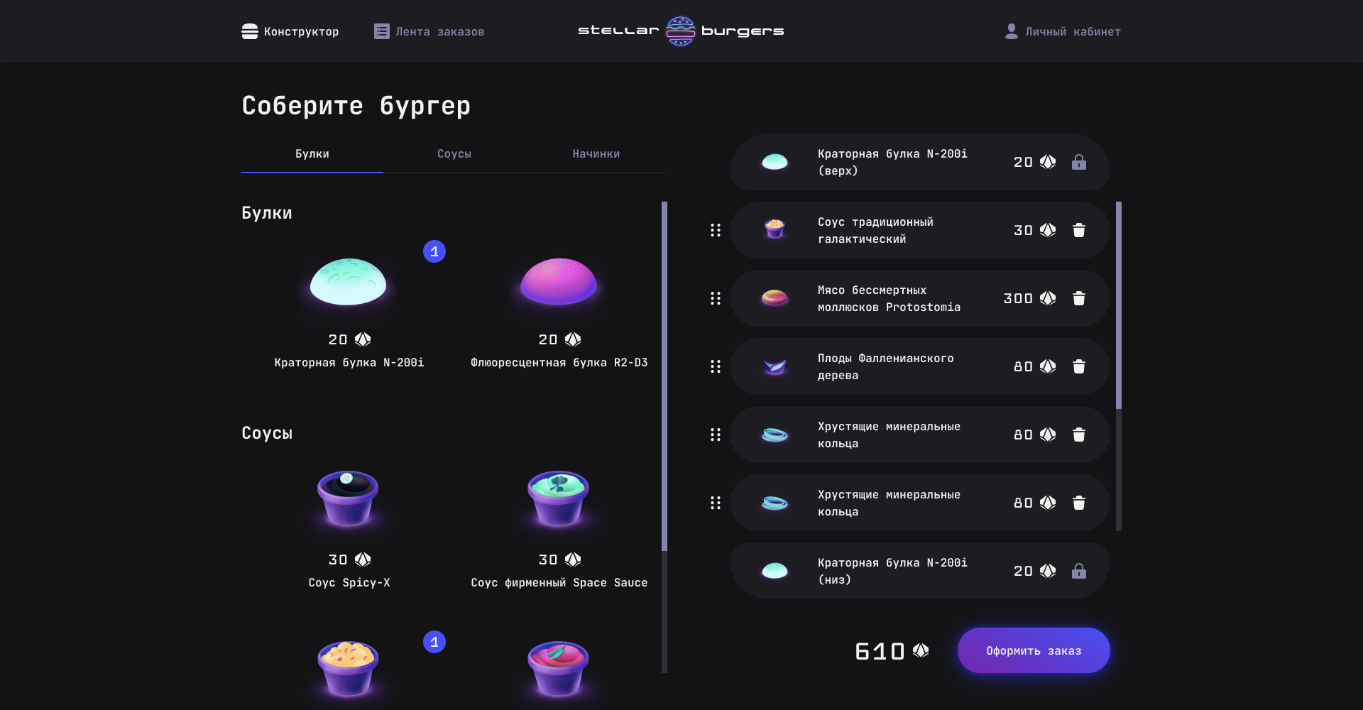
**Проект. Создать CRA-заготовку и создать структуру папок под компоненты**

Приступим к разработке нового приложения. В этой проектной работе вам предстоит создать главную страницу бургерной. Ознакомиться с макетом можно [по ссылке](https://www.figma.com/file/vejDm3dVTUor3wBdNO137u/React-%2F-%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%BD%D1%8B%D0%B5-%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8?node-id=0%3A1). А вот изображение главной страницы:



*Главная страница “Stellar Burgers”*

В этой проектной работе вы организуете инфраструктуру проекта и сверстаете несколько компонентов:

* создадите новый проект;
* установите дополнительные пакеты;
* инициализируете git-репозиторий;
* реализуете компоненты главной страницы.

Начнём с простого — создания нового проекта на CRA.

