

Информация, необходимая для установки и эксплуатации фреймворка для создания высоконагруженных e-commerce-проектов

Общие сведения

Фреймворк для создания высоконагруженных e-commerce-проектов — программная платформа для разработки и эксплуатации интернет-магазинов, B2B-порталов и сложных веб-проектов электронной коммерции. Платформа устанавливается на серверную инфраструктуру заказчика или в иную согласованную среду и используется как технологическая основа для проекта заказчика.

Для работы фреймворка требуется наличие лицензированной CMS 1С-Битрикс: Управление сайтом. Указанная CMS является самостоятельным программным продуктом, приобретается заказчиком отдельно и не входит в состав поставки фреймворка. Фреймворк использует Bitrix как внешнюю программную платформу и административную среду, обеспечивая поверх неё расширенную прикладную логику, серверную обработку и интеграцию с внешними системами. Фреймворк не является публично распространяемым открытым программным обеспечением. Установка выполняется специалистами ООО "Интаро Софт" либо уполномоченными техническими специалистами заказчика при наличии доступа к закрытым репозиториям и установочным материалам.

Платформа объединяет:

- административную систему и структуру данных 1С-Битрикс;
- backend на Symfony;
- REST API для взаимодействия с frontend-приложением;
- модульную систему composer-пакетов и Symfony-бандлов;
- возможность разработки SPA/SSR frontend на Vue.js, Nuxt.js, React или иной согласованной технологии;
- интеграцию с поиском, каталогом, корзиной, пользователями, контентом и внешними системами проекта.

Требования к серверной инфраструктуре

Конкретные требования к инфраструктуре определяются масштабом проекта, предполагаемой нагрузкой, объёмом каталога, количеством пользователей, числом интеграций и требованиями к отказоустойчивости. Для типового развёртывания требуются:

- сервер с операционной системой семейства Linux;
- PHP версии 8.2 или выше, для новых установок рекомендуется PHP 8.4;
- веб-сервер Nginx с настройкой корня сайта на каталог публичной части;
- MySQL или совместимая СУБД, поддерживаемая установленной версией Bitrix;
- composer для установки PHP-зависимостей;
- Git и SSH-доступ к внутренним репозиториям ООО "Интаро Софт";
- Docker и Docker Compose при использовании контейнеризированного окружения;
- Elasticsearch или совместимый поисковый сервис при использовании модуля каталога и поиска;

- cron или аналогичный планировщик для регулярных фоновых задач;
- доступ к SMTP, SMS-шлюзам, платежным системам, службам доставки и другим внешним сервисам, если они предусмотрены проектом;
- система резервного копирования файлов и базы данных;
- средства мониторинга доступности, ошибок и производительности.

В локальной и тестовой среде может использоваться готовое Docker-окружение, содержащее необходимые сервисы для работы Bitrix и Symfony-части.

Требования к рабочему месту администратора и пользователя

Для работы с административной частью и сопровождения проекта пользователю требуется:

- персональный компьютер или ноутбук;
- современный веб-браузер с поддержкой актуальных стандартов HTML, CSS и JavaScript;
- стабильное подключение к сети "Интернет" или к корпоративной сети, где размещён проект;
- учётная запись с соответствующими правами доступа в административной панели;
- средства деловой коммуникации для взаимодействия с технической поддержкой.

Для технического специалиста, участвующего в установке или эксплуатации, дополнительно требуются:

- доступ по SSH к серверу;
- права на управление файлами проекта, настройками веб-сервера и переменными окружения;
- доступ к базе данных;
- доступ к журналам приложения, веб-сервера и системных сервисов;
- доступ к Git-репозиториям и composer-пакетам, если специалист выполняет обновления;
- навыки работы с Linux, PHP, composer, Docker, Bitrix и Symfony.

Информация, необходимая для установки

До начала установки заказчик или ответственная проектная команда предоставляет:

1. Доменное имя или список доменных имён проекта.
2. Доступы к серверной инфраструктуре или параметры создаваемой среды.
3. Требования к размещению: сервер заказчика, выделенная инфраструктура, облачная среда или локальный контур.
4. Параметры базы данных или разрешение на создание новой базы данных.
5. Доступ к DNS-зоне для настройки доменов.
6. SSL-сертификаты или разрешение на выпуск сертификатов.
7. Сведения о требуемых интеграциях: CRM, ERP, платежи, доставки, email, SMS, маркетинговые сервисы, складские системы.
8. Требования к миграции данных, если проект переносится с существующего интернет-магазина.
9. Требования к учётным записям администраторов и разграничению прав.
10. Требования к резервному копированию, мониторингу, журналированию и хранению персональных данных.

