

## Создание корзины товаров для интернет-магазина

*Круглик Роман Игоревич*

*Приамурский государственный университет им. Шолом-Алейхема*

*Студент*

### Аннотация

В статье поэтапно рассматривается создание корзины для добавления товаров. Реализован динамический модуль, который может быть внедрён в любой интернет-магазин. Так же разработаны дополнительные функции для редактирования уже собранной корзины товаров.

**Ключевые слова:** реляционная БД, нереляционная БД, сравнительный анализ.

## Creating a shopping cart for an online store

*Kruglik Roman Igorevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

In article gradually considers the creation of a basket for adding goods. A dynamic module has been implemented that can be implemented in any online store. Also developed additional functions for editing an already assembled basket of goods.

**Keywords:** relational database, non-relational database, comparative analysis.

На сегодняшний день каждый второй человек хотя бы раз в жизни совершал покупку через интернет и с каждым годом таких покупателей всё больше. Вместе с этим растёт популярность создания интернет-магазинов так как появляются всё новые потенциальные покупатели.

Интернет-магазин удобен не только пользователю, но и самому создателю, так как существенно уменьшаются издержки производителя, сэкономя на содержании обычного магазина, расширяют рынки сбыта, так же, как и расширяет возможность покупателя – покупать любой товар в любое время в любой стране, в любом городе, в любое время суток, в любое время года. Это даёт неоспариваемое преимущество перед обычными магазинами. Этот момент является существенным при переходе производителей с «обычной» торговли на «электронную».

Исследования в области создания интернет-магазина актуальны и по сей день. В статье Заяц Т.А. [1] рассматривается использование web-технологий в создании интернет-магазинов. Антонов И. [2] разбирает CMS для создания интернет-магазинов. В статье Ермолик А.В., Багаева А.П. [3] рассматривается описание этапов создания интернет магазина, а также его

преимущества, недостатки и основная классификация. В работе Гафаров Н.Ф., Хусаинова Г.Я. [4] рассматривается разработка и создание интернет-магазина для мебельного предприятия «Уют».

Создание собственного интернет-магазина - это достаточно сложный и трудоёмкий процесс. В данной статье будет рассматриваться создание одного из самых важных модулей для системы – корзина товаров.

Систему будет состоять из нескольких частей:

- 1) Страница с товарами.
- 2) Корзина, в которой будут отображаться выбранные товары.

Начнём с главной страницы товаров, которая будет называться index.html (см. рис. 1).

```
<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Интернет-магазин.</title>
  <meta name="description" content="Главная страница с каталогом" />
  <link href='https://fonts.googleapis.com/css?family=Ubuntu:400,700&subset=latin,cyrillic-ext' rel='stylesheet' type='text/css'>
  <link href="css/bootstrap.min.css" rel="stylesheet" type="text/css">
  <link href="css/main.css" rel="stylesheet" type="text/css">
</head>
<body data-page="catalog">
  <div class="container">
    <div class="container">
      <h1 align="center">Интернет-магазин.</h1>
      <br />
      <ul class="nav nav-pills">
        <li class="active"><a href="/">Каталог</a></li>
        <li><a href="cart.html">Корзина<span id="total-cart-count" class="badge"></span></a></li>
        <li><a href="order.html">Оформление заказа</a></li>
      </ul>
      <br />
      <ul id="goods" class="list-unstyled">
        
      </ul>
    </div>
    <script id="catalog-template" type="text/template">
      <%_.each(goods, function(good) { %>
        <li class="good-item row">
          <div class="col-md-4">
            
          </div>
          <div class="col-md-8">
            <div class="good-item_name"><%= good.name %></div>
            <div class="good-item_price"><%= good.price %> руб.</div>
            <button
              class="good-item_btn-add btn btn-info btn-sm js-add-to-cart"
              data-id="<%= good.id %>"
              data-name="<%= good.name %>"
              data-price="<%= good.price %>"
            >Добавить в корзину</button>
          </div>
        </li>
      <% }); %>
    </script>
    <script src="js/vendor/jquery.min.js" type="text/javascript"></script>
    <script src="js/vendor/jquery.cookie.js" type="text/javascript"></script>
    <script src="js/vendor/underscore.min.js" type="text/javascript"></script>
    <script src="js/modules/catalog.js" type="text/javascript"></script>
    <script src="js/modules/cart.js" type="text/javascript"></script>
    <script src="js/modules/main.js" type="text/javascript"></script>
  </body>
</html>
```

