



# 1С-Битрикс: Управление сайтом

Быстрый старт разработчика

Модули





## Содержание

<b>Введение .....</b>	<b>3</b>
<b>Глава 1. Модули.....</b>	<b>4</b>
СТРУКТУРА ФАЙЛОВ .....	6
ОПИСАНИЕ И ПАРАМЕТРЫ .....	7
ВЗАИМОДЕЙСТВИЕ МОДУЛЕЙ .....	8
Явное взаимодействие .....	8
Взаимодействие через события .....	8
Пример взаимодействия.....	8



## Введение

---

Учебный курс для начинающего разработчика. Вы первый раз обращаетесь к популярной платформе Bitrix Framework и не знаете откуда начать? Начните с этого курса.

**⚠ Внимание!** Курс *Быстрый старт разработчика* - "выжимка" из огромного объема справочной информации по Bitrix Framework. Он создан с целью облегчить начинающему разработчику изучение системы и не является ни в коей мере заменой документации и базовых учебных курсов. В нем даются только основы с указанием более детальных источников информации.

При условии качественного изучения материалов курса, по его окончании специалист получит базовые понятия о платформе и ссылки на все, что может вам понадобиться при углубленном изучении Bitrix Framework.

Минимальный уровень знаний, необходимый при изучении:

- PHP
- HTML, CSS

**⚠ Примечание.** Для успешной разработки структуры информационных блоков сложных по структуре проектов, полезно будет получить хотя бы базовые знания по [реляционным базам данных](#).

**Список ссылок по теме:**

- [PHP](#)
- [HTML, CSS](#)



## Глава 1. Модули

*Bitrix Framework* имеет модульную структуру. Каждый модуль отвечает за управление определенными элементами и параметрами сайта: информационным наполнением и структурой сайта, форумами, рекламой, рассылкой, распределением прав между группами пользователей, сбором статистики посещений, оценкой эффективности рекламных кампаний и т.д.

**⚠ Модуль** - это модель данных и API для доступа к этим данным. Статические методы классов модуля могут вызываться в компонентах, шаблонах, других модулях. Также внутри контекста **Bitrix Framework** могут создаваться экземпляры классов.

Модули системы, главным образом, работают независимо друг от друга. Однако в целом ряде случаев функционал одних модулей основан на возможностях других. Например:

- Модуль **Торговый каталог** расширяет возможности модуля **Информационные блоки** и позволяет выполнять настройку цен товара в зависимости от различных условий, применять к товарам наценку и скидки и т.п.
- Модуль **Документооборот** позволяет организовать последовательную коллективную работу с содержимым модулей **Информационные блоки** и **Управление структурой**.

После установки системы список используемых модулей можно просмотреть на странице **Управление модулями** (*Настройки > Настройки продукта > Модули*) в административном разделе системы:

Управление модулями				
Рабочий стол > Настройки > Настройки продукта > Модули				
Название	Версия	Дата обновления	Статус	Действие
<b>Главный модуль</b> Ядро продукта с технологией "SiteUpdate".	10.0.1	18.03.2011	Установлен	
<b>AD/LDAP интеграция</b> Модуль для работы с Active Directory и LDAP.	10.0.0	15.03.2011	Установлен	Удалить
<b>Wiki</b> Модуль дает возможность ведения wiki-страниц на сайте.	10.0.0	09.03.2011	Установлен	Удалить
<b>Бизнес-процессы</b> Модуль для создания и работы с бизнес-процессами	9.5.3	28.12.2010	Установлен	Удалить
<b>Блоги</b> Модуль дает возможность ведения блогов на сайте.	10.0.1	09.03.2011	Установлен	Удалить
<b>Валюты</b> Модуль управления валютами позволяет управлять валютами сайта и их курсами.	10.0.0	15.03.2011	Установлен	Удалить
<b>Веб-аналитика</b> Модуль сбора и отображения статистики сайта.	10.0.1	23.03.2011	Установлен	Удалить
<b>Веб-кластер</b> Модуль поддержки веб-кластера.	10.0.1	24.12.2010	Не установлен	Установить
<b>Веб-сервисы</b> Модуль позволяющий организовать систему веб-сервисов и SOAP.	10.0.0	15.03.2011	Установлен	Удалить


Для экономии дискового пространства неиспользуемые модули рекомендуется удалить, при этом дистрибутив модуля остается в системе, и он в любое время может быть снова



установлен. При деинсталляции некоторых модулей система предлагает сохранить накопленные модулем данные (таблицы модуля). Если вы в дальнейшем планируете использовать эти данные, то при удалении модуля необходимо отметить соответствующую опцию.

