

# VXReady – принципы работы библиотеки

---

## Оглавление

Ключевые аспекты VXReady .....	1
Поддержка адаптивности .....	5
VXReady: Принципы работы.....	6
Списки .....	6
Представления .....	8
Как работает система управления представлениями .....	9
Маркеры.....	10
Что дает VXReady.....	10
Кастомизация системных представлений .....	11
Создание собственных представлений .....	11

## Ключевые аспекты VXReady

На сайтах, в том числе и созданных на базе Bitrix Framework могут изобиловать списки, каждый элемент в которых является подобием друг друга (т.е. повторяющийся код). Это можно представить как набор одинаковых элементов

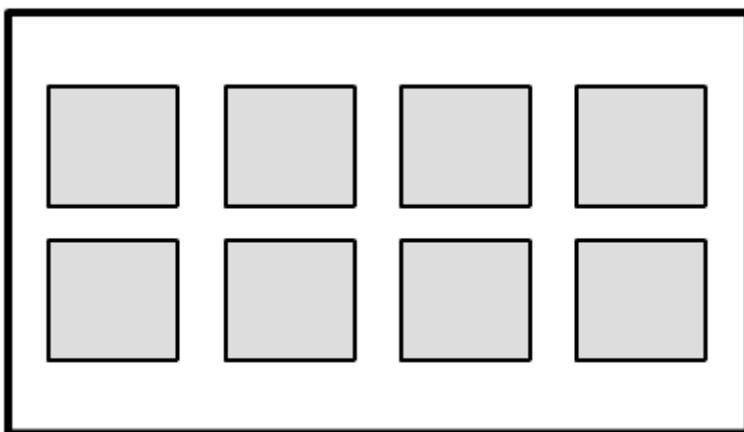


Рисунок 1

Могут встречаться даже списки, создаваемые разными компонентами и на разных страницах и разделах сайта у которых опять же – повторяющийся код.

Следить за актуальностью таких списков в будущем тяжело.

В современном интернет-магазине, например, можно встретить до десятка и более различных шаблонов компонент, работающих с одним и тем же кодом.

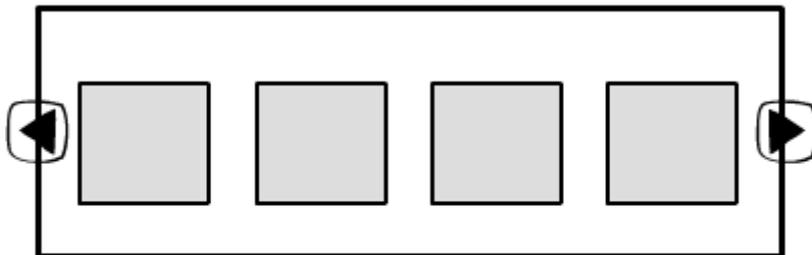


Рисунок 2. Список элементов в виде слайдера

Это могут быть даже слайдеры элементов (товаров)

VXReady упрощает работу со списками элементов.

Каждый элемент начинает выступать просто как отдельная сущность, имеющая свою обертку.

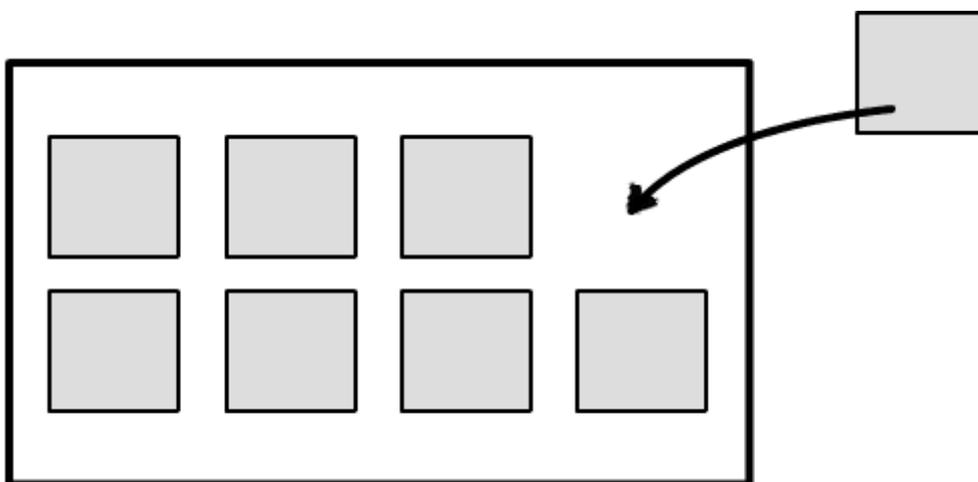


Рисунок 3

Мы работаем с этой сущностью только в одном месте.

