COMP6080
Web Front-End Programming

Testing - Unit Tests, Component Tests

Component Testing

Component testing is a process of unit testing where we ensure that our Framework's components, regardless of how they're integrated into the broader app, function as expected.

Concepts surrounding unit & component testing were discussed in the testing introduction lecture.

Testing ReactJS

There are two main libraries we will use to test ReactJS components:

1. Jest - a Javascript unit testing framework
2. Enzyme - A unit testing framework specific to ReactJS, that further extends our capabilities

Testing ReactJS - Installation

Thankfully, Create-react-app (CRA) installs many of the dependencies related to Jest as part of it's installation. If using CRA, there are two extra steps you need to take:

```
1 $ yarn add --dev react-test-renderer # For the renderer
2 $ yarn add --dev enzyme enzyme-adapter-react-16 # For installation of Enzyme
```

Testing Element Example

Let's take a look at a basic black-box test for a simple ReactJS component. We are expected "title" to be optional with a fallback of 'Click Me!'

```
export const Button = ({ onClick, title = 'Click Me!', mode }) => {
  const _class = mode === 'dark' ? 'dark-mode' : 'light-mode';
  return (
    <button onClick={onClick} className={_class}>
      {title}
    </button>
  )
};
```

Further Testing Abilities

When writing tests in COMP6080, the two easiest places to look are:

- Shallow Rendering
- Full DOM Rendering

Shallow rendering is used when you don't need any children components loaded to properly test your component.