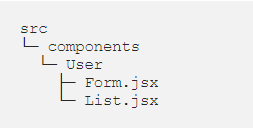
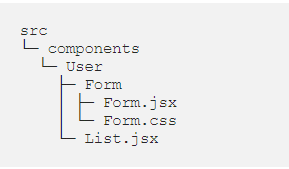
**Разделение и группировка кода**

Внутри папки *components* мы группируем файлы по модулям.

Когда компонент состоит из более чем одного файла, мы кладем этот компонент и его файлы в папку с тем же именем. Например: вы имеете Form.css, содержащий Form.jsx стили. В этом случае структура будет выглядеть так:

*!!!*

*Тестовые файлы остаются с проверяемым файлом. В приведенном выше случае, тест для****Form.jsx****останется в той же папке и будет называться****Form.spec.jsx***

*!!!*

## Компоненты пользовательского интерфейса

Помимо разделения компонентов на модули, добавим папку *UI* в *src/components*, чтобы сохранить в ней наши общие компоненты.

UI Components это настолько общие компоненты, что они не принадлежат конкретному модулю. Эти компоненты вы можете выделить в отдельную библиотеку, потому что они не имеют бизнес логики. Примером таких компонентов могут быть: Buttons, Inputs, Checkboxes, Selects, Modals, Data display elements, и т.д.

## Именование компонентов

*Когда мы говорим об именовании компонента, это относится к имени, которое мы даем****классу****или****переменной****, определяющей компонент:*

*class****MyComponent****extends Component {  
}  
const****MyComponent****() => {};*

Имена компонентов должны быть понятны и уникальны для всего приложения, чтобы сделать их легче для поиска и избежать возможных неточностей.

Имена компонентов очень удобны для отладки, когда мы используем инструменты, такие как React Dev Tools, и когда происходят run time errors. Ошибки всегда содержат имя компонента, в котором они произошли.

При именовании компонентов мы следуем паттерну path-based-component-naming, который содержит в названии компонент соответствующий его относительному пути для папки *components* или *src*, если вы находитесь вне папки *components*. Например компонент находящийся в папке *components/User/List.jsx* должен быть назван *UserList.*

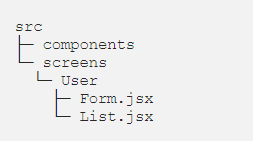
Если файл находится в папке с таким же именем, то нам не нужно повторять его. То есть *components/User/Form/Form.jsx* должен быть назван *UserForm*, а не *UserFormForm*.

## Экраны

Название говорит само за себя, будем называть им экраны в нашем приложении.

Экран — это место, где вы используете компоненты, для составления их в страницу. В идеале экран не должен хранить никакой логики и должен быть функциональным компонентом.

Мы сохраняем экраны в отдельной папке, находящейся в *src*, потому что они будут сгруппированы в соответствии с маршрутом, а не модулем:



**Router**

Объясню на примере: Мы хотим, чтобы по ссылке mavka/znoMatan/ запускался компонент ZnoMatan.

Открываем файл Root.jsx и прописываем следующие:

import ZnoMatan from PATH to ZnoMatan(e.g. './User/Form');

<Route path="/znoMatan" component={ZnoMatan} />

Больше информации тут: [тык](https://medium.com/@timafon/%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0-%D0%B8-%D0%B8%D0%BC%D0%B5%D0%BD%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2-%D0%B2-react-2a29e96e0e07)

**Импорт компонентов**

Одним из недостатков этой структуры каталогов является то, что для импорта компонентов требуется импортировать полный путь, например:

**import** Login **from 'components/Login/Login'**

Но мы хотели бы написать следующее:

**import** Login **from 'components/Login'**

Решение, с которым мы столкнулись, использует стандартный разрешающий механизм Node.js, что делает его надежным. Нам лишь нужно добавить файл package.json в файловую структуру.

И внутри package.json мы используем основное свойство, чтобы установить нашу точку входа в компонент, например:   
{ **"main"**: **"Login.js"** }

С этим дополнением мы можем импортировать компонент следующим образом:

**import** Login **from 'components/Login'**

**Assets**

В assets храним практически любые медиа файлы, от файлов изображений, таких как jpg и gif, до видео-файлов, таких как wmv и mov, и, конечно, стандартных файлов документов, таких как ppt и docs.

CSS стили храним в самой компоненте в components