Но более того.

VXReady дает возможность любому веб-мастеру быстро и просто создавать свои сущности или модифицировать уже существующие. Это можно делать даже в горячем режиме.

При этом любые модификации распространяются на все участки, где используется такая сущность.

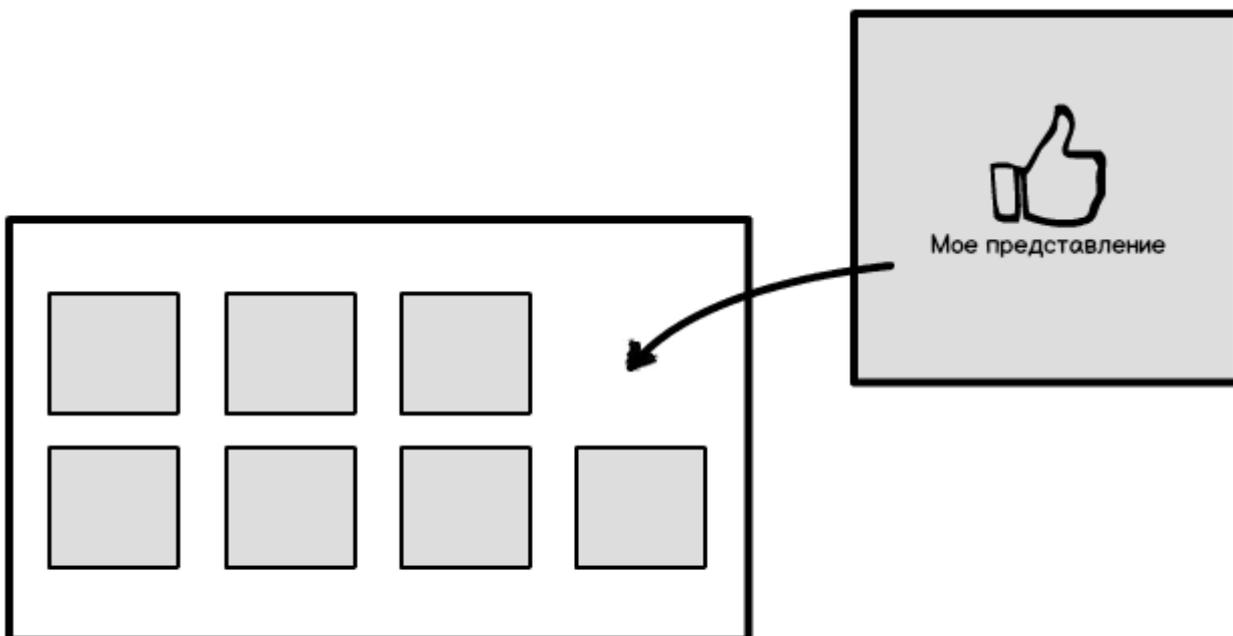


Рисунок 7. Есть возможность создавать свои собственные представления!!!

Для удобства – мы называем такие сущности в VXReady просто **ПРЕДСТАВЛЕНИЯМИ**

*Например: В готовое решении **VXReady: Интернет-магазин** ([marketplace.1c-bitrix.ru/solutions/alexkova.market/](http://marketplace.1c-bitrix.ru/solutions/alexkova.market/)) мы уже включили библиотеку VXReady и практически все используемые в решении компоненты работают с представлениями.*

*Даже нелюбимые многими компоненты БигДата и ее несколько, имеющие тяжелый для переработки код уже готовы и настроены для этого*



А VXReady позволит использовать собственные представления быстро и оперативно

*Когда мы создавали BXReady, мы думали о выгоде владельцев интернет- проектов плюс постоянно помнили про веб-мастера. Ведь именно ему предстоит **создавать, сопровождать и модернизировать**. Но самое главное **создавать свое**. И так, чтобы это свое сразу же автоматически становилось часть нашей общей системы BXReady.*

## Поддержка адаптивности

- 1. Любое представление должно быть адаптивным (т.е. подстраиваться под предоставляемую ему область показа в сетке Bootstrap или любом другом контейнере)
- 2. Представление может поддерживать любую ширину области в которой оно выводится, либо иметь только «условные» ограничения. Не все представления одинаково хорошо выглядят при любой ширине, но это допустимо, в данном случае веб-мастер, создающий представление может указать рекомендуемые параметры отображения для него. Все равно представление должно поддерживать принцип 1.

