as shown below:

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';
```

```
ReactDOM.render(
  <App />,
  document.getElementById(
    'root'
  )
);
```

Here are the files on Github

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe1/src/bootstrap.js>

We do this step to make bootstrap load asynchronously. It is needed for MFE2 to wait for Webpack to fetch components from MFE1.

3. In both Apps, create a new index.js that has one line

```
import('./bootstrap');
```

Here is the file on Github

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe1/src/index.js>

Create Micro Frontends Step By Step: Step2 #shorts #microfrontends #react #we...



4. In MFE1, Create. a Button component

Create a file src/Button.js

Copy inside it this code (It is a very simple component)

```
import React from 'react';
```

```
const Button = () => (  
  <button>MFE1 Button</button>  
)
```

```
export default Button;
```

Code can be found at

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe1/src/Button.js>

4.Create Micro Frontends: create a button on MFE1 #shorts Step3



4.1. In MFE1, update App.js as follows

```
import React from 'react';
import Button from './Button';
const App = () => {
  return (
    <div>
      <Button />
    </div>
  );
};

export default App;
```

Code can be found at

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe1/src/App.js>

5.Create Micro Frontends MFE1 App.js #shorts



5. In MFE1, Create a Webpack.config.js on the root and put the following inside it

(details about this file and building it step by step can be

found [here https://www.linkedin.com/pulse/understanding-micro-frontends-webpack5-configurations-rany/](https://www.linkedin.com/pulse/understanding-micro-frontends-webpack5-configurations-rany/) and the video <https://youtu.be/AZDDIgJSKU0>)

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
```

```
const ModuleFederationPlugin = require('webpack/lib/container/ModuleFederationPlugin');
```

```
module.exports = {
  mode: 'development',
  devServer: {
    port: 8083,
  },
  module: {
    rules: [
      {
        /* The following line to ask babel
        to compile any file with extension
```

```

    .js */
    test: /\.js?$/,

    /* exclude node_modules directory from babel.
       Babel will not compile any files in this directory*/
    exclude: /node_modules/,

    // To Use babel Loader
    loader:
      'babel-loader',
    options: {
      presets: [
        '@babel/preset-env' /* to transfer any advanced ES to ES5 */,
        '@babel/preset-react',
      ], // to compile react to ES5
    },
  ],
},

plugins: [
  new ModuleFederationPlugin(
    {
      name: 'MFE1',
      filename:
        'remoteEntry.js',

      exposes: {
        './Button':
          './src/Button',
      },
    }
  ),
  new HtmlWebpackPlugin({
    template:
      './public/index.html',
  }),
],
};

```

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe1/webpack.config.js>

Here we created MFE1 and made it expose the Button Component through

<http://localhost:8083/remoteEntry.js>. Webpack renders this on <http://localhost:8083>.

```
plugins: [  
  new ModuleFederationPlugin(  
    {  
      name: 'MFE1',  
      filename:  
        'remoteEntry.js',  
      exposes: {  
        './Button':  
          './src/Button',  
      },  
    },  
  ),  
]
```

MFE1

webpack.config.js

Create Micro Frontends: Webpack configuration



6. Create Micro Frontends: yarn add webpack webpack-cli #shorts**7. Create Micro Frontends: Mode: Development in webpack.config: #shorts**

6. In MFE2, Create a Webpack.config.js on the root and put the following inside it

```
const HtmlWebpackPlugin = require('html-webpack-plugin');  
const ModuleFederationPlugin = require('webpack/lib/container/ModuleFederationPlugin');
```

```
module.exports = {  
  mode: 'development',  
  devServer: {  
    port: 8082,  
  },  
  module: {  
    rules: [  
      {  
        /* The following line to ask babel  
           to compile any file with extension  
           .js */  
        test: /\.js?$/,  
  
        /* exclude node_modules directory from babel.  
           Babel will not compile any files in this directory*/  
        exclude: /node_modules/,  
      },  
    ],  
  },  
};
```