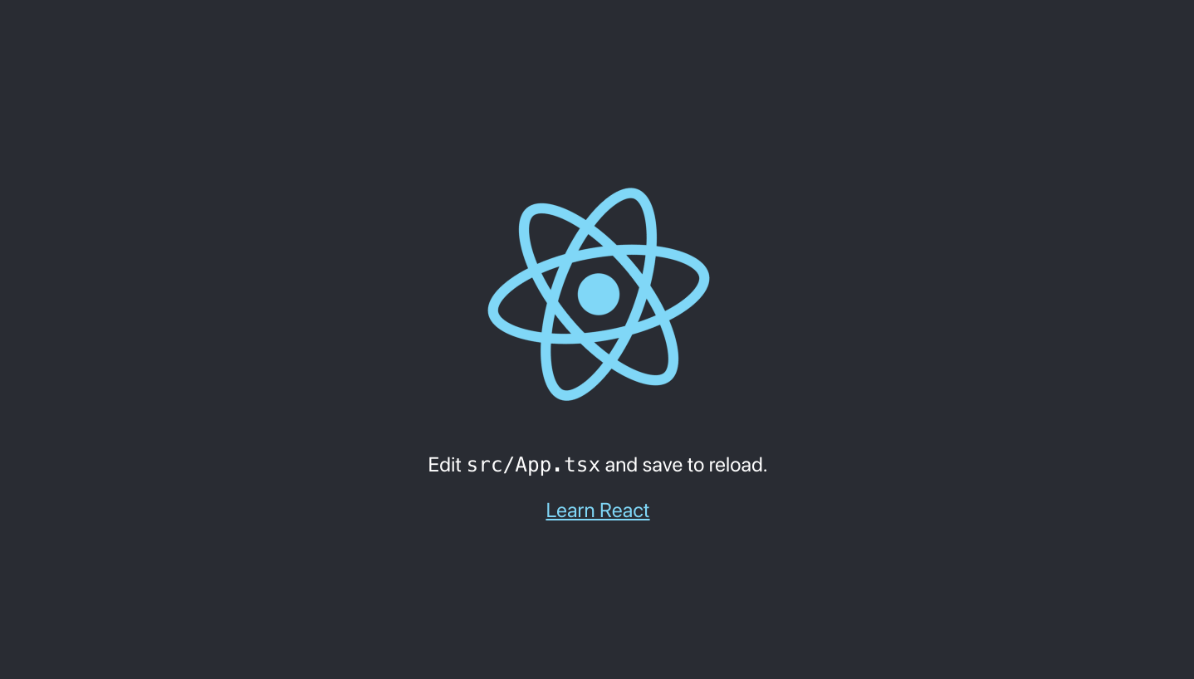
**1. Создание нового проекта на CRA**

Установите CRA глобально. После — инициализируйте новый CRA-проект react-burger. Воспользуйтесь следующей командой:

Скопировать кодJSX

npx create-react-app react-burger --template typescript

Для создания нового приложения на CRA с использованием TypeScript укажите флаг --template со значением typescript. В этой работе он вам не пригодится, но важно заранее заложить фундамент для комфортного перехода на TypeScript. После того как создадите новый проект и запустите его, вы увидите такой экран:



*Стартовый экран проекта на CRA*

Это не все инфраструктурные задачи для вашего проекта. Подробнее расскажем о них в следующих разделах.

**2. Установка библиотеки UI-компонентов**

Вёрстку некоторых маленьких (и не очень) компонентов мы взяли на себя, а вы сможете сосредоточиться на реализации основной логики приложения. Воспользуйтесь нашим [готовым набором компонентов для бургерной](https://www.npmjs.com/package/@ya.praktikum/react-developer-burger-ui-components). Установите npm-пакет как зависимость в приложение:

Скопировать кодBASH

npm i @ya.praktikum/react-developer-burger-ui-components

Мы подготовили документацию с описанием каждого UI-компонента, его пропсами и внешним видом. Ознакомиться с документацией можно [по ссылке](https://yandex-praktikum.github.io/react-developer-burger-ui-components/docs/).

