

# Аспекты разработки современного WEB-приложения

<sup>1</sup>\*САТИНОВ Акжол Кудайбергенович, студент, [akjolsatinov01@gmail.com](mailto:akjolsatinov01@gmail.com),

<sup>1</sup>СМАГУЛОВА Асемгуль Сериковна, к.т.н., доцент, [assemgul\\_work@mail.ru](mailto:assemgul_work@mail.ru),

<sup>2</sup>БЕЛОЗУБОВ Александр Владимирович, к.т.н., преподаватель, зам. декана, [belozubov@itmo.ru](mailto:belozubov@itmo.ru),

<sup>1</sup>Карагандинский технический университет, Казахстан, 100027, Караганда, пр. Н. Назарбаева, 56,

<sup>2</sup>Университет ИТМО, Россия, 197101, Санкт-Петербург, Кронверкский пр., 49,

\*автор-корреспондент.

**Аннотация.** В статье рассматриваются аспекты разработки WEB-приложения. Приводится описание принципов работы Single Page Application (SPA). В качестве примера была разработана корзина интернет-магазина. Были применены современные подходы при разработке WEB-приложения, такие как ReactJS и Redux. Продемонстрирована функциональная парадигма, которая используется при разработке WEB-приложения. В статье рассматривается способ разработки современного Single Page Application с помощью React версии 17. Рассмотрен принцип работы Redux. Также был показан алгоритм приложения.

**Ключевые слова:** современное WEB-приложение, WEB-разработка, одностраничное приложение, алгоритмы, функциональное программирование, разработка интернет-магазина, JavaScript, библиотека ReactJS, программирование, JavaScript паттерны.

## Введение

Когда только начали разрабатывать WEB-страницы, они представляли собой только текст, изображения или даже видео, т.е. были статичными. Но с развитием технологий, например, как Ajax, появилась возможность разрабатывать динамические WEB-страницы, т.е. WEB-приложения.

Разработка WEB-приложений не зависит от платформы, поэтому не нужно разрабатывать версии для Windows, IOS, Android и т.д., и это значительно уменьшает расходы на разработку и поддержку. На сегодняшний момент благодаря возможностям современных браузеров, JavaScript и сообществам WEB-разработчиков, WEB-приложения по функциональности не отличаются от мобильных или десктопных приложений.

Даже известным компаниям, у которых нативные приложения являются очень популярными, необходимо WEB-приложение. В статье приводится описание разновидности WEB-приложения, а именно SPA (Single Page Application).

SPA или одностраничное приложение – это WEB-приложение, которое загружается на одну HTML-страницу.

В отличие от простой HTML-страницы, при переключении на другие страницы необходимые элементы просто изменяются вместо полной перезагрузки. YouTube, Facebook, Twitter, Google Docs и многие другие популярные сервисы используют SPA.

Для разработки SPA использованы ReactJS,

Redux и Material-UI.

ReactJS [1] – это библиотека JavaScript [2], которая сочетает в себе скорость JavaScript и использует новый способ рендеринга веб-страниц [3], что делает их очень динамичными и быстро реагирующими на ввод пользователя.

Разработка ReactJS основана на компонентном подходе [4]. Написание компонентов похоже на создание строительных блоков, которые будут использованы в системе. Из рисунка 1 видно, что приложение состоит из пяти различных компонентов. Курсивом выделены данные, которые эти компоненты представляют.

- FilterableProductTable (оранжевый): контейнер, содержащий пример целиком;
- SearchBar (синий): поле пользовательского ввода;
- ProductTable (зелёный): отображает и фильтрует список данных, основанный на пользовательском вводе;
- ProductCategoryRow (голубой): наименования категорий;
- ProductRow (красный): отдельно взятый товар.

Redux [5] – это менеджер состояний, предназначенный для помощи при разработке на JavaScript. На рисунке 2 показана схема принципа работы Redux. Redux можно использовать в любом фреймворке или библиотеке, в данном случае использован с ReactJS. Для связи с ReactJS, будет использоваться модуль react-redux.

