ReactTS Best practice 5-day project based course

V1 – November 2024

# General delivery plan

This section explains generally how the course can be delivered over four days.

|  |  |
| --- | --- |
| **DAY 1** | |
| 09:30 | Introduction – Info on how the course will run |
| 1 – Review.pptx – Installing via Vite and the files created |
| 11:00 | *Break* |
| 11:15 | Components, JSX, demo and concept of Pure Components |
| 12:30 | *Lunch* |
| 13:30 | Props in React – Demo of best practices.  Useful Techniques  Component lifecycles  Component Hierarchies  Forms + Libraries(react-hook-form) |
| 15:00 | *Break* |
| 15:15 | JSON Server + Demo of Single page Blog |
| 16:30 | *Finish* |
| **DAY 2** | |
| 09:30 | Recap and review of blog sites created on Day 1  Introduction to Routing  2-Routing.pptx  Encased around the Helpdesk App  RouteProvider  Layouts |
| 11:00 | *Break* |
| 11:15 | Continue with Helpdesk App –  Custom 404  Loaders |
| 12:30 | *Lunch* |
| 13:30 | Route Parameters  ErrorElements |
| 15:00 | Break |
| 15:15 | Protected Routes |
| 16:30 | *Finish* |
| **DAY 3** | |
| 09:30 | Project time (at the end of presentation 2) |
| 11:00 | *Break* |
| 11:15 | 3 – Hooks.pptx  useState()  useEffect() |
| 12:30 | *Lunch* |
| 13:30 | useContext()  useReducer()  useRef()  useCallback() |
| 15:00 | *Break* |
| 15:15 | useMemo()  custom hooks  challenges - Individual |
| 16:30 | *Finish* |
| **DAY 4** | |
| 09:30 | Recap on previous days learning  4 – state Management.pptx  useContext Demo with Flower Shop |
| 11:00 | *Break* |
| 11:15 | Continue flowerShop work |
| 12:30 | *Lunch* |
| 13:30 | Redux – Discussion and Walkthrough  Redux Vs Context API |
| 15:00 | *Break* |
| 15:15 | Caching Options in React  React-query demo |
| 16:30 | *Finish* |
| **DAY 5** | |
| 9:30 | Recap on Previous day learning  5 – Patterns.pptx  Performance Demo at end of first section |
| 11:00 | *Break* |
| 11:15 | 5 – Patterns.pptx  Common Design Patterns + Anti Patterns |
| 12:30 | *Lunch* |
| 13:30 | Introduction to final exercise with QAEState Agents |
| 15:00 | *Break* |
| 15:15 | QAEState Agents |
| 16:30 | *Finish* |

# Notes on delivering this course

This course is designed to be as practical as possible. The delegates should be familiar with React, but perhaps not new techniques of creating projects. The presentations contain information and snips of the main information needed, but the course should be completed via demonstrations and explanations throughout.

In each of the day folders, there are examples of working versions of each of the challenges that have been created. Day 1, for example has a rudimentary blog which pulls and puts data into a JSON file via a running JSON server.

There are many challenges, which are reflected by completed react projects. As the trainer delivering the content, it is completely at your discretion as to how you demonstrate these.

Towards the end of the week, it may be pertinent to include demonstrations with GitHub CoPilot in VSCode as we can expedite the process. Throughout the week, we will use the explain as much as possible.

Repo is available [here](https://github.com/AndySmithQA/QAADVREACTTS)

Day 1 Blog Demo

It would be good to follow this pattern while completing the Demo – Ensuring that the delegates are aware that there is a set way of building React apps/webapps which allow for the best outcomes.

1. Create a data folder which contains the db.json folder and demonstrate how to run the server

npx json-server --watch data\db.json --port 8000

1. Create a simple Navbar.tsx component which has a simple header (no routes or links at this point). Export it an put it in App.tsx.
2. Create a Home.tsx file which creates a const of a url where the data is accessed and create a simple state of [blogs, setBlogs] = useState(null)

In this component, create a useEffect(not too much details as there is a hooks day) to fetch the blogs from the URL

1. In the JSX return, call blogs component (to be created) using conditional rendering

{blogs && <Blogs blogs={blogs} />}

1. Create a Blogs.tsx file and map the blogs as props to show just the title and the author of the blogs in the JSON file.

Once they are pulled into the file successfully, add a form in the Home.tsx which allows the user to create a new blog (at this stage, a fetch and ‘POST’ method is easiest)

You will need to force a reload of the window at this stage, which will eventually be replaced by a redirect when routing is introduced.

Day 2 Helpdesk Demo

This demo follows some of the latest advice on routing in react-router-dom v6.4. This is not the latest release, but was a major update which has been improved with additional hooks in subsequent releases.

This demo uses RouteProvider (very similar in application to useContext).

It uses nesting and layouts to create specific routes

Day 2 Protected Routes Demo

The repo has skeleton and fully working code. The order in which the code should be as follows:

1. Build the Authentication Wrapper
2. Update RenderNavigation()
   1. RenderRoutes first
   2. RenderMenu Second
3. Update the Login component
4. Update the Account Page

It uses another method, called BrowserRouter, with all routes wrapped in authentication.

It uses hooks which have not yet been discussed with the delegates such as useContext(), useNavigate() etc, but these are all routing specific hooks.

Day 3

The delegates should start the day with the first session in pairs, looking at the start of the shop challenge. (Outlined at the end of presentation 2) This will be extended throughout the week and after more learning, but a good chunk of time with them working together to plan and prep repos etc.

After break, start to look at the third presentation about hooks. As always, the tutor should be demonstrating these hooks through coding and discussion. Encouraging the delegates to try and code along and dig out understanding as you go is essential at this point.

It is likely that they would have used some of these before, but probably have a surface level understanding of how and why the work how they do.

Time should be taken to spend the day coding and discussing hooks. Culminating in the discussion of Custom hooks. There is a demonstration of how to go about creating a custom hook.

Day 4

Following on from the flower shop project, the repo can be used to demonstrate how you might implement a cart with Context. Notice 2 branches exist one with working cart and one completely without.

The next section requires the install and use of the redux-toolkit and react-redux.

The demo and instruction show how to build a store and wrap the App in the state management tools. It also includes content on the actions.

A short section on when to use Context API or Redux is followed by an overview on how react-query works.

In the repo, you will see 2 branches again, a starter and the final working app. (called tandemo as TanStack are the organisation responsible for updating the library).

V1.2 – an update to this section has added more information on Redux with guidance on how to install and use the developer tools in the browser with Redux

Day 5

As extension, there has been a new unit created which looks at the use of Google Firebase as a method to externally authorise users and create user accounts within your app. The final working code is [here](https://github.com/AndySmithQA/fireBaseAuth). There is also a presentation to compliment the content in the folders called 5-Authentication.pptx.

The final day focuses on pulling all the learning together. The presentation is based around identifying where the patterns of good coding have been used throughout the week.

There is a demonstration which shows the performance difference between child components and where they are called. This is a complex demo and will need to be discussed while it is being coded.

The final challenge will be to show the delegates the [QAEstateAgents](https://github.com/AndySmithQA/QAEstate) website and have them try to duplicate the way it works. This will be a significant challenge and will likely take up the remainder of the day.