Управление уровнем прав пользователей на доступ к модулям системы осуществляется отдельно для каждого модуля на странице его настроек. На этой же странице выполняется управление общими параметрами работы модулей.

Страница настроек конкретного модуля может иметь различное число вкладок и полей, в зависимости от функционала модуля. Перейти к ней можно следующими способами:

- с помощью административного меню: *Настройки > Настройки продукта > Настройки модулей > имя\_модуля*;
- с помощью кнопки **Настройки** , расположенной на административной панели. Данная кнопка позволяет перейти к настройкам модуля, страницы (формы) которого открыты в текущий момент в основной рабочей области.

## Модули и компоненты

Модули в *Bitrix Framework* представляют собой модели и контроллеры нижнего уровня (в понятиях **MVC**), а компоненты - контроллеры верхнего уровня, включающие представления, которые основаны на иерархии файловой структуры сайта. Весь функционал любого сайта, как правило, реализуется на стандартных модулях, но приходится кастомизировать компоненты (или написать свои) для формирования и вывода страниц и подключать их на соответствующих страницах сайта.

## Разработка модулей

**Bitrix Framework** допускает разработку пользовательских модулей.

### Список ссылок по теме:

- Раздел [Модули](#) в [Документации для разработчиков](#)
- Учебный курс [Администрирование. Часть 2](#) с описанием основных модулей.
- [Документация](#) по созданию собственного модуля



## Структура файлов

Структура файлов модуля:

- **/bitrix/modules/ID модуля/** - корневой каталог модуля
  - **/admin/** - каталог с административными скриптами модуля;
    - **menu.php** - файл с административным меню модуля;
  - **/classes/** - скрипты с классами модуля;
    - **/general/** - классы модуля, не зависящие от используемой базы данных;
    - **/mysql/** - классы модуля, предназначенные для работы только с MySQL;
    - **/oracle/** - классы модуля, предназначенные для работы только с Oracle;
  - **/lang/ID языка/** - каталог с языковыми файлами скриптов модуля;
  - **/install/** - каталог с файлами используемыми для инсталляции и деинсталляции модуля;
    - **/admin/** - каталог со скриптами подключающими административные скрипты модуля (*вызывающие скрипты*);
    - **/db/** - каталог с SQL скриптами для инсталляции/деинсталляции базы данных;
      - **/mysql/** - SQL скрипты для инсталляции/деинсталляции таблиц в MySQL;
      - **/oracle/** - SQL скрипты для инсталляции/деинсталляции таблиц в Oracle;
    - **/images/** - каталог с изображениями используемыми модулем; после инсталляции модуля они должны быть скопированы в каталог **/bitrix/images/ID модуля/**;
    - **/templates/** - каталог с компонентами модуля;
      - **/ID модуля/** - каталог с основными файлами компонент;
      - **/lang/ID языка/ID модуля/** - в данном каталоге находятся языковые файлы компонент модуля;
    - **index.php** - файл с описанием модуля;
  - **include.php** - данный файл подключается в тот момент, когда речь идет о подключении модуля в коде, в нем должны находиться включения всех файлов с библиотеками функций и классов модуля;
  - **default\_option.php** - содержит массив с именем **\$ID модуля\_default\_option**, в котором заданы значения по умолчанию для параметров модуля;
  - **options.php** - данный файл подключается на странице настройки параметров модулей в административном меню **Настройки**;
  - **prolog.php** - файл должен подключаться во всех административных скриптах модуля, помимо всего прочего в нем должны быть определены следующие две константы:

**ADMIN\_MODULE\_NAME** - идентификатор модуля;

**ADMIN\_MODULE\_ICON** - HTML код для большой иконки модуля выводимой над заголовком страницы.



## Описание и параметры

### Описание

Каждый модуль должен быть корректно описан в системе для того, чтобы система знала, как с этим модулем работать. Некорректно описанные модули могут привести к полной или частичной неработоспособности системы (например, может не работать система обновлений).