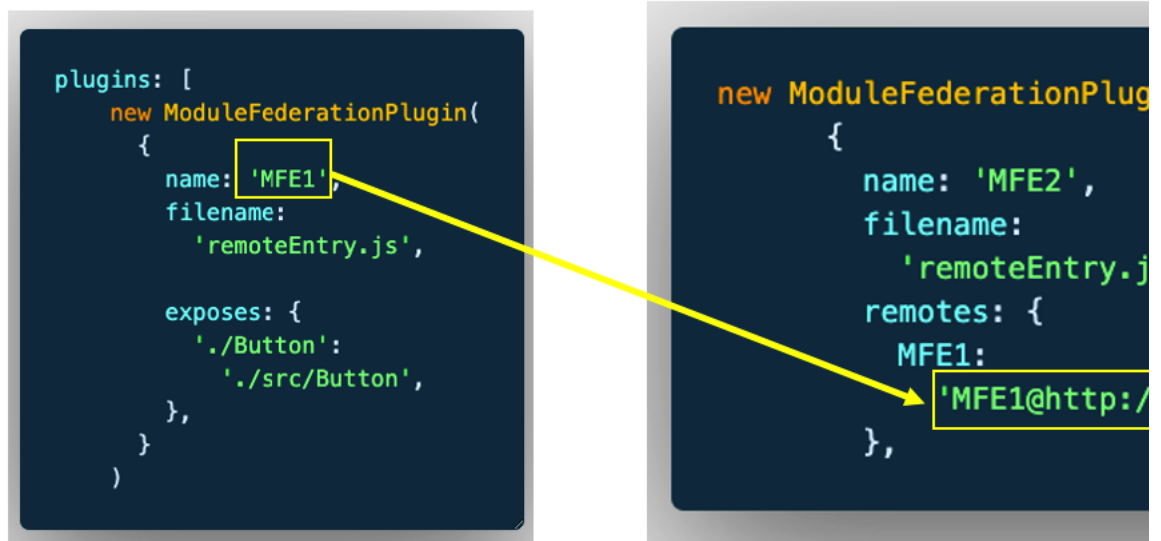
```

// To Use babel Loader
loader:
  'babel-loader',
options: {
  presets: [
    '@babel/preset-env' /* to transfer any advanced ES to ES5 */,
    '@babel/preset-react',
  ], // to compile react to ES5
},
},
],
},

plugins: [
  new ModuleFederationPlugin(
    {
      name: 'MFE2',
      filename:
        'remoteEntry.js',
      remotes: {
        MFE1:
          'MFE1@http://localhost:8083/remoteEntry.js',
      },
    }
  ),
  new HtmlWebpackPlugin({
    template:
      './public/index.html',
  }),
],
};

```

<https://github.com/ranyelhousieny/react-microfrontends/blob/main/mfe2/webpack.config.js>



MFE1
webpack.config.js

M
webp

Here we just said that MFE1 exposed components can be available in MFE2 remotely through <http://localhost:8083/remoteEntry.js>

7. In MFE2 src/App.js import Button (Lazy import)

```
const MFE1_Button = React.lazy(  
  () => import('MFE1/Button')  
);
```

Add it to the App as follows

```
function App() {  
  return (  
    <div>  
      <h1>MFE2</h1>  
      <div>  
        <React.Suspense fallback='Loading Button'>  
          <MFE1_Button />  
        </React.Suspense>  
      </div>  
      <h2>MFE2</h2>  
    </div>  
  );  
}
```