## BXReady: Принципы работы

### Основные сущности

В редакции 1.0.0 BXready есть две основные сущности

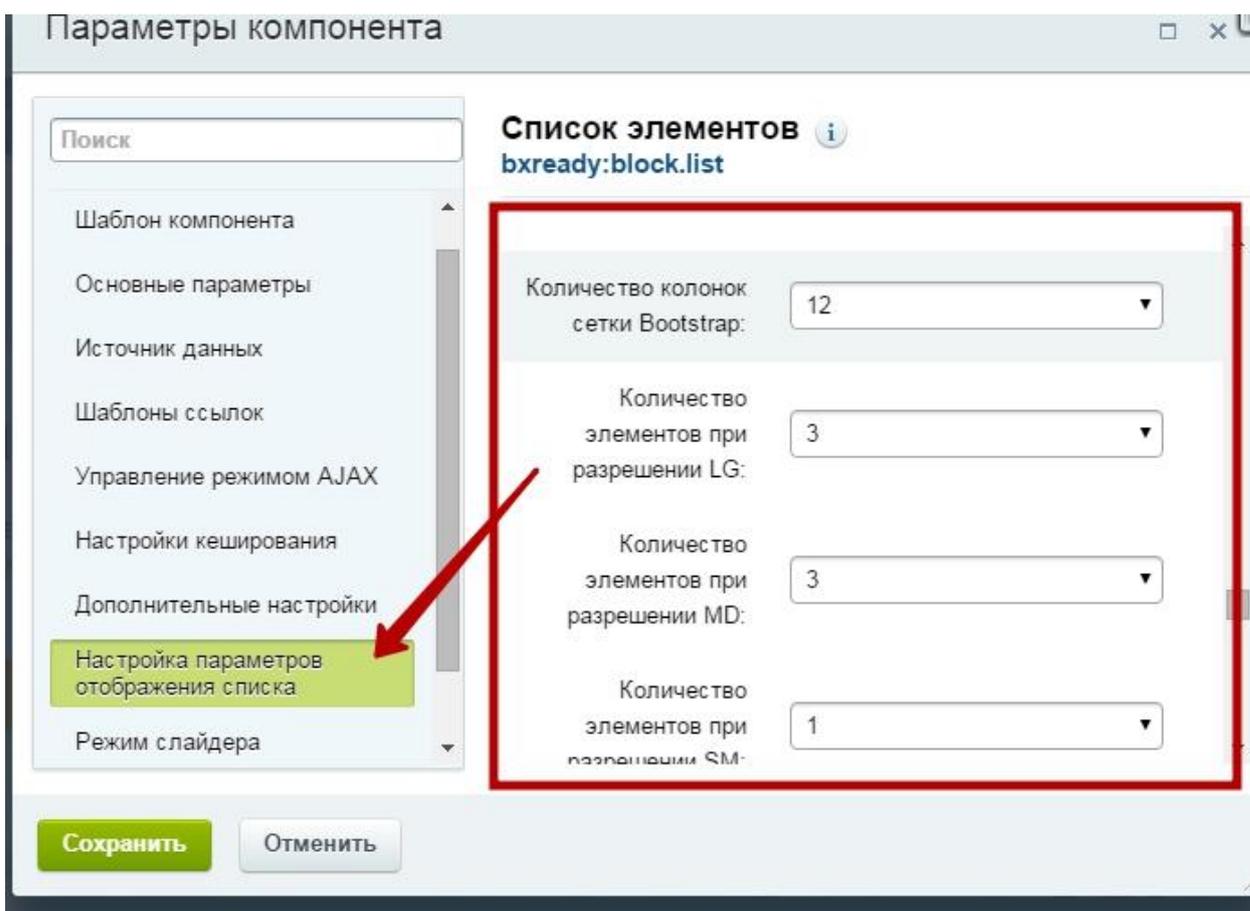
1. Представление
2. Список

### Списки

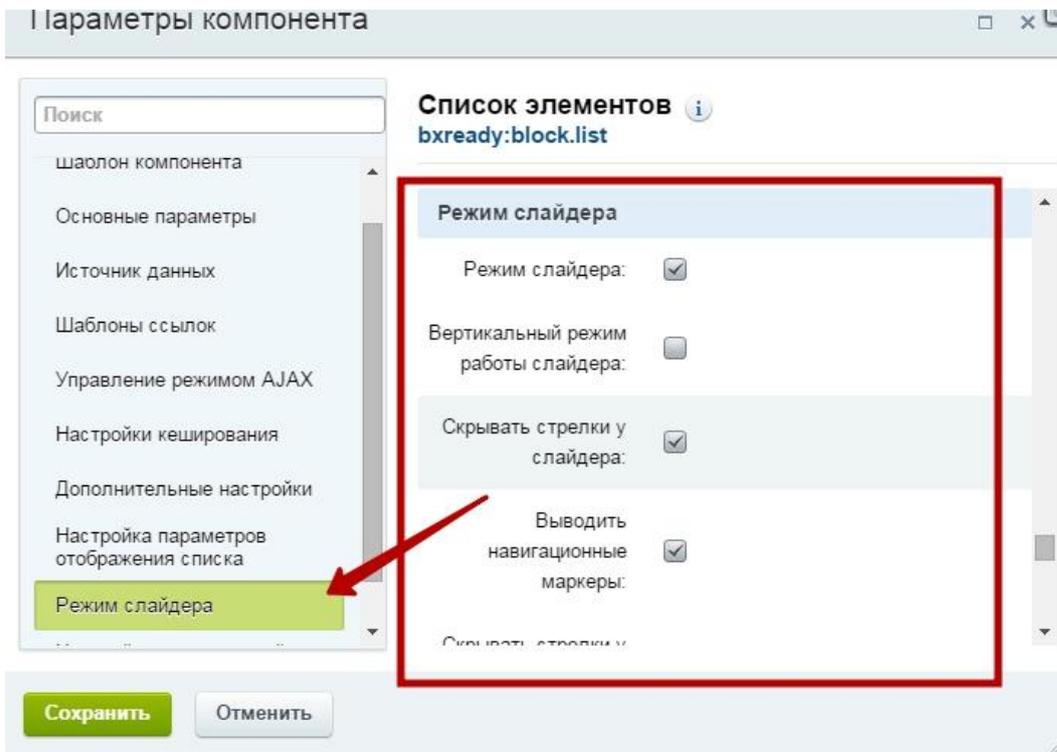
Списки элементов представлены двумя компонентами **block.list** (обертка для компонента news.list) и **ecommerce.list** (обертка для компонента catalog.section)

Помимо всех возможностей, которые присущи типовым компонентам Битрикс эти компоненты содержат:

