



Diagram illustrating the structure of a Redux reducer function. The code is as follows:

```
let countReducer = (state, action) => {  
  switch(action.type) {  
    case "INCREMENT":  
      return {count: state.count + action.payload};  
    case "DECREMENT":  
      return {count: state.count - 1};  
    default: return state;  
  }  
}
```

Annotations:

- A blue arrow points down to the opening curly brace of the function.
- A blue arrow points down to the opening curly brace of the switch statement.
- A blue arrow points from the closing curly brace of the switch statement to the closing curly brace of the function.
- A red arrow points from the closing curly brace of the switch statement to the closing curly brace of the function.
- A blue squiggle is under the word "default".

dispatch → delegate action to

```
initialState = {  
  count : 0
```

```
function App() {  
  let [state, dispatch] = useReducer(countReducer, initialState);
```

```
    function handleIncrement() {  
      dispatch({type: "INCREMENT", payload: 10});
```

```
    }  
  
    function handleDecrement() {  
      dispatch({type: "DECREMENT"});  
    }
```

reducers