Рисунок 1. Файл index.html

В данном файле выводятся все товары. Так же подключим стили через файл `css/main.css` (см. рис. 2).

```
body {  
  font-family: Ubuntu;  
}  
  
.good-item {  
  margin: 10px 0;  
  padding: 20px;  
  border: solid 1px gray;  
  border-radius: 3px;  
}  
  
.good-item__id {  
  padding-top: 30px;  
  color: steelblue;  
}  
  
.good-item__name {  
  padding-top: 10px;  
  font-size: 1.1em;  
}  
  
.good-item__price {  
  padding-top: 10px;  
  color: red;  
}  
  
.good-item__btn-add {  
  margin-top: 30px;  
}
```

Рисунок 2. Файл `main.css`

Теперь необходимо создать файл в формате JSON с товарами, которые будут выводиться в системе. Для этого создадим файл `goods.json` и заполним его (см. рис. 3).

```
[
  {
    "id": "1",
    "name": "Телевизор SUPRA",
    "price": "19000",
    "img": "img/cars/tv_supra.jpg"
  },
  {
    "id": "2",
    "name": "Фотокамера Nikon",
    "price": "25000",
    "img": "img/cars/camera_nikon.jpg"
  },
  {
    "id": "3",
    "name": "Apple ipad",
    "price": "35000",
    "img": "img/cars/ipad.jpg"
  },
  {
    "id": "4",
    "name": "Samsung Galaxy",
    "price": "20000",
    "img": "img/cars/phone_galaxy.jpg"
  }
]
```

Рисунок 3. Файл goods.json

Каждый товар имеет 4 атрибута:

- 1) Id (номер)
- 2) Name (название)
- 3) Price (цена)
- 4) Img (путь к изображению)

Все эти данные хранятся в JSON формате и выводятся на главную страницу (см. рис. 4).

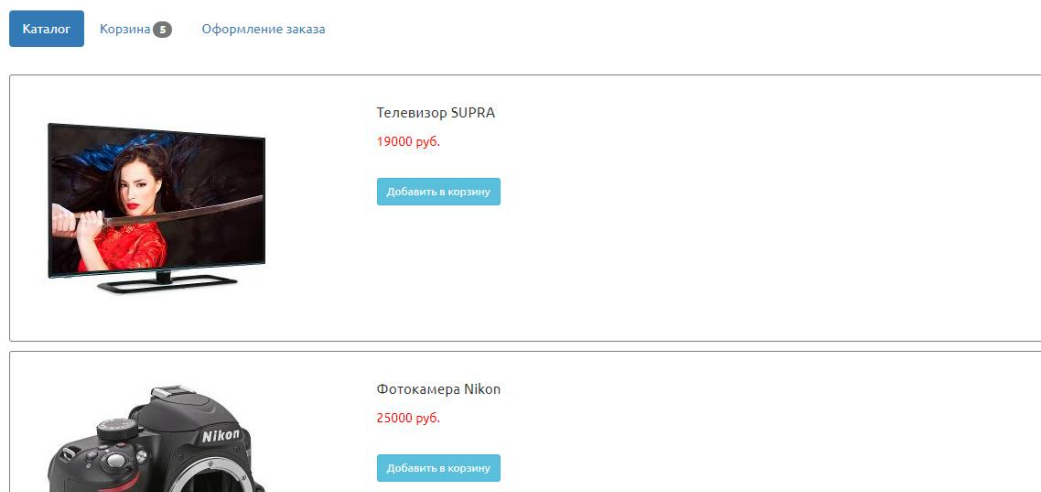


Рисунок 4. Главная страница

Все товары выводятся и теперь можно перейти к функционалу. При нажатии на кнопку «Добавить в корзину», товар должен добавляться в корзину (см. рис. 5).

```
'use strict';
// Модуль каталога
var catalog = (function($) {
  // Инициализация модуля
  function init() {
    _render();
  }
  // Рендерим каталог
  function _render() {
    var template = _.template($('#catalog-template').html()),
        $goods = $('#goods');
    $.getJSON( a: 'data/goods.json', b: function(data) {
      $goods.html(template({goods: data}));
    });
  }
  // Экспортируем наружу
  return {
    init: init
  }
})(jQuery);
```

Рисунок 5. Обработка страницы с каталогами

Далее перейдём к функциям самой корзины. Данный функционал разделён на 3 части:

- 1) Работа с данными;
- 2) Работа с событиями;
- 3) Дополнительные настройки.