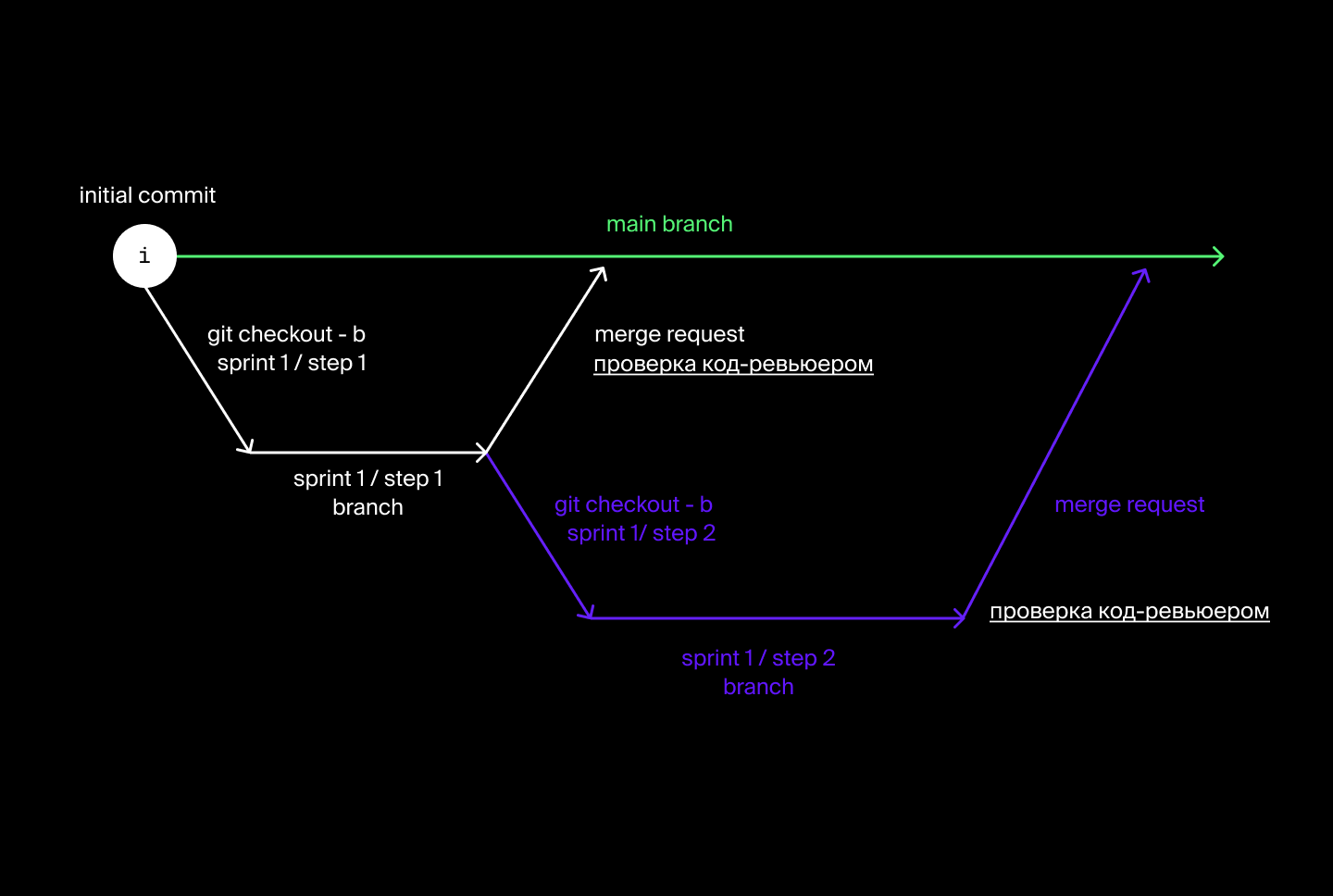
В этой проектной работе мы подскажем, какие компоненты библиотеки нужно использовать. А дальше вам предстоит самостоятельно определять нужные компоненты. Теперь поговорим про git и ветвление.