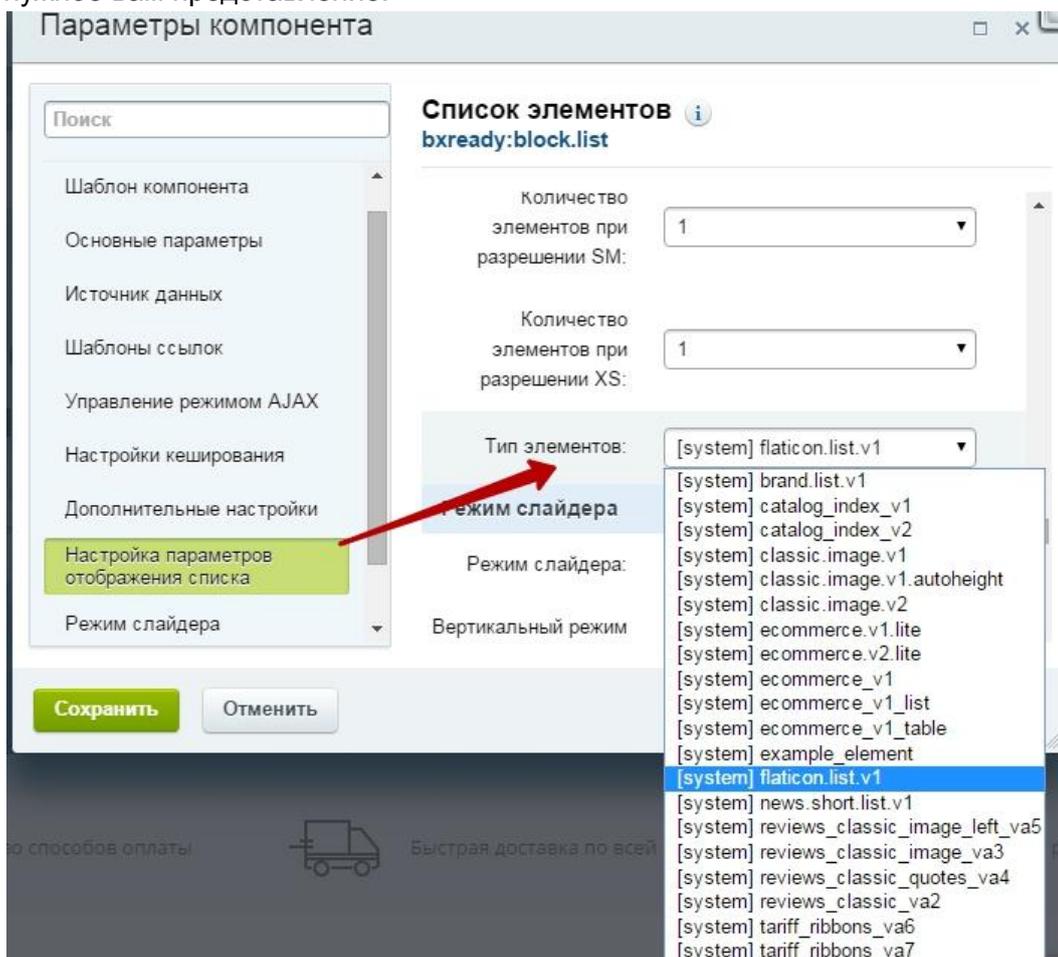
1. дополнительные настройки для управления адаптивностью (Ноу-хау решений BXReady)



2. настройку параметров показа в виде слайдера, а также настройки тюнингующие слайдер



3. возможность привязки к представлению из библиотеки элементов. Просто выберите нужное вам представление.



Т.е. достаточно вам разместить любой из этих компонентов – вы автоматически получаете **готовые списки с параметрами управления адаптивностью, но самое главное – подключенные к системе управления представлениями BXReady**

Также планируется выпуск и специальных комплексных компонент (ряд из них уже используется нами)

## Представления

Представление может состоять всего из одного файла element.php

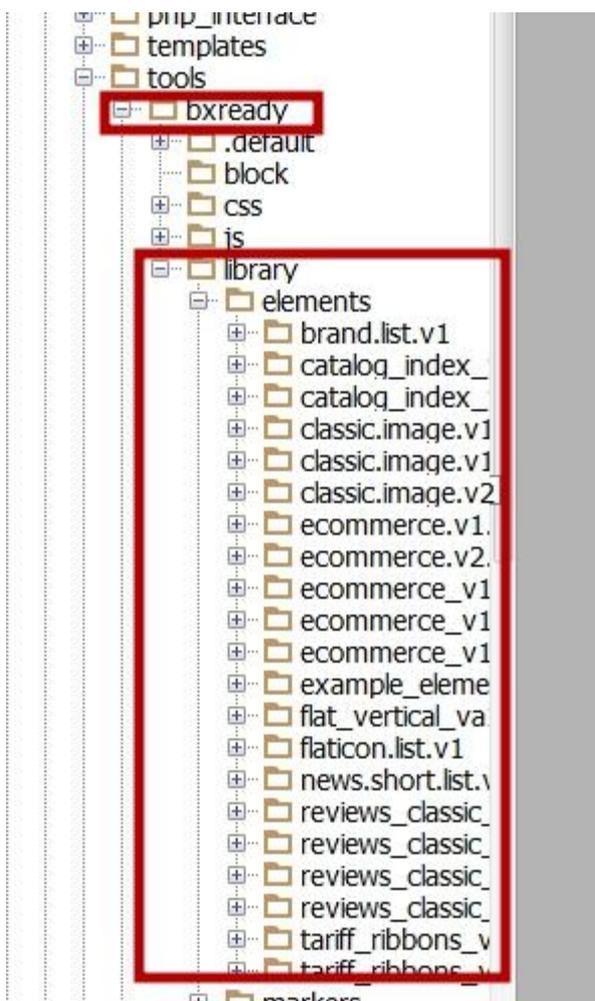
Который должен размещаться в папке  
/bxready/library/elements/#КОД\_ШАБЛОНА\_ПРЕДСТАВЛЕНИЯ#/

