



Чек-лист по самопроверке №4

В этом документе описаны критерии, которым должна соответствовать работа. Перед отправкой работы на ревью, убедитесь, что она соответствует всем критериям.

Работа отклоняется от проверки

Если не соблюдены хотя бы один из критериев этого блока, ревьюеры не станут проверять работу.

- Пул-реквест не отправлен на проверку.
- Проект не собирается или не запускается.

Работа принимается

В этом блоке собраны требования к итоговой работе, по которым вы можете выявить и самостоятельно исправить частые ошибки:

Общее

- Функциональность, требуемая по заданию предыдущей проектной работы, реализована корректно и продолжает работать. Её описание вы можете найти [в чек-листе](#)
- Пул-реквест создан из ветки `sprint-4/step-1` в `main`.
- Сборка и запуск проекта выполняются без ошибок.
- В проекте есть:
 - директория `components` с компонентами и стилями: `App`, `AppHeader`, `BurgerIngredients`, `BurgerConstructor`, `Modal`, `ModalOverlay`, `OrderDetails`, `IngredientDetails`;
 - если в проекте используются сторонние шрифты или изображения, то они хранятся в папках `fonts` и `images`;
 - единая директория `services` с вложенными `actions` и `reducers` со всей логикой приложения, либо `services` с `actions` и `reducers` для каждого отдельного компонента;
 - директория `pages` со всеми страницами, которые описаны в задании;
 - файл `README.md`;
 - файл `.gitignore`.
- Стили портированы как модули. Если есть общие стили, они портированы в глобальную область видимости.
- Код оформлен без ошибок:
 - имена переменных и функций написаны в camelCase;
 - имена классов и функциональных компонент — существительные с прописной буквы;
 - имена переменных — существительные;
 - имя функции отражает то, что она делает.

Для именования запрещены:

- транслит;
- неуместные сокращения.
- Корректно выполняются запросы к API:
 - URL-адрес домена вынесен в отдельную константу;
 - есть проверка успешности выполнения запроса;
 - цепочка обработки промисов завершается блоком `catch`.

React

- Разметка портирована в JSX:
 - разметка заключена в `()`;
 - разметка вынесена в соответствующие ей компоненты.
- Компоненты написаны корректно:
 - хуки не используются внутри условных блоков;
 - хуки вызываются в основной функции компонента;
 - при использовании классовых компонентов эффекты описаны внутри методов жизненного цикла компонента.
- Типизация реализована верно:
 - Установлены все необходимые пакеты для `TypeScript`. Конфигурация `tsconfig.json` настроена верно.
 - `any` не используются без необходимости.
 - Инлайновое отключение линтера не используется без необходимости.
 - Для описания структуры объектов используется `interface` или `type`.
 - Описания одинаковых типов не дублируются.
 - Типизированы все утилитарные функции, например, функции запросов к серверу. Все файлы с утилитарными функциями имеют расширение `.ts`.
 - Типизация компонентов реализована верно:
 - Все файлы с компонентами имеют расширение `.tsx`.
 - Для компонентов используется `TypeScript` вместо `PropTypes`.
 - Для функциональных компонентов типизируются пропсы и хуки.
 - Для классовых компонентов типизируются пропсы, параметры конструктора, состояния компонента, аргументы методов жизненного цикла.