Нужно импортировать из библиотеки файл в index.css: @import '~@ya.praktikum/react-developer-burger-ui-components/dist/ui/box.css'; После этого можно использовать классы вида mt-1 в файлах css модулей с помощью composes:*.tabs* {  
    composes: mb-10 mt-5 from global;  
    display:flex;  
    flex-grow: 0;  
}Либо прямо в коде: className={`${styles.tabs} mt-10 mt-5`}

**3. Настройка и ведение git**

Инициализируйте git в проекте. Сделайте первый коммит в текущей ветке — это изначальное состояние вашего проекта. Другими словами — первое стабильное состояние проекта, поэтому дальнейшую разработку проекта ведите в других ветках. Создайте пустой репозиторий на сервисе GitHub. Настройте удалённый и локальный репозитории с помощью git remote add.

В основную ветку вливайте проверенные код-ревьюером изменения — следующие стабильные состояния проекта. Если планируете отправлять работу на промежуточную проверку, то придерживайтесь такого ведения веток:



*Ветвление для сдачи работы на проверку в мягкий и строгий дедлайн*

Обратите внимание, что на проверку нужно отправлять ссылку на созданный открытый merge request. Подробнее о сдаче проектной работы мы расскажем в самом конце урока, а пока перейдём к файловой структуре проекта.

**4. Организация файловой структуры**

При организации файловой структуры мы рекомендуем пользоваться следующими правилами:

* Хранить компоненты в директории components/, шрифты — в fonts/, изображения — в images/.
* Каждый компонент хранить в отдельной директории.
* Для стилизации компонентов использовать CSS-модули.
* Называть директории и файлы компонентов в нотации kebab-case.

Мы рассказали про все инфраструктурные задачи проектной работы. Теперь поговорим о компонентах.

**5. Вёрстка первого экрана проекта. PropTypes**

Ваша задача — сверстать несколько компонентов главного экрана:

* AppHeader — шапка приложения;
* BurgerIngredients — список ингредиентов;
* BurgerConstructor — текущий состав бургера.

Подробнее поговорим о каждом компоненте.

**AppHeader**



*Внешний вид шапки*

Из библиотеки UI-компонентов возьмите следующие:

* логотип,
* иконки,
* типографику,
* систему отступов.

Остальную вёрстку выполните самостоятельно.