Перейдём к самой функции (см. рис. 6-7).

```
'use strict';
// Модуль корзины
var cart = (function($) {
  var cartData,
      opts = {};
  // Инициализация модуля
  function init(options) {
    _initOptions(options);
    updateData();
    if (opts.renderCartOnInit) {
      renderCart();
    }
    if (opts.renderMenuCartOnInit) {
      renderMenuCart();
    }
    _bindHandlers();
  }
  // Получаем данные
  function updateData() {
    cartData = JSON.parse(localStorage.getItem( key: 'cart')) || [];
    return cartData;
  }
  // Возвращаем данные
  function getData() {
    return cartData;
  }
  // Сохраняем данные в localStorage
  function saveData() {
    localStorage.setItem('cart', JSON.stringify(cartData));
    return cartData;
  }
  // Очищаем данные
  function clearData() {
    cartData = [];
    saveData();
    return cartData;
  }
  // Поиск объекта в коллекции cartData по id
  function getById(id) {
    return _.findWhere(cartData, { id: id });
  }
  // Добавление товара в коллекцию
  function add(item) {
    var oldItem;
    updateData();
    oldItem = getById(item.id);
    if (!oldItem) {
      cartData.push(item);
    } else {
      oldItem.count = oldItem.count + item.count;
    }
    saveData();
    return item;
  }
}
```

Рисунок 6. Модуль корзины

```
// Удаление товара из коллекции
function remove(id) {
  updateData();
  cartData = _.reject(cartData, t function(item) {
    return item.id === id;
  });
  saveData();
  return cartData;}

// Изменение количества товара в коллекции
function changeCount(id, delta) {
  var item;
  updateData();
  item = getById(id);
  if(item) {
    item.count = item.count + delta;
    if (item.count < 1) {
      remove(id);}
    saveData();}
  return _.findWhere(cartData, t {id: id}) || {};}

// Возвращаем количество товаров (количество видов товаров в корзине)
function getCount() {
  return _.size(cartData);}

// Возвращаем общее количество товаров
function getCountAll() {
  return _.reduce(cartData, function(sum, item) {return sum + item.count}, 0);}

// Возвращаем общую сумму
function getSumma() {
  return _.reduce(cartData, function(sum, item) {return sum + item.count * item.price}, 0);}
```

Рисунок 7. Модуль корзины

Реализовано несколько операций:

- 1) Получить данные из корзины.
- 2) Сохранить данные в localStorage (Чтобы при обновлении страницы собранная корзина не куда не исчезла).
- 3) Очистить данные.
- 4) Добавление и удаление товара.
- 5) Внутренний поиск по id.

Корзина должна наглядно отображаться на отдельной странице, для этого создаётся файл cart.html (см. рис. 8).

```

<script id="cart-template" type="text/template">
  <% _.each(goods, function(good) { %>
    <tr class="cart-item js-cart-item" data-id="<%= good.id %>">
      <td><%= good.name %></td>
      <td><%= good.price %> руб.</td>
      <td><span
        class="cart-item_btn-dec-count js-change-count"
        title="Уменьшить на 1"
        data-id="<%= good.id %>"
        data-delta="-1"
      >
        <span class="glyphicon glyphicon-minus"></span>
      </span>
      <span class="js-count"><%= good.count %></span>
      <span
        class="cart-item_btn-inc-count js-change-count"
        title="Увеличить на 1"
        data-id="<%= good.id %>"
        data-delta="1"
      >
        <span class="glyphicon glyphicon-plus"></span>
      </span>
    </td>
    <td><span class="js-summa"><%= good.count * good.price %></span> руб.</td>
    <td>
      <span class="cart-item_btn-remove js-remove-from-cart" title="Удалить из корзины" data-id="<%= good.id %>">
        <span class="glyphicon glyphicon-remove"></span></span></td></tr>
  <% }); %>
</script>

```

Рисунок 8. Файл cart.html

Далее визуализируется корзина с товарами (см. рис. 9).

```

// Рендерим корзину
function renderCart() {
  var template = _.template($('#cart-template').html()),
      data = {
        goods: cartData
      };
  $(opts.elCart).html(template(data));
  renderTotalCartSumma();
}

// Рендерим количество товаров в меню
function renderMenuCart() {
  var countAll = getCountAll();
  $(opts.elTotalCartCount).html(countAll !== 0 ? countAll : '');
}

// Рендерим общую сумму товаров
function renderTotalCartSumma() {
  $(opts.elTotalCartSumma).html(getSumma());
}

