

Документация

Расширенные настройки продукта **native.settings**

Версия 1.3.1 от 04.02.2026

Содержание

Страниц: 9 · Слов: 533 · Время чтения: ≈ 3 мин

Обзор модуля	3
Обновления	5
API модуля	7
Пример использования	9

Обзор модуля

Название	Описание
Наименование	Расширенные настройки продукта
Код модуля	native.settings
Дата релиза	14.07.2022
Дата обновления	04.02.2026
Версия	1.3.1

Позволяет создать собственные настройки для сайта, с их расширенной конфигурацией, которые позже можно использовать в скриптах сайта.

Основная специфика модуля в том, что имеется возможность указать гораздо больше настроек для параметра, чем это допускают штатные настройки параметров. Помимо расширенных настроек для параметров, их также можно распределить по разделам для лучшего удобства работы с ними.

Главная задача, которую решили с помощью модуля - максимально ушли от использования констант и/или часто используемых переменных в коде.

Например, добавляем параметр режима работы магазина, прописываем получение параметра в нужном шаблоне сайта, а далее, при необходимости, администратор сможет самостоятельно, через админ раздел менять значение параметра. Без необходимости постоянно править код шаблона.

Возможности

- Размещение параметров по разделам меню.
- Задать тип значения параметра.
- Указать зависимость от языка, модуля, сайта, группы пользователя.
- Запретить переписывать значение параметра в ходе отработки скриптов.
- Логирование в Журнал событий действий связанных с параметрами.
- При возврате значений модуль приводит значение параметра к указанному типу.
- В случае возникновения ошибок в процессе получения значения, все ошибки будут залогированы в системный файл лога, обычно это `__bx_log.log`

- Кеширование данных модуля производится штатными функциями платформы.
- Время кеширования данных таблицы можно регулировать штатными средствами, с помощью файла `/bitrix/.settings.php`, через массив `cache_flags`.

Особенности

- Модуль устанавливается в глобальное меню - Расширения.
- Все настройки можно выполнять только из-под пользователя с правами администратора.
- Минимальная версия PHP ≥ 8.1 (согласно требованиям платформы).

Обновления

Релизы: 2026

1.3.1

04.02.2026

✓ Текущая версия

Тип	Описание
	Добавлен обязательный закрывающий тег.
	Обновлена ссылка на Документацию в меню модуля.

1.3.0

04.01.2026

Тип	Описание
	Техническое обновление.
	Улучшена поддержка версии PHP выше 8.1.
	Незначительно поправлены стили полей.

Релизы: 2022

1.2.0

22.08.2022

Тип	Описание
	Поправлена работа модуля на серверах с Microsoft Windows.
	В меню модуля добавлена ссылка на документацию.

1.1.0

03.08.2022

Тип	Описание
	Добавлена проверка версии PHP при переходе по страницам модуля в административном разделе.

	Добавлена очистка <code>managed_cache</code> .
	При использовании кнопки тулбара для очистки кеша, теперь удаляется и параметр <code>mode</code> из текущего url.
	Улучшена логика переходов по страницам модуля при наличии разделов.
	Поправлено удаление записей в режиме - Для всех.
	Мелкие правки.

API модуля

Подключение модуля

Для использования методов модуля в публичной части сайта, предварительно необходимо подключить модуль на странице.

PHP

```
1 \Bitrix\Main\Loader::includeModule('native.settings');
```

Получение значения параметра

В случае отсутствия значения для параметра или невозможности отработки условий при получении значения - будет возвращено **N**.

PHP

```
1 // Название параметра: schedule
2 // Значение параметра по умолчанию: N
3
4 $value = \Native\Settings\Option::get('schedule', 'N');
```

Установка значения параметра

На время отработки скрипта, если параметр не только для **чтения** - будет установлено значение **Y**.

PHP

```
1 // Название параметра: schedule
2 // Значение параметра: Y
3
4 \Native\Settings\Option::set('schedule', 'Y');
```

Типы значений параметров

PHP

```
1 \Native\Settings\Option::getValueTypes();
```

CRUD

Во всех методах доступны штатные параметры ORM.

PHP

```
1 //Поля параметра
2 \Native\Settings\OptionTable::getMap();
3
4 // Получить список параметров
5 \Native\Settings\Option::getList(['select' => ['*'], 'filter' => []]);
6
7 // Добавить параметр
8 \Native\Settings\Option::add($data);
9
10 // Обновить параметр
11 \Native\Settings\Option::update($primary, $data);
12
13 // Удалить параметр
14 \Native\Settings\Option::delete($primary);
```

Крайне не рекомендуется манипулировать параметрами напрямую через таблицы базы данных.

Для манипуляции необходимо использовать описанные выше методы, так как при их отработке выполняются и другие сопутствующие действия.

Константы

PHP

```
1 \Native\Settings\Option::VALUE_TYPE_STRING; // Строка
2 \Native\Settings\Option::VALUE_TYPE_NUMBER; // Число
3 \Native\Settings\Option::VALUE_TYPE_ARRAY; // Массив
4 \Native\Settings\Option::VALUE_TYPE_JSON; // Json
5 \Native\Settings\Option::VALUE_TYPE_DATE; // Дата
6 \Native\Settings\Option::VALUE_TYPE_DATE_TIME; // Дата и время
```

Пример использования

PHP

```
1  use \Bitrix\Main\Loader;
2  use \Native\Settings\Option;
3
4  require $_SERVER['DOCUMENT_ROOT'] . '/bitrix/header.php';
5
6  // Получение параметра Графика работы (schedule)
7  try {
8      if(Loader::includeModule('native.settings')){
9          $schedule = Option::get('schedule', 'Круглосуточно');
10         echo $schedule;
11     }
12 } catch (LoaderException $e) {
13     echo $e->getMessage();
14 }
15
16 require $_SERVER['DOCUMENT_ROOT'] . '/bitrix/footer.php';
```