**BurgerIngredients**



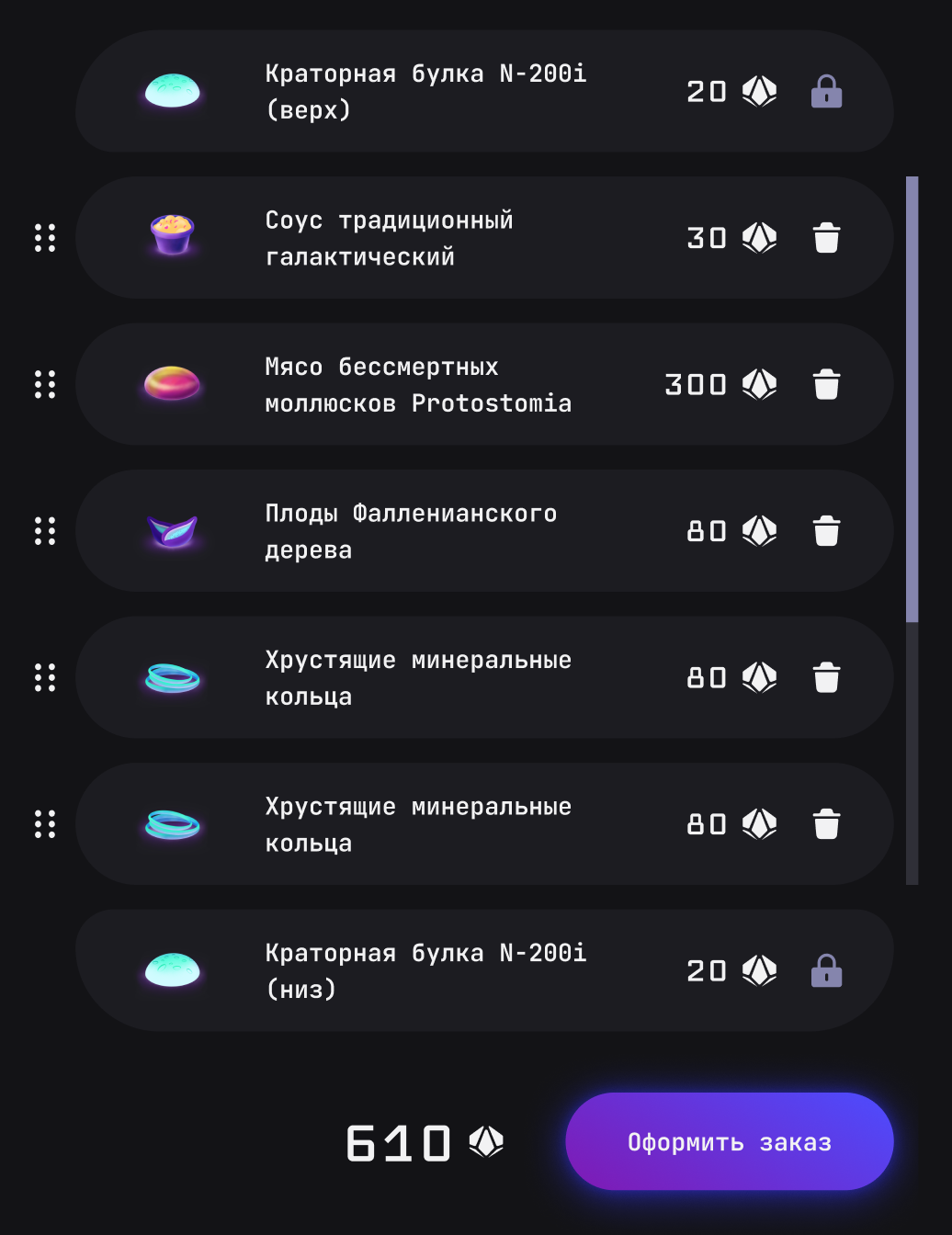
*Внешний вид и составляющие компонента BurgerIngredients*

Из библиотеки UI-компонентов возьмите следующие:

* счётчики,
* иконки,
* переключатели,
* типографику,
* систему отступов.

У компонента свой кастомизированный скроллбар. Подумайте над реализацией и возможным ограничением высоты блока, в том числе и на разных разрешениях экранов.

**BurgerConstructor**



*Внешний вид и составляющие компонента BurgerConstructor*

Из библиотеки UI-компонентов возьмите следующие:

* элементы списка,
* иконки,
* кнопку,
* типографику,
* систему отступов.

Отображение списка организуйте самостоятельно. Подумайте над реализацией и возможным ограничением высоты блока, в том числе и на разных разрешениях экранов. Скроллбар не распространяется на заблокированные позиции конструктора.

**Несколько советов по вёрстке**

Мы изучаем React, но базовые правила семантики вёрстки остаются актуальными. Чтобы им следовать, рекомендуем:

* использовать семантические HTML5-элементы, которые точно располагаются на главной странице: header, nav, main, section;
* оформлять повторяющиеся элементы в неупорядоченные списки;
* стараться не использовать элемент div. Он подходит исключительно для «упаковки» нескольких блоков с их последующей стилизацией и построением сетки.