open a terminal inside MFE1 directory and run

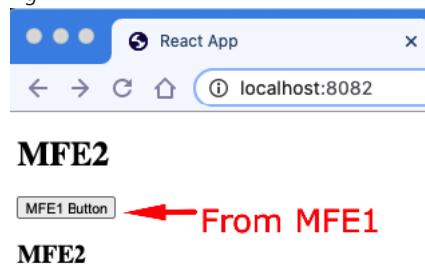
```
yarn webpack serve
```

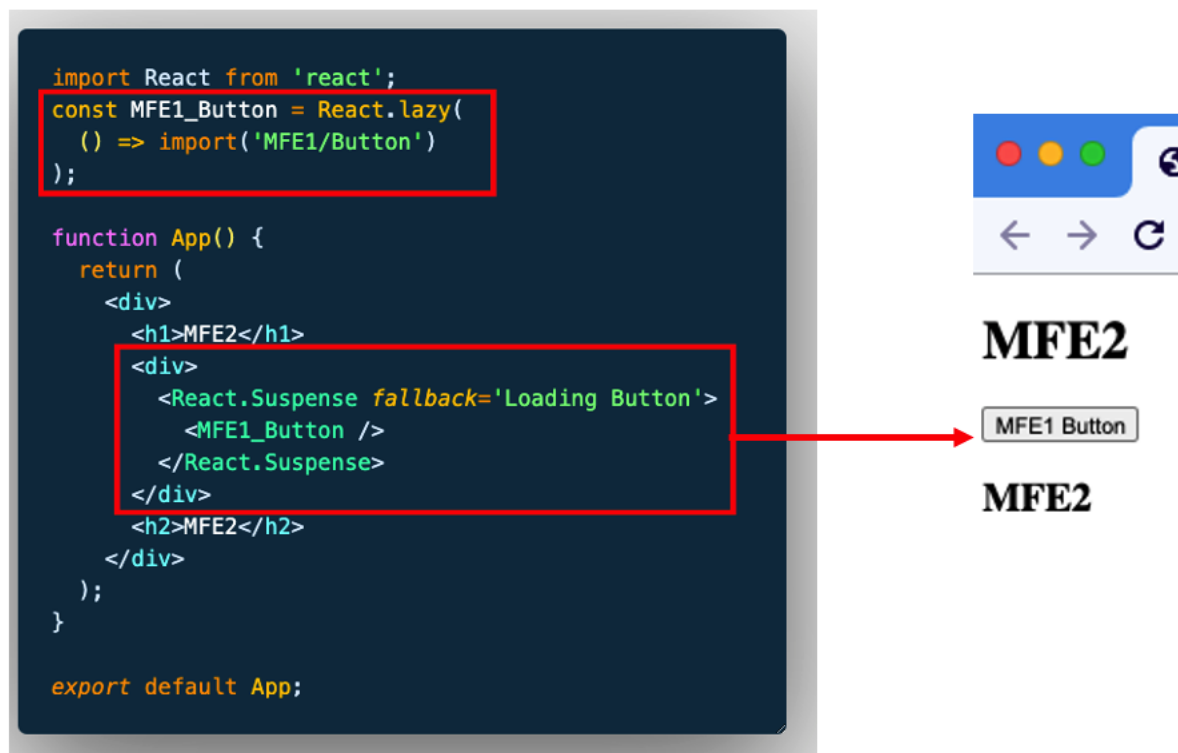
Do the same on MFE2 directory

Now open the browser and navigate to

<http://localhost:8082/>

You should see the following





MFE2

App.js

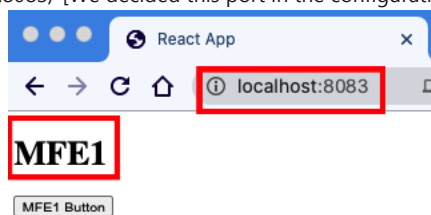
=====

Explanation:

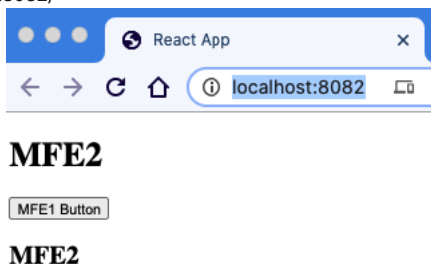
Now, let's take a step back and understand what we have done.

First: Webpack created two servers for us

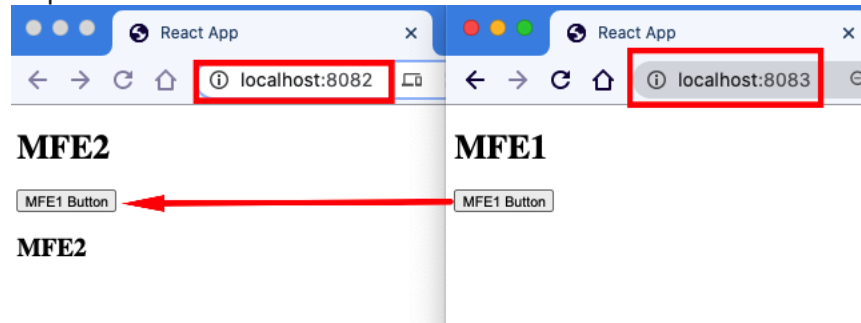
MFE1 on <http://localhost:8083/> [We decided this port in the configuration]