Основным файлом используемым системой для манипуляции модулем является **/bitrix/modules/ID модуля/install/index.php**. Основное назначение этого файла - это размещение в нем класса с именем, совпадающим с **ID модуля**.

Обязательные методы этого класса:

- `DoInstall` - запускается при нажатии кнопки **Установить** на странице **Модули административного раздела**, осуществляет инсталляцию модуля.
- `DoUninstall` - запускается при нажатии кнопки **Удалить** на странице **Модули административного раздела**, осуществляет деинсталляцию модуля.

Необязательный метод этого класса:

- `GetModuleRightList` - возвращает список уникальных прав (или ролей) модуля.

Обязательные свойства объекта этого класса:

- `MODULE_ID` - хранит **ID модуля**;
- `MODULE_VERSION` - текущая версия модуля в формате **XX.XX.XX**;
- `MODULE_VERSION_DATE` - строка содержащая дату версии модуля; дата должна быть задана в формате **YYYY-MM-DD HH:MI:SS**;
- `MODULE_NAME` - имя модуля;
- `MODULE_DESCRIPTION` - описание модуля;
- `MODULE_GROUP_RIGHTS` - если задан метод `GetModuleRightList`, то данное свойство должно содержать **Y**.

### Параметры

**Параметры модуля** доступны для изменения в административном интерфейсе на странице **Настройки модулей** (Настройки > Настройки продукта > Настройки модулей). При выборе модуля на данной странице, система подключает файл **/bitrix/modules/ID модуля/options.php**, предназначенный для управления параметрами модуля, назначения прав на модуль и т.п.

Параметры модуля хранятся в базе данных.

Для работы с параметрами модуля предназначен класс `COption`.



## Взаимодействие модулей

Модули могут взаимодействовать между собой двумя способами: явно (прямым вызовом) и скрыто (через систему событий).

### Явное взаимодействие

Явное взаимодействие подразумевает:

- Подключение модуля с помощью функции `CModule::IncludeModule`;
- Непосредственный вызов метода класса или функции модуля.

### Взаимодействие через события

**⚠ Событие** - это какое либо произвольное действие, в момент выполнения которого (до или после) собираются все обработчики этого события и выполняются по одному.

Сущность **событие** позволяет сделать модули максимально независимыми друг от друга. Модуль ничего не знает об особенностях функционирования другого модуля, но может взаимодействовать с ним через интерфейс событий.

Схема работы с событиями. Модуль, инициирующий событие, в том месте кода, где это событие происходит, должен выполнить следующее:

- Собрать все зарегистрированные обработчики с помощью функции `GetModuleEvents`.
- Выполнить их по одному с помощью функции `ExecuteModuleEvent`, обрабатывая соответствующим образом возвращаемые обработчиками значения.

В свою очередь, модуль, который хочет выполнить какие либо действия на это событие, должен:

- Зарегистрировать в момент инсталляции свой обработчик с помощью функции `RegisterModuleDependences`.
- Соответственно необходимо иметь эту функцию-обработчик и убедиться, что скрипт, в котором эта функция находится, подключается в файле `/bitrix/modules/ID модуля/include.php`.

### Пример взаимодействия

Ярким примером взаимодействия такого типа является взаимодействие модулей системы с модулем **Поиска**. Этот модуль не имеет никакой информации о данных других модулей, особенностях их хранения и обработки. Он только предоставляет интерфейс для индексации данных. Любой модуль системы, который должен индексироваться, при инсталляции регистрирует обработчик на событие `OnReindex`. Каждый такой обработчик в свою очередь возвращает данные для индексации, которые модуль **Поиска** использует для наполнения своей базы.