☐ Only show products in stock

Name	Price
<b>Sporting Goods</b>	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
<b>Electronics</b>	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

Рисунок 1 – Компонентный подход ReactJS

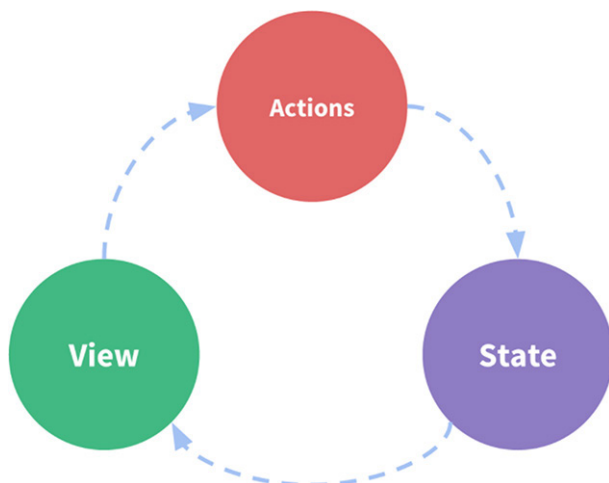


Рисунок 2 – Принцип работы Redux

Если говорить о принципе работы Redux, то он работает на основе шаблона проектирования Наблюдатель (Observer) [6]. Redux имеет локальное состояние, состояние доступно только для чтения, изменить ничего напрямую нельзя. Чтобы изменить состояние, Redux предоставляет функцию dispatch, при вызове этой функции необходимо передать action (действие).

При запуске действия обязательно что-то произойдет и состояние приложения изменится. Это работа редукторов.

Редуктор [7] – это чистая функция, которая вычисляет следующее состояние дерева на основе его предыдущего состояния и примененного действия.

Также имеются две важные функции, которые предоставляет Redux, такие как:


- getState() – позволяет узнать текущее состояние;
- subscribe – через эти функции мы подписываемся на изменения состояния.

React Material-UI – это набор компонентов React, которые реализуют дизайн Material Design от Google.

В качестве наглядного примера создана корзина интернет-магазина. На рисунке 3 и 4 можно посмотреть список товаров и корзину. Приложение разделено на два больших компонента, это список товаров и сама корзина.

При клике на ДОБАВИТЬ В КОРЗИНУ, один товар добавляется в корзину, как показано на рисунке 4. В корзине показана информация о товаре, количество и общая стоимость, а также есть три кнопки для увеличения, уменьшения и удаления товара.


Это приложение было разработано в соответствии с функциональной парадигмой [8]. Изначально корзина имеет следующую структуру (рисунок 5), далее она будет называться состояни-



Ноутбук

120000 тг


ДОБАВИТЬ В КОРЗИНУ



Смартфон

80000 тг

ДОБАВИТЬ В КОРЗИНУ



Клавиатура

10000 тг

ДОБАВИТЬ В КОРЗИНУ

Рисунок 3 – Список товаров

ID	Наименование товара	Общая Стоимость	Кол-во	Действия
1	Ноутбук	840000 тг	7	+1 -1 
2	Смартфон	240000 тг	3	+1 -1 
3	Клавиатура	50000 тг	5	+1 -1 
Итого			1130000 тг	

Рисунок 4 – Корзина

```
const initialState: ShoppingCart = {
  shoppingCartItems: [],
  orderTotal: 0,
}
```

Рисунок 5 – Структура корзины

ем корзины. В поле `shoppingCartItems` хранятся все товары, добавленные в корзину, а поле `orderTotal` хранит общую стоимость корзины.

Для добавления, уменьшения и удаления товара вызываются соответственно функций `addToCart`, `removeFromCart` и `removeAllFromCart` (рисунок 6). Также используются и вспомогательные функции как `updateOrder` для обновления заказа (рисунок 7), `updateCartItems` для обновлений товаров (рисунок 8) и `updateCartItem` для обновления товара (рисунок 9).

При добавлении товара в корзину вызывается функция `addToCart`, которая принимает товар, далее вызывает функция `updateOrder`, где передается товар, и если товар уже имеется в корзине, то количество товара увеличивается, если нет, то товар добавляется в корзину.

При уменьшении товара в корзине, вызывается функция `removeFromCart`, которая передает при вызове `updateOrder` последним аргументом `-1` [9], тем самым `updateOrder` уменьшает количество товара в корзине, но если количество товара равно нулю, то товар удаляется из корзины.

Для удаления товара из корзины вызывается

функция [10] `removeAllFromCart`, которая вызывая функцию `updateOrder`, передает отрицательное число с количеством товаров в корзине.

### Заключение

Основной целью статьи является изучение аспектов разработки современного WEB-приложения и особенностей приложений на основе ReactJS.

Исходя из собранных результатов, в статье представлен ряд преимуществ, которые привели к актуальности разработки проекта на ReactJS, отражающего возможности библиотеки. Говоря о качестве приложений, ReactJS помогает создавать интерактивные пользовательские интерфейсы, которые обеспечивают интерфейс между Пользователем и приложением.

Связка React-Redux является самой популярной при разработке Single Page Application. Когда данные разделены между несколькими компонентами программы, самое время выбрать React-Redux. React-Redux помогает логично и надежно управлять общими данными. Чтобы обновить данные в React-Redux, разработчики должны публиковать каждое действие. Затем они решают, как обновить данные в Reducer. React-Redux также снижает адгезию компонентов, поэтому компоненты могут быть многоразовыми и динамическими.