Вашего шаблона сайта

Либо по аналогичному пути в шаблоне **.default**

Системные же представления хранятся в папке в /bitrix/tools/bxready/library/elements/

Пример представлений с одного из проектов



## Как работает система управления представлениями

Если в какой то момент времени система получает задачу отрисовать представление с кодом **my1** сначала проверяется его наличие в шаблоне сайта, затем в шаблоне сайта .default, затем уже в системной библиотеке.

Условие наличия представления – наличие в папке с кодом представления файла element.php

Этот файл подключается в месте вызова, и на вход его подаются массивы \$arElement и \$arElementParams содержащие данные для прорисовки элемента и дополнительные параметры (их определяет тот, кто вызывает представление, мы в своих компонентах передаем данные параметров компонента, из которого происходит вызов)

Задача файла element.php просто сформировать на основе поступивших данных требуемый HTML

Если для представления требуется подключение каких то CSS или JS – используйте специальную конструкцию

```
$elementDraw = \Alexkova\Bxready\Draw::getInstance(); // иницируем объект рисования представлений
$dirName = str_replace($_SERVER["DOCUMENT_ROOT"], '', dirname(__FILE__));
$elementDraw->setAdditionalFile("CSS", $dirName."/include/style.css", false); //подключение файла стилей
$elementDraw->setAdditionalFile("JS", $dirName."/include/script.js", false); //подключение файла стилей
```

В компонентах BXReady данная конструкция позволяет вынести подключаемые файлы в параметры кеша шаблона, из которого вызываются представления.

Таким образом вы можете создать выделенный css и js для любого из представлений и они будут подключаться только там, где эти представления используются.