**Данные**

Компоненты BurgerConstructor и BurgerIngredients используют данные. Пока вам не нужно делать запрос к API, поэтому ниже есть массив с захардкоженными данными, которыми можно воспользоваться на уровне компонента App:

Скопировать кодJSON

[

{

"\_id":"60666c42cc7b410027a1a9b1",

"name":"Краторная булка N-200i",

"type":"bun",

"proteins":80,

"fat":24,

"carbohydrates":53,

"calories":420,

"price":1255,

"image":"https://code.s3.yandex.net/react/code/bun-02.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/bun-02-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/bun-02-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b5",

"name":"Говяжий метеорит (отбивная)",

"type":"main",

"proteins":800,

"fat":800,

"carbohydrates":300,

"calories":2674,

"price":3000,

"image":"https://code.s3.yandex.net/react/code/meat-04.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/meat-04-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/meat-04-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b6",

"name":"Биокотлета из марсианской Магнолии",

"type":"main",

"proteins":420,

"fat":142,

"carbohydrates":242,

"calories":4242,

"price":424,

"image":"https://code.s3.yandex.net/react/code/meat-01.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/meat-01-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/meat-01-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b7",

"name":"Соус Spicy-X",

"type":"sauce",

"proteins":30,

"fat":20,

"carbohydrates":40,

"calories":30,

"price":90,

"image":"https://code.s3.yandex.net/react/code/sauce-02.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/sauce-02-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/sauce-02-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b4",

"name":"Мясо бессмертных моллюсков Protostomia",

"type":"main",

"proteins":433,

"fat":244,

"carbohydrates":33,

"calories":420,

"price":1337,

"image":"https://code.s3.yandex.net/react/code/meat-02.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/meat-02-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/meat-02-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b9",

"name":"Соус традиционный галактический",

"type":"sauce",

"proteins":42,

"fat":24,

"carbohydrates":42,

"calories":99,

"price":15,

"image":"https://code.s3.yandex.net/react/code/sauce-03.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/sauce-03-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/sauce-03-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b8",

"name":"Соус фирменный Space Sauce",

"type":"sauce",

"proteins":50,

"fat":22,

"carbohydrates":11,

"calories":14,

"price":80,

"image":"https://code.s3.yandex.net/react/code/sauce-04.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/sauce-04-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/sauce-04-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9bc",

"name":"Плоды Фалленианского дерева",

"type":"main",

"proteins":20,

"fat":5,

"carbohydrates":55,

"calories":77,

"price":874,

"image":"https://code.s3.yandex.net/react/code/sp\_1.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/sp\_1-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/sp\_1-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9bb",

"name":"Хрустящие минеральные кольца",

"type":"main",

"proteins":808,

"fat":689,

"carbohydrates":609,

"calories":986,

"price":300,

"image":"https://code.s3.yandex.net/react/code/mineral\_rings.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/mineral\_rings-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/mineral\_rings-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9ba",

"name":"Соус с шипами Антарианского плоскоходца",

"type":"sauce",

"proteins":101,

"fat":99,

"carbohydrates":100,

"calories":100,

"price":88,

"image":"https://code.s3.yandex.net/react/code/sauce-01.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/sauce-01-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/sauce-01-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9bd",

"name":"Кристаллы марсианских альфа-сахаридов",

"type":"main",

"proteins":234,

"fat":432,

"carbohydrates":111,

"calories":189,

"price":762,

"image":"https://code.s3.yandex.net/react/code/core.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/core-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/core-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9be",

"name":"Мини-салат Экзо-Плантаго",

"type":"main",

"proteins":1,

"fat":2,

"carbohydrates":3,

"calories":6,

"price":4400,

"image":"https://code.s3.yandex.net/react/code/salad.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/salad-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/salad-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b3",

"name":"Филе Люминесцентного тетраодонтимформа",

"type":"main",

"proteins":44,

"fat":26,

"carbohydrates":85,

"calories":643,

"price":988,

"image":"https://code.s3.yandex.net/react/code/meat-03.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/meat-03-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/meat-03-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9bf",

"name":"Сыр с астероидной плесенью",

"type":"main",

"proteins":84,

"fat":48,

"carbohydrates":420,

"calories":3377,

"price":4142,

"image":"https://code.s3.yandex.net/react/code/cheese.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/cheese-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/cheese-large.png",