Таким образом, в данной статье приведено описание технологий разработки Web-приложений, в частности, ReactJS и Redux. Также была описана одна из разновидностей Web-приложений, то есть SPA.

```

addToCart(state, action: PayloadAction<Product>) {
  return updateOrder(state, action.payload, 1)
},
removeFromCart(state, action: PayloadAction<Product>) {
  return updateOrder(state, action.payload, -1)
},
removeAllProductFromShoppingCart(
  state,
  action: PayloadAction<Product>,
) {
  const { shoppingCartItems } = state
  const { id } = action.payload
  const productCount = shoppingCartItems.find(
    (shoppingCartItem) => shoppingCartItem.product.id === id,
  )?.count
  return updateOrder(state, action.payload, -(productCount || 1))
}

```

Рисунок 6 – Код обновления заказа в корзине

```

const updateOrder = (
  state: ShoppingCart,
  product: Product,
  quantity: number,
): ShoppingCart => {
  const { shoppingCartItems } = state

  const shoppingCartItemIndex = shoppingCartItems.findIndex(
    ({ product: { id } }) => id === product.id,
  )
  const shoppingCartItem = shoppingCartItems[shoppingCartItemIndex]

  const newShoppingCartItem: ShoppingCartItem = updateCartItem(
    product,
    shoppingCartItem,
    quantity,
  )

  const updatedShoppingCartItems = updateCartItems(
    shoppingCartItems,
    newShoppingCartItem,
    shoppingCartItemIndex,
  )

  const orderTotal = updatedShoppingCartItems.reduce((acc, item) => {
    return acc + item.total
  }, 0)
  console.log(shoppingCartItems, orderTotal, 'total')

  return {
    orderTotal,
    shoppingCartItems: updatedShoppingCartItems,
  }
}

```

Рисунок 7 – Код обновления заказа в корзине



```

const updateCartItem = (
  product: Product,
  shoppingCartItem: ShoppingCartItem,
  quantity: number,
) => {
  const updatedShoppingCartItem: ShoppingCartItem = shoppingCartItem
    ? {
      ...shoppingCartItem,
      count: shoppingCartItem.count + quantity,
      total: shoppingCartItem.total + quantity * product.price,
    }
    : {
      count: quantity,
      product,
      total: quantity * product.price,
    }
  return updatedShoppingCartItem
}

```

Рисунок 8 – Код обновления товара в корзине

```

const updateCartItems = (
  shoppingCartItems: ShoppingCartItem[],
  shoppingCartItem: ShoppingCartItem,
  idx: number,
) => {
  if (shoppingCartItem.count === 0) {
    return [
      ...shoppingCartItems.slice(0, idx),
      ...shoppingCartItems.slice(idx + 1),
    ]
  }

  if (idx === -1) {
    return [...shoppingCartItems, shoppingCartItem]
  }

  return [
    ...shoppingCartItems.slice(0, idx),
    shoppingCartItem,
    ...shoppingCartItems.slice(idx + 1),
  ]
}

```

Рисунок 9 – Алгоритм обновления товаров в корзине

## СПИСОК ЛИТЕРАТУРЫ

1. Мардан А. Реакт быстро. – СПб: Питер, 2019. 560 с.
2. Физерс Майкл К. Эффективная работа с унаследованным кодом. – СПб: Питер, 2016. 400 с.
3. Дуглас Крокфорд. JavaScript: сильные стороны. – СПб: Питер, 2018. 176 с.
4. Фримен Эрик, Робсон Элизабет. Изучаем программирование на JavaScript. – СПб: Питер, 2018. 640 с.
5. Крокфорд Дуглас. Как устроен JavaScript. – СПб: Питер, 2019. 304 с.

6. Стефанов. JavaScript. Шаблоны. – СПб: Питер, 2017. 272 с.
7. Бибо Беэр. Секреты JavaScript ниндзя. – СПб: Питер, 2018. 544 с.
8. Томас Х. Кормен, Клиффорд Штайн, Рональд Л. Ривест, Чарльз И. Лейзерсон. Алгоритмы. Построение и анализ. – СПб: Питер, 2019. 997 с.
9. Мартин Фаулер – Рефакторинг. Улучшение существующего кода. – СПб: Питер, 2018. 432 с.
10. Әділхан А.Б., Смагулова А.С. Android қосымшаларын әзірлеуде Java мен Kotlin бағдарламалау // «Университет еңбектері» республикалық журналы. 2021. 3 (84). Б. 253-262.