```

Рисунок 9. Рендеринг товаров и корзины

После дополним наш модуль некоторыми функциями для более удобного пользования корзиной (см. рис. 10)



```
// Поиск продукта в корзине по id
function findCartElemById(id) {
    return $(opts.elCartItem + '[' + opts.attrId + '=' + id + '']');
}

// Добавление в корзину
function _onClickAddBtn() {
    $('body').on( a: 'click', opts.elAddToCart, c: function(e) {
        var $this = $(this);
        add( item: {
            id: +$this.attr(opts.attrId),
            name: $this.attr(opts.attrName),
            price: +$this.attr(opts.attrPrice),
            count: 1});
        renderMenuCart();
        alert('Товар добавлен в корзину');
    });
}

// Меняем количество товаров в корзине
function _onClickChangeCountInCart() {
    $('body').on( a: 'click', opts.elChangeCount, c: function(e) {
        var $this = $(this),
            id = +$this.attr(opts.attrId),
            delta = +$this.attr(opts.attrDelta),
            $cartElem = findCartElemById(id),
            cartItem = changeCount(id, delta);
        if (cartItem.count) {
            $cartElem.find(opts.elCartCount).html(cartItem.count);
            $cartElem.find(opts.elCartSumma).html(cartItem.count * cartItem.price);
        } else {
            $cartElem.remove();
        }
        renderMenuCart();
        renderTotalCartSumma();});
}

// Удаляем товар из корзины
function _onClickRemoveFromCart() {
    $('body').on( a: 'click', opts.elRemoveFromCart, c: function(e) {
        if(!confirm('Удалить товар из корзины?')) return false;
        var $this = $(this),
            id = +$this.attr(opts.attrId),
            $cartElem = findCartElemById(id);
        remove(id);
        $cartElem.remove();
        renderMenuCart();
        renderTotalCartSumma();});
}
```

Рисунок 10. Дополнительные функции

Теперь необходимо навесить обработчики событий и обозначить дополнительные настройки (см. рис. 11)

```
// Навешиваем события
function _bindHandlers() {
  _onClickAddBtn();
  _onClickChangeCountInCart();
  _onClickRemoveFromCart();
}

// Инициализируем настройки
function _initOptions(options) {
  var defaultOptions = {
    renderCartOnInit: true,
    renderMenuCartOnInit: true,
    elAddToCart: '.js-add-to-cart',
    attrId: 'data-id',
    attrName: 'data-name',
    attrPrice: 'data-price',
    attrDelta: 'data-delta',
    elCart: '#cart',
    elTotalCartCount: '#total-cart-count',
    elTotalCartSumma: '#total-cart-summa',
    elCartItem: '.js-cart-item',
    elCartCount: '.js-count',
    elCartSumma: '.js-summa',
    elChangeCount: '.js-change-count',
    elRemoveFromCart: '.js-remove-from-cart'
  };
  _.defaults(options || {}, defaultOptions);
  opts = _.clone(options);
}
```

Рисунок 11. Дополнительные функции

Посмотрим на результат выполнения модуля. (см. рис. 12)

### Интернет-магазин.

Каталог [Корзина 6](#)

Название	Цена	Количество	Сумма	
Телевизор SUPRA	19000 руб.	- 1 +	19000 руб.	✘
Фотокамера Nikon	25000 руб.	- 2 +	50000 руб.	✘
Apple Ipad	35000 руб.	- 2 +	70000 руб.	✘
Samsung Galaxy	20000 руб.	- 1 +	20000 руб.	✘

Итого: 159000 руб.

[Оформить заказ](#)

Рисунок 11. Результат работы модуля

Модуль, который был рассмотрен в данной статье, сделан как динамическая составляющая любой системы. Данную работу можно интегрировать в любой проект без каких-либо ограничений.

### **Библиографический список**

1. Заяц Т.А. Использование web-технологий в создании интернет-магазинов // Потребительская кооперация. 2011. № 3 (34). С. 55-62.
2. Антонов И. Обзор CMS для создания интернет-магазинов // Системный администратор. 2014. № 5 (138). С. 62-69.
3. Ермолик А.В., Багаева А.П. Создание интернет-магазина // Актуальные проблемы авиации и космонавтики. 2013. Т. 1. № 9. С. 425-426.
4. Гафаров Н.Ф., Хусаинова Г.Я. Разработка и создание интернет-магазина "УЮТ" // Актуальные направления научных исследований XXI века: теория и практика. 2015. Т. 3. № 7-3 (18-3). С. 462-464.