

ПРЕВЬЮ ЗАДАЧНИКА

React и Redux

React и Redux: архитектура SPA

Попробуйте формат на практике. От первого результата — к инженерному решению.



ТЕРАКОР
ТЕХНОЛОГИИ РАЗВИТИЯ
КОРПОРАТИВНЫХ РЕШЕНИЙ

Программирование с практикующим экспертом

Живые онлайн-занятия • Практика • Разбор кода

- Обучение онлайн
- Оценка знаний
- Технические собеседования

terakor.ru →

Лицензированный учебный центр

Сергей Герасименко
Основатель ТЕРАКОР

Люди.
Знания.
Развитие.

12

заданий в превью

240

в полной версии

3

уровня

Сергей Валерьевич Герасименко

Технический директор и основатель IT-компаний «ТЕРАКОР» и «ТЕРАКОР Digital», ведущий преподаватель и практикующий эксперт.

Около 5% заданий • Полные условия • Исходные номера

ВАШ ПЕРВЫЙ РЕЗУЛЬТАТ

Попробуйте. Проверьте. Продолжите.

Здесь 12 из 240 заданий полной версии (5.0%). Это выборка из разных модулей, которая показывает формат практики и рост сложности.

1

Базовый

Получите работающий результат и сравните его с критерием проверки.

2

Средний

Найдите границу решения: некорректный ввод, повтор или отказ.

3

Продвинутый

Проверьте инженерные решения и объясните выбранный компромисс.

Как работать с выборкой

Выберите доступную вам задачу. Сохраните код, входные данные и результат проверки. Номера совпадают с полной версией: по ним легко найти продолжение. Задания из поздних модулей предполагают знание соответствующих тем.

JavaScript и стек соответствующего курса: браузер/Vite, Node.js/Express/MongoDB либо React/Redux Toolkit. Зафиксируйте версии и lock-файл.

Если исходные данные не перечислены полностью, подготовьте небольшой фиксированный учебный набор. Готовые решения в задачник не включены.

ВЫБОРКА ИЗ ПОЛНОЙ ВЕРСИИ

Практические задания

01-Б01

Модуль 01 • React 18: компонентная модель и JSX

Базовый / Карточка

TaskCard получает title и status через props.

Проверка. Изменение props меняет текст. Сохраните результат запуска или проверки.

Результат: рабочий пример и краткое объяснение выбранного приёма.

02-Б03

Модуль 02 • Props, state и композиция интерфейса

Базовый / Перемещение

Перенесите задачу из todo в done в одном источнике состояния.

Проверка. Задача находится только в done. Сохраните результат запуска или проверки.

Результат: рабочий пример и краткое объяснение выбранного приёма.

03-Б03

Модуль 03 • Хуки и переиспользуемая логика

Базовый / Debounce

Custom hook задерживает поиск на 300 мс.

Проверка. Серия вводов даёт последний запрос. Сохраните результат запуска или проверки.

Результат: рабочий пример и краткое объяснение выбранного приёма.

ВЫБОРКА ИЗ ПОЛНОЙ ВЕРСИИ

Практические задания

03-C05

Модуль 03 • Хуки и переиспользуемая логика

Средний / Мемо

Профилируйте фильтрацию до и после `useMemo`. Дополнительное условие: Оптимизация не меняет результат и не скрывает неверные зависимости.

Проверка. Сохранены условия замера. Проверьте обычный случай и описанную границу/ошибку отдельно.

Результат: доработанное решение и минимум три проверки, включая указанный крайний случай.

04-C04

Модуль 04 • React Router и многостраничные SPA

Средний / Фильтр

Храните статус задачи в `search params`. Дополнительное условие: Неизвестный статус нормализуется по контракту.

Проверка. URL можно скопировать и восстановить фильтр. Проверьте обычный случай и описанную границу/ошибку отдельно.

Результат: доработанное решение и минимум три проверки, включая указанный крайний случай.

05-C02

Модуль 05 • Redux Toolkit: store, slice и предсказуемое состояние

Средний / Selector

Выберите задачи `status=done` через `selector`. Дополнительное условие: `Selector` не меняет `store`.