Это поможет вам экономить ресурсы и вынести код в отдельные файлы

Пример файла представления



The image shows a file explorer on the left and a code editor on the right. The file explorer displays a directory structure with 'element.php' highlighted in a red box. The code editor shows the content of 'element.php', which is a PHP script that uses the Alexkova\Bxready\Draw class to generate HTML output. The code includes comments and logic for handling preview and detail pictures, and for outputting HTML tags for a container, image, and name.

```
k?
use Alexkova\Bxready\Draw;
$elementDraw = \Alexkova\Bxready\Draw::getInstance($this);
$arMatrix = array('width' => 280, 'height' => 140);

//echo "<pre>"; print_r($arElementParams); echo "</pre>";

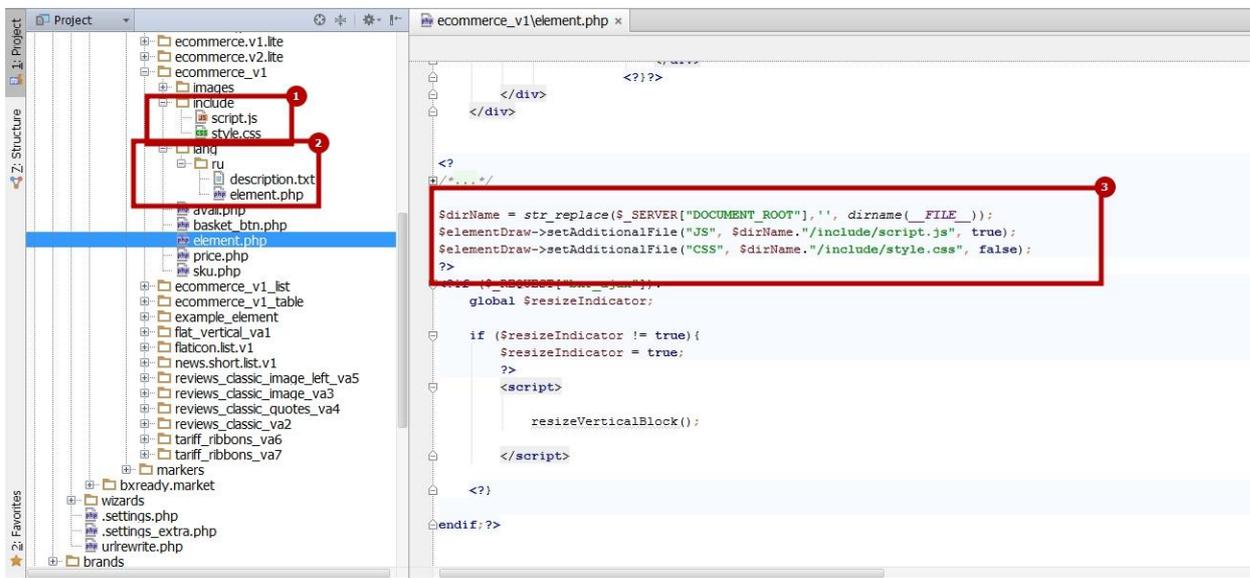
if (is_array($arElement["PREVIEW_PICTURE"])){
    $picture = $elementDraw->prepareImage($arElement["PREVIEW_PICTURE"]["ID"], $arMatrix);
}else{
    if (is_array($arElement["DETAIL_PICTURE"])){
        $picture = $elementDraw->prepareImage($arElement["DETAIL_PICTURE"]["ID"], $arMatrix);
    }
}

if (!is_array($picture) || strlen($picture["src"])<=0){
    $picture = array("src" => $elementDraw->getDefaultImage());
}

?>

<div class="bxr-classic-image-v1" data-uid="<?=$arElementParams["UNICUM_ID"]?>">
    <div class="bxr-element-container">
        <div class="bxr-element-image">
            ">
        </div>
        <div class="bxr-element-name">
            <a href="<?=$arElement["DETAIL_PAGE_URL"]?>"><?=$arElement["NAME"]?></a>
        </div>
    </div>
    </div>
```

Пример более сложного представления, включающего 1-дополнительные файлы, 2- дополнительные подключаемые скрипты-фрагменты и 3- пример подключения дополнительных CSS и JS библиотек с учетом возможного кэширования компонента.



## Маркеры

Частный случай представлений – маркеры

Мы привыкли видеть на товарах например ярлыки новинка и пр.

Но их вывод тоже всегда один и тот же код

Маркеры были вынесены в отдельный тип представлений, хранятся в отдельной папке

Vxready/library/markers/

Но по сути – это 100% те же элементы.

Маркеры могут подключаться в самих представлениях всего лишь одним вызовом API.

*Планируется пошаговая статья на базе примера создания реального представления из Vxready: Интернет-магазин*

## Что дает VXReady

Представьте, что какой-то элемент (его представление) вас перестали устраивать.

Есть два варианта действий

1. Перенастроить компоненты на новое представление

2. Просто изменить представление и автоматически все изменения отразятся во всех компонентах, которые его показывают

И тот и другой вариант имеют свои преимущества и недостатки и различную ценность в той или иной ситуации.

Но все они дают огромную экономию времени в процессе разработки и эксплуатации

## Кастомизация системных представлений

Представьте, что вам понадобилось изменить что-то в списках элементов

Если речь о системных представлениях – просто скопируйте их себе в ваш шаблон или шаблон `.default`

И все

Теперь по правилам подключения представлений – система найдет одноименное у вас и подключит ваше а не системное.

Так вы можете получить контроль над всем проектом и в соответствии со стандартами Bitrix Framework, не вторгаясь в чужие пространства имен или в ядро продукта.

В VXReady: Интернет магазин мы планируем реализовать еще ряд возможностей поддержки VXReady, которые сделают процедуру управления представлениями еще проще.

## Создание собственных представлений

*Планируется пошаговая статья на базе примера создания реального представления из Vxready: Интернет-магазин*

## Предложения

Мы всегда рады вашим предложениям и пожеланиям

Присылайте их на адрес [kova@kuznica74.ru](mailto:kova@kuznica74.ru)