### Қазіргі WEB-қосымшаны әзірлеу аспектілері

<sup>1\*</sup>САТИНОВ Акжол Кудайбергенович, студент, akjolsatinov01@gmail.com,

<sup>1</sup>СМАГУЛОВА Асемгуль Сериковна, т.ғ.к., доцент, assemgul\_work@mail.ru,

<sup>2</sup>БЕЛОЗУБОВ Александр Владимирович, т.ғ.к., оқытушы, декан орынбасары, belozubov@itmo.ru ,

<sup>1</sup>Қарағанды техникалық университеті, Қазақстан, 100027, Қарағанды, Н. Назарбаев даңғылы, 56,

<sup>2</sup>ИТМО университеті, Ресей, 197101, Санкт-Петербург, Кронверкский даңғылы, 49,

\*автор-корреспондент.

**Аңдатпа.** Мақалада WEB-қосымшаны әзірлеудің аспектілері қарастырылған. Single Page Application-ның жұмыс істеу принциптерінің сипаттамасы келтірілген. Мысал ретінде интернет-дүкенге арналған корзина жасалды. Мұнда ReactJS және Redux сияқты WEB-қосымшасын әзірлеу кезінде заманауи тәсілдер қолданылды. WEB-қосымшасын жасауда қолданылатын функционалды парадигма көрсетілді. Мақалада React 17 нұсқасын қолдана отырып, заманауи бірыңғай бет қосымшасын жасау әдісі қарастырылған. Redux-тің жұмыс істеу принципі қарастырылған. Қосымшаның алгоритмі де көрсетілді.

**Кілт сөздер:** қазіргі WEB-қосымша, WEB-бағдарлама, бір беттік қосымша, алгоритмдер, функционалды бағдарламалау, интернет-дүкенді әзірлеу, JavaScript, ReactJS кітапханасы, бағдарламалау, JavaScript үлгілері.

### Aspects of Developing a Modern WEB-application

<sup>1\*</sup>SATINOV Akzhol, student, akjolsatinov01@gmail.com,

<sup>1</sup>SMAGULOVA Asemgul, Cand. Tech. Sci., Associate Professor, assemgul\_work@mail.ru,

<sup>2</sup>BELOZUBOV Alexander, Cand. Tech. Sci., Lecturer, Deputy Dean, belozubov@itmo.ru ,

<sup>1</sup>Karaganda Technical University, Kazakhstan, 100027, Karaganda, N. Nazarbayev Avenue, 56,

<sup>2</sup>ITMO University, Russia, 197101, St. Petersburg, Kronverksky Avenue, 49,

\*corresponding author.

**Abstract.** The article discusses aspects of the development of a WEB-application. The description of the principles of operation of Single Page Application is given. As an example, a shopping cart was developed for an online store. Modern approaches were applied when developing a WEB application, such as ReactJS and Redux. The functional paradigm that is used in the development of a WEB-application has been demonstrated. This article discusses a way to develop a modern Single Page Application using React version 17. The principle of operation of Redux is considered. The algorithm of the application was also shown.

**Keywords:** modern WEB-application, WEB-development, Single Page Application, algorithms, functional programming, online store development, JavaScript, ReactJS library, programming, JavaScript patterns.

### REFERENCES

1. Mardan A. React bystro. – Saint Petersburg: Piter, 2019. 560 p.
2. Fizers Maikl K. Effektivnaya rabota s unasledovannym kodom. – Saint Petersburg: Piter, 2016. 400 p.
3. Duglas Krokford. JavaScript: silnye storony. – Saint Petersburg: Piter, 2018. 176 p.
4. Frimen Erik, Robson Elizabet. İzuchaem programmirovaniye na JavaScript. – Saint Petersburg: Piter, 2018. 640 p.
5. Krokford Duglas. Kak ustroen JavaScript. – Saint Petersburg: Piter, 2019. 304 p.
6. Stefanov. JavaScript. Şablony. – Saint Petersburg: Piter, 2017. 272 p.
7. Bibo Beer. Sekrety JavaScript nindzya. – Saint Petersburg: Piter, 2018. 544 p.
8. Tomas H. Kormen, Klifford Ştain, Ronald L. Rivest, Charlz İ. Leizerson. Algoritmy. Postroenie i analiz. – Saint Petersburg: Piter, 2019. 997 p.
9. Martin Fauler – Refaktoring. Uluchşenie suestvuego koda. – Saint Petersburg: Piter, 2018. 432 p.
10. Ädilhan A.B., Smagulova A.S. Android qosymşalaryn äzirleude Java men Kotlin bağdarlamalau // «Universitet eñbekteri» respublikalyq jurnaly. 2021. 3 (84). pp. 253-262.