



# Чек-лист по самопроверке №3

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-3/step-1` в `main`.
- Сборка и запуск проекта выполняются без ошибок.
- В зависимостях установлены пакеты `react-router` и `react-router-dom`.
- 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.
- Для компонентов с пропсами описан `propTypes`. Если в качестве пропсов передается объект или массив объектов, то в `propTypes` описана структура этого объекта.
- Функциональность из брифа реализована полностью:
  - Свёрстаны страницы `/register`, `/login`, `/forgot-password`, `/reset-password` и `/profile` с использованием библиотеки компонентов.
  - При нажатии на ссылки «Зарегистрироваться», «Восстановить пароль» и «Войти» происходит переход на `/register`, `/forgot-password` и `/login` соответственно.
  - Восстановление пароля и сохранение нового работает корректно.
  - В шапке панели навигации активна та ссылка на маршрут, где сейчас находится пользователь. Ссылка остаётся активной, если пользователь попадает на вложенные динамические маршруты.
  - При клике на «История заказов» происходит переход на экран `/profile/orders`, ссылка «История заказов» становится активной, экран `/profile/orders` пока отображается пустым.
  - При клике по ингредиенту происходит переход на маршрут `/ingredients/:id` и открывается модальное окно с описанием ингредиента. При прямом переходе на этот маршрут открывается страница с информацией об ингредиенте.
  - Заказы могут делать только авторизованные пользователи, все остальные действия доступны и неавторизованным.
  - На странице профиля пользователя в поля формы подставляются данные пользователя.
  - Если пользователь отредактировал данные в форме, то появляются кнопки «Отмена» и «Сохранить».
  - При нажатии кнопки «Сохранить» на странице профиля пользователя отправляется соответствующий запрос к API, который содержит изменённые данные.
  - При нажатии кнопки «Отмена» значения полей формы возвращаются в состояние до редактирования, а кнопки «Отмена» и «Сохранить» скрываются.
  - События отправки форм отслеживаются по `onSubmit` формы.
- Регистрация и авторизация реализованы корректно:
  - Роуты `/profile` и все вложенные в `/profile` маршруты доступны только авторизованным пользователям, то есть защищены НОС-компонентом `ProtectedRoute`.
  - Роуты `/login` и `/register`, `/forgot-password` и `/reset-password` недоступны авторизованным пользователям.
  - `/reset-password` доступен неавторизованным пользователям только после посещения маршрута `/forgot-password` и ввода адреса электронной почты для восстановления пароля.
  - После успешной авторизации происходит редирект на `/`.
  - При попытке неавторизованных пользователей перейти на защищённые роуты происходит редирект на `/login`. После успешной авторизации происходит переадресация на тот маршрут, к которому пользователь не получил доступ ранее.
  - После выхода из системы происходит переадресация на маршрут `/login`.
  - При запросах, использующих `accessToken`, проверяется ответ сервера. Если время жизни токена истекло, то выполняется запрос обновления токена и повторяется запрос, который не был выполнен успешно.
  - Для обновления токена и выхода из системы используется `refreshToken`.

- При получении и обновлении информации о пользователе серверу передаётся токен в поле `authorization`.
- Корректно используются компоненты `<Switch />`, `<Route />`, `<Redirect />`.