COURSEWARE

Professional Skills

Agile Fundamentals

Jira

Git

Databases Introduction

Java Beginner

Maven

Testing (Foundation)

Java Intermediate

HTML

CSS

Javascript

Spring Boot

Selenium

Sonarqube

Advanced Testing (Theory)

Cucumber

MongoDB

Express

NodeJS

React

○ Introduction
○ JSX
○ Babel
○ Component Hierarchy
○ Components
○ Props
○ Lifecycle
○ State
○ Lifting State
○ Hooks
○ React Routing

Babel

Contents

Overview

In this module, we will be looking at Babel.

What is Babel?

Babel is a JavaScript Compiler

Babel is a tool-chain that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments.

In other words - Babel is a compiler (source code => output code).

Here are the main things Babel can do for you:

- Transform syntax
- Polyfill features that are missing in your target environment (through `@babel/polyfill`)
- Source code transformations

Let's have a look at how Babel converts JavaScript code:

```
// Babel Input: ES2015+ arrow function
[1,2,3].map((n) => n+1);

// Babel Output: ES5 Equivalent
[1,2,3].map(function(n){
    return n+1;
});
```

ES2015 and Beyond

Babel has support for the latest version of JavaScript through *syntax transformers*.

These plugins allow you to use new syntax, **right now** without having to wait for browser support!

Pluggable

Babel is built out of plugins. With Babel it is possible to compose custom transformation pipelines using plugins or writing your own. We can also use a set of plugins by using or creating **presets**

References

For more information Refer to Babel's Documentation;

Tutorial

○ Data Requests

○ Static Data

○ State Management

Express-Testing

Networking

Security

Cloud Fundamentals

AWS Foundations

AWS Intermediate

Linux

DevOps

Jenkins Introduction

Jenkins Pipeline

Markdown

IDE Cheatsheet

1. Using `npm` import the `preset-react` package from `@babel`

   ```
   npm install --save-dev @babel/preset-react
   ```

2. Add `@babel/preset-react` to your `package.json`

3. Create a class which uses a random number generator to generate a number on a die.

   ```
   const Dice = () => {
       function getRandomNumber(){
           return Math.ceil(Math.random() * 6);
       }

       const num = getRandomNumber();

       return ( <p>Your dice roll: {num}</p> );
   }

   export default Dice;
   ```

4. Paste the above code into Babel's online Editor and observe the output.

Exercises

There are no exercises for this module.