# Module\_46\_Simple\_React\_REST\_Countries

#### **46-1 Module Introduction, load countries using useEffect**

* Load Countries data using REST Countries API (‘<https://restcountries.com/v2/all>’).

#### **46-2 Display countries in a simple way, Folder Structure**

* To organizing the Components for small and medium applications, we can create a folder named **Components** inside the **src** and keep the Component by creating a future related folder. i.e. for Countries component - **src/Components/Countries/** - **Countries.js** & **Countries.css**

#### **46-3 React Extension Pack and component with css file**

* Install an extension named ‘**react extension pack.**’ It makes our life easy. i.e. -> **rsc** (**stateless components skeleton**) - creates a skeleton for the Components (includes import form react and Component Arrow function). It also auto-import the Component file in **App.js** while we use the Component.

#### **46-4 Components inside the folder structure and connect them**

* Delete the Components folder and rebuild the folder structure using Components for Countries (practice purpose).

#### **46-5 Create child component and pass data via props**

* Create a child component of **Countries** named **Country**. Pass country’s name, capital, and population from **Countries**. Receive them in **Country** using **props** and display the **Country** name, Capital, and population on the page.

#### **46-6 (advanced) Pass multiple data to a child component**

* When we need to pass multiple data to a child component, we can send the entire object and access it in the child component using **props.objectName**. Use destructuring for easy access and use of the data.

#### **46-7 (advanced) add unique key for mapping components**

* For avoiding this warning - Each child in a list should have a unique "key" prop - in the console, we have to add a property named ‘**key’** and the value will be unique for every element.
* This warning commonly showed when we passed multiple props using the **map**.

#### **46-8 build your react app and host your react app in Netlify**

* To make ready the react-app for deployment, we have to run ‘*npm run build’*
* The deployment-ready files will be store in build bolder. Upload this folder to Netlify for deploying the site. If it needs to update, make the necessary change, re-run the build command and re-upload the newly created build folder. Learn more about [react deployment](https://create-react-app.dev/docs/deployment).

#### **46-9 Module Summary and recap**

* We can add **Bootstrap** to react projects in various ways. A simple way is just to copy the Bootstrap CDN link and paste it into the **public/index.html** file. We can now use the Bootstrap **class** in any **Component**.