



REACT EXAMPLE CODE

```
import { useState, useEffect }
  from "react";
import "./App.css";
import sendIcon from
  "./images/envelope.png";
import Button from
  "./components/Button";

function App() {
  // Define starting state
  const [message, setMessage] =
    useState("");
  const [chatLog, setChatLog] =
    useState([]);

  // Form elements need state
  // modifying functions
  function onMsgChange (ev) {
    setMessage(ev.target.value);
  }

  function sendMsg (ev) {
    setChatLog(
      [message, ...chatLog]);
  }

  // When you need something to
  // happen after first render
  useEffect(() => {
    alert("First render!")
  });

  // When you need something to
  // happen after every render
  useEffect(() => {
    alert("Just rendered!")
  }, []);

  // When you need something to
  // happen after every time
  // the chatLog state changes
  useEffect(() => {
    alert("chatLog changed!")
  }, [chatLog]);

  const count = chatLog.length;
  return (

    <div className="App">
      <h1>{count} messages</h1>
      {chatLog.map(text => {
        return <p>{text}</p>
      })}
      <input
        value={message}
        onChange={onMsgChange} />
      <Button onClick={sendMsg}>
        <img src={sendIcon} />
        Send message
      </Button>
    </div>
  );
}
```

DEFINING COMPONENTS

```
function Button(props) {
  return (
    <button className="Button"
      onClick={props.onClick}>
      {props.children}
    </button>
  )
}

export default Button;
```

USEFUL REACT PATTERNS

Conditional rendering:

```
if (!props.text) {
  return "Empty...";
}

return (
  // full render here ...
)
```

Using map to loop through data:

```
<div>{
  data.map((item, i) => (
    <p onClick={pClicked}>
      {i}: {item}
    </p>
  ))
}</div>
```

Using ternary operator:

```
<div>{
  props.image ? (
    <img src={props.image} />
  ) : (
    <em>No image provided.</em>
  )
}</div>
```

Using ref to incorporate legacy JS:

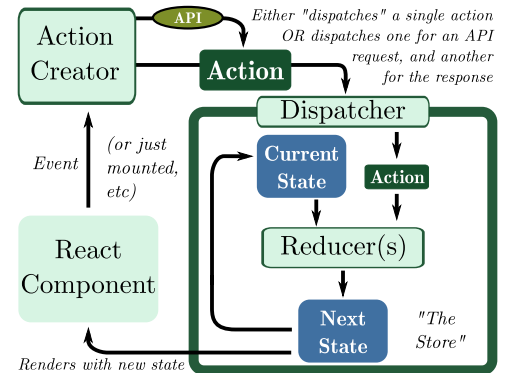
```
// At the top of the component
const [el, setEl] = useState(null);

// In a useEffect call
useEffect(() => {
  $(el).somePlugin();
}, [el]);

// Somewhere in JSX
<div ref={setEl}>Click me!</div>
```

REACT, ROUTER, REDUX

REACT REDUX



Action Creators (found in actions/)

```
const doIncrement = () =>
  ({type: INCREMENT});
const addTodo = (item) =>
  ({type: ADD_TODO, text: item});
```

Dispatching (found in components/)

```
let action =
  addTodo(text);
dispatch(action);
```

Reducers (found in reducers/)

```
const initialState = {
  count: 0,
  todoList: [],
};
const todo = (state, action) => {
  switch (action.type) {
    case INCREMENT:
      return {...state, ...{
        count: state.count + 1,
      }};
    case ADD_TODO:
      return {...state, ...{
        todoList: [...todoList,
          action.text]},
      }); /* etc ... */
  }
}
```

REACT ROUTER

```
<nav>
  <Link to="/about/">About</Link>
  <Link to="/post/"+postId+"/">
    Read More...</Link>
</nav>
<main>
  <Switch>
    <Route path="/about/"
      component={About} />
    <Route path="/post/:id/"
      component={BlogPost} />
  </Switch>
</main>
```