"\_\_v":0

},

{

"\_id":"60666c42cc7b410027a1a9b2",

"name":"Флюоресцентная булка R2-D3",

"type":"bun",

"proteins":44,

"fat":26,

"carbohydrates":85,

"calories":643,

"price":988,

"image":"https://code.s3.yandex.net/react/code/bun-01.png",

"image\_mobile":"https://code.s3.yandex.net/react/code/bun-01-mobile.png",

"image\_large":"https://code.s3.yandex.net/react/code/bun-01-large.png",

"\_\_v":0

}

]

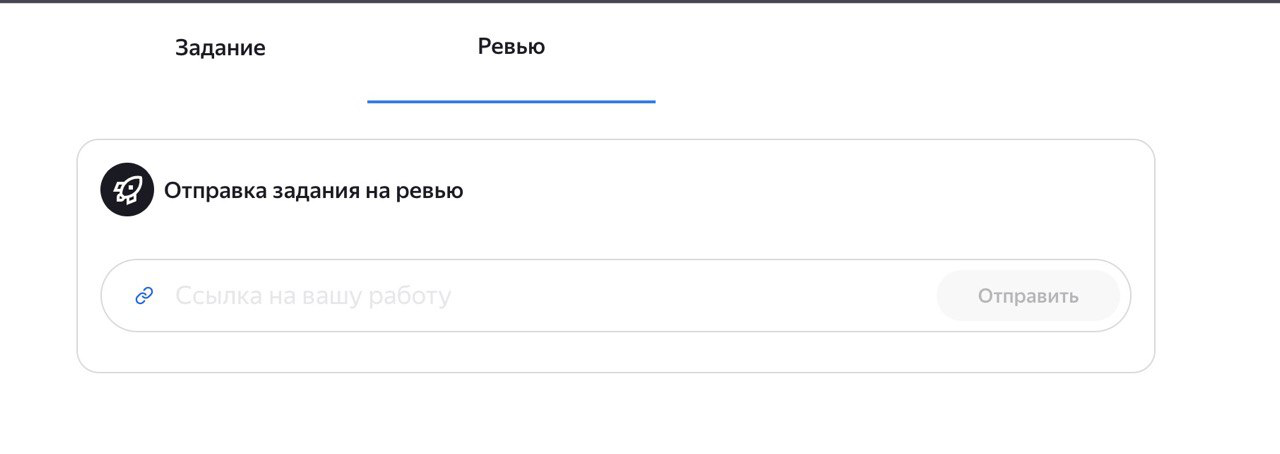
Чтобы эти данные не занимали много места в компоненте App — положите их в отдельный файл data.js директории utils/.

**Проверка типов данных компонентов**

Сейчас вы используете тестовые данные. Но уже скоро вы замените их на настоящие, поэтому важно сразу озаботиться проверкой типов данных. Опишите проверку типов данных propTypes компонентов BurgerConstructor и BurgerIngredients.

**6. Первая отправка работы на проверку**

Когда будете готовы показать код-ревьюеру промежуточный результат, отправьте ссылку на пул реквест, открытый из ветки sprint-1/step-1, в main через интерфейс Практикума. Отправить ссылку можно в отдельном уроке, который доступен [по ссылке](https://practicum.yandex.ru/learn/react/courses/511eb872-b65e-42c8-bbc2-0443bfe038d5/sprints/9583/topics/5a7e0e00-c33c-4c33-b001-c074c558c58e/lessons/c812eafa-a8f6-4357-a529-6f4ec243c1c4/). В этом уроке будет находиться интерфейс промежуточной сдачи проектной работы.



*Окно отправки проектной работы. Вставьте сюда ссылку на пул реквест*

Обратите внимание, что нужно прикладывать ссылку именно на пул реквест, а не на репозиторий. Как только работа будет зачтена, нажмите кнопку “Merge”, чтобы применить изменения из ветки sprint-1/step-1 в main.