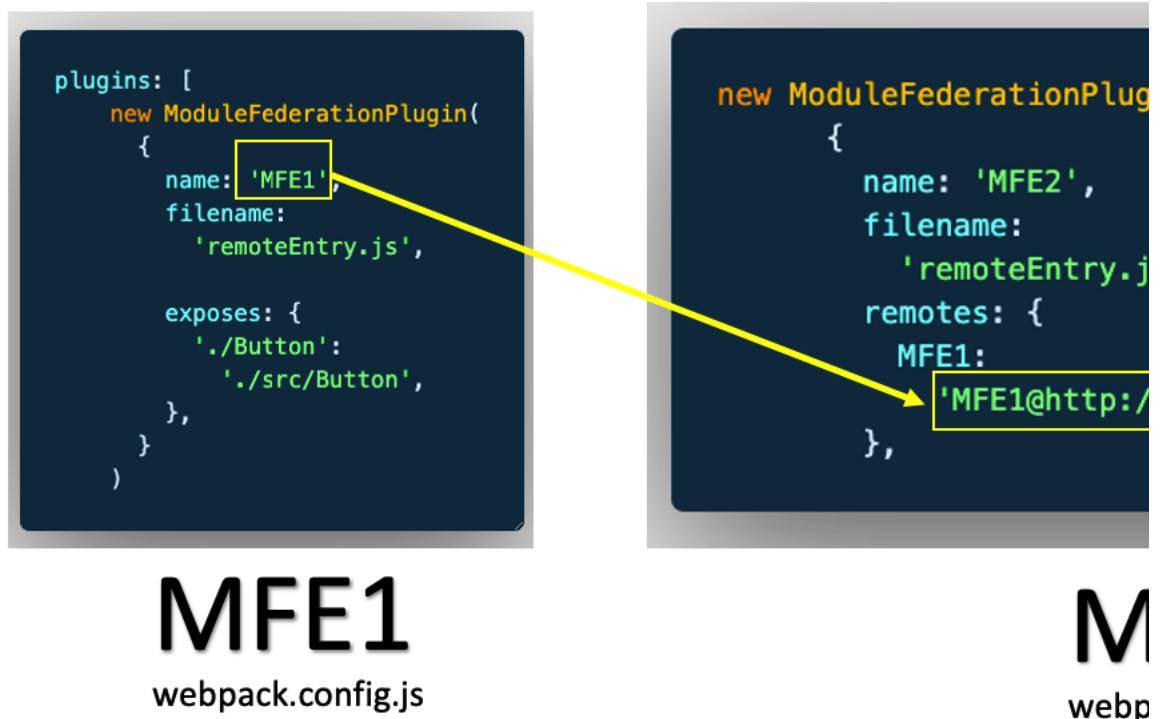
MFE2 on <http://localhost:8082/>



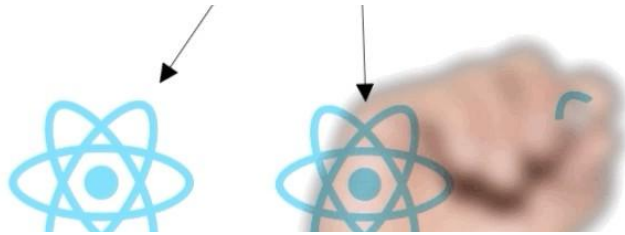
Second: Using Module Federation Plugin we exposed the Button component from MFE1 site (<http://localhost:8083/>) and imported it into MFE2



Here is the configuration that did that



Other Micro Frontend articles at:



Micro Frontends Hands-On Example Using React, Webpack 5, and Module Federation: Adding a third...

Medium

This article is a continuation of the previous article



Micro-Front-Ends: Hands-On Project

Medium

Prepare your machine and install nodejs, if needed follow the article/video, below:

Published By



Rany ElHousieny , PhD^{ABD}
SENIOR MANAGER SOFTWARE ENG...

Follow

#microfrontends

0 comments



[Sign in](#) to leave your comment

More from Rany ElHousieny , PhD^{ABD} 59 articles

Steps for Migrating from Monolith to...

June 19, 2021

Design Twitter

June 18, 2021

Create Single MicroFrontend with...

June 18, 2021

