

```

    ),
    item: '',
  });
};

```

最后看按钮：

testing/react-basics/src/App.js

```

<button
  className='ui button'
  type='submit'
  disabled={submitDisabled}
>
  Add item
</button>

```

我们在按钮上设置了 `disabled` 属性。`submitDisabled` 变量在 `render()` 函数的顶部定义，它的值取决于输入字段是否被填充：

testing/react-basics/src/App.js

```

render() {
  const submitDisabled = !this.state.item;
  return(

```

8.5.3 App 组件的第一个用例

为了编写第一个用例，需要有两个库：Jest 和 Enzyme。

在上一章中，我们注意到 `create-react-app` 在 `package.json` 中设置了一些命令，其中一个就是 `test`。

`react-scripts` 已将 Jest 指定为依赖项。要启动 Jest，只需运行 `npm test` 命令。与 `create-react-app` 创建的其他命令一样，`test` 在 `react-scripts` 中运行了一个脚本。该脚本配置并执行 Jest。



若要查看 `react-scripts` 包含的所有包，请看 `./node_modules/react-scripts/package.json` 文件。

`create-react-app` 在 `App.test.js` 中为我们设置了一个虚拟测试。下面从 `testing/react-basics` 文件夹内部执行 Jest，并看看会发生什么。

```
$ npm test
```

Jest 运行，生成了一个格式良好的测试套件的结果报告（见图 8-9）。