Проверка. Возвращаются только `done`. Проверьте обычный случай и описанную границу/ошибку отдельно.

Результат: доработанное решение и минимум три проверки, включая указанный крайний случай.

ВЫБОРКА ИЗ ПОЛНОЙ ВЕРСИИ

Практические задания

06-C03

Модуль 06 • Асинхронные данные: createAsyncThunk и RTK Query

Средний / Optimistic

Переместите задачу оптимистично. Дополнительное условие: При 500 возвращается предыдущее состояние и сообщение.

Проверка. UI меняется до ответа. Проверьте обычный случай и описанную границу/ошибку отдельно.

Результат: доработанное решение и минимум три проверки, включая указанный крайний случай.

04-П03

Модуль 04 • React Router и многостраничные SPA

Продвинутый / 404

Добавьте страницу неизвестного маршрута. Обработайте усложнение: Ссылка возвращает к существующей странице.

Проверка. Маршрутизация клиента не является серверной авторизацией; проверьте прямой URL, Back и неизвестные параметры.

Результат: решение, повторяемая проверка исходного результата («/unknown показывает 404 UI») и усложнения; пояснение одного технического компромисса.

05-П03

Модуль 05 • Redux Toolkit: store, slice и предсказуемое состояние

Продвинутый / Перенос

Одно действие меняет колонку задачи. Обработайте усложнение: Неизвестный ID не портит список.

Проверка. Состояние должно быть сериализуемым; действия описывают события, а селекторы вычисляют производные данные без мутаций.

Результат: решение, повторяемая проверка исходного результата («Нет промежуточного дублирования») и усложнения; пояснение одного технического компромисса.

ВЫБОРКА ИЗ ПОЛНОЙ ВЕРСИИ

Практические задания

06-П05

Модуль 06 • Асинхронные данные: createAsyncThunk и RTK Query

Продвинутый / Повтор

Повторите GET после ошибки по кнопке. Обработайте усложнение: Число автоматических повторов ограничено.

Проверка. Управляйте cache invalidation и гонками; optimistic update обязан откатываться при серверном отказе.

Результат: решение, повторяемая проверка исходного результата («Успех убирает состояние ошибки») и усложнения; пояснение одного технического компромисса.

07-П05

Модуль 07 • Тестирование, мемо и production-паттерны

Продвинутый / Регрессия

Внесите дефект: замените строгое сравнение конца периода на включительное. Обработайте усложнение: После исправления обе границы периода проходят.

Проверка. Разделите unit и integration проверки; тест должен падать при намеренном дефекте, а после исправления проходить.

Результат: решение, повторяемая проверка исходного результата («Тест на дате end падает») и усложнения; пояснение одного технического компромисса.

08-П04

Модуль 08 • Итоговый проект: TaskBoard Pro

Продвинутый / Контроль

Добавьте один успешный и один ошибочный сквозной сценарий. Обработайте усложнение: Повтор сценария не оставляет лишних записей или файлов.

Проверка. Соберите воспроизводимый сценарий из модулей курса; покажите успешный запуск, отказ и восстановление без ручной правки исходников.

Результат: решение, повторяемая проверка исходного результата («У каждого заранее определён ожидаемый результат») и усложнения; пояснение одного технического компромисса.

ПРОДОЛЖИТЕ ПРАКТИКУ

От выборки — к полному маршруту

В полной версии — 240 заданий в 8 тематических модулях. В каждом модуле: 10 заданий базового, 10 среднего и 10 продвинутого уровня.

Что можно разобрать дальше

Хуки и переиспользуемая логика

Следующий сценарий: «Effect». Условия и критерии проверки — в полной версии.

Асинхронные данные: `createAsyncThunk` и `RTK Query`

Следующий сценарий: «Query». Условия и критерии проверки — в полной версии.

Итоговый проект: **TaskBoard Pro**

Следующий сценарий: «Запуск». Условия и критерии проверки — в полной версии.

Превью показывает формат. Полная версия даёт последовательную практику по всем модулям, дополнительные сценарии и условия усложнения.

Запросить полную версию задачника

info@terakor.ru • Укажите название купса: React и Redux

Страница купса: <https://terakor.ru/catalog/javascript/react-redux/>