Если внедрение выполняется для существующего проекта на Bitrix, дополнительно требуются резервные копии файлов и базы данных, сведения о текущих модулях, шаблонах, интеграциях, доработках и регламентных заданиях.

Порядок установки

Типовая установка фреймворка включает следующие этапы:

1. Подготовка инфраструктуры. Устанавливаются и настраиваются Linux, веб-сервер, PHP, СУБД, composer, Git, Docker Compose при необходимости, поисковый сервис и иные инфраструктурные компоненты.
2. Получение исходного кода. Основной проект и необходимые модули клонируются из внутренних репозиториях ООО "Интаро Софт". Доступ выполняется по SSH с использованием учётных данных и ключей, выданных уполномоченным специалистам.
3. Подготовка структуры Bitrix. В проекте используется структура, при которой общее ядро располагается в каталоге core, а публичный сайт – в каталоге sites/sl. Для каталогов bitrix, local и upload настраиваются символические ссылки.
4. Установка или восстановление Bitrix. Выполняется чистая установка Bitrix, установка демонстрационной версии или восстановление из резервной копии существующего проекта в зависимости от сценария внедрения.
5. Настройка Bitrix. Заполняются файлы конфигурации базы данных и системных настроек. В конфигурации указывается путь к composer-конфигурации проекта.
6. Установка PHP-зависимостей. Выполняется composer install с подключением закрытых composer-пакетов фреймворка и зависимостей Symfony.
7. Настройка переменных окружения. Создается файл локальных переменных окружения, задаётся секрет приложения, параметры базы данных, окружение выполнения, пути, интеграционные ключи и другие настройки проекта.
8. Генерация ключей безопасности. Для JWT-аутентификации создаётся пара ключей, используемая при выпуске и проверке токенов.
9. Применение миграций. Выполняются миграции Bitrix и Doctrine, необходимые для создания или обновления структуры данных.
10. Настройка маршрутизации API. В Bitrix настраивается маршрутизация запросов /api/* на Symfony-приложение через специальную точку входа.
11. Настройка frontend. Подключается или разворачивается frontend-приложение, взаимодействующее с backend через API. Конкретная технология frontend определяется проектом.
12. Первичная проверка. Проверяются административная панель Bitrix, публичное API, авторизация, регистрация, каталог, поиск, корзина, оформление заказа, контентные разделы и интеграции.
13. Передача в эксплуатацию. Формируются доступы, инструкции, регламенты резервного копирования, мониторинга и сопровождения.

Эксплуатация системы

В процессе эксплуатации выполняются следующие регулярные действия:

- контроль доступности сайта и административной панели;
- контроль ошибок backend, frontend, веб-сервера и фоновых процессов;

- контроль состояния базы данных, дискового пространства, кеша, очередей и поискового индекса;
- резервное копирование файлов проекта, пользовательских загрузок и базы данных;
- проверка успешности регламентных заданий и интеграционных обменов;
- обновление платформенных модулей и проектного кода по согласованному регламенту;
- обновление сертификатов, ключей и интеграционных токенов;
- управление учётными записями и правами доступа;
- анализ производительности и масштабирование инфраструктуры при росте нагрузки.

Публичный frontend взаимодействует с backend через API. API построено как интерфейс без сохранения состояния: серверная часть не должна хранить пользовательское состояние в PHP-сессиях для запросов к API. Для авторизации используются JWT-токены и refresh-токены. Данные, требующие длительного хранения, сохраняются в базе данных или передаются клиентом в запросах в соответствии с логикой проекта.

Обновление системы

Обновление фреймворка выполняется по согласованной процедуре:

1. Определяется состав обновления: платформенные пакеты, проектный код, конфигурация, миграции, frontend, инфраструктура.
2. Создаётся резервная копия рабочей среды или проверяется актуальность существующих резервных копий.
3. Обновление устанавливается на тестовую среду.
4. Выполняются composer install или composer update в согласованном объёме.
5. Применяются миграции Bitrix и Doctrine.
6. Очищается кеш Bitrix и Symfony.
7. Проверяются критичные пользовательские сценарии.
8. Согласуется окно установки на рабочую среду.
9. Выполняется установка на рабочую среду.
10. После установки проверяются логи, доступность API, административная панель, оформление заказа, поиск и интеграции.

Обновления должны выполняться квалифицированными специалистами, так как система содержит несколько взаимосвязанных уровней: Bitrix, Symfony, composer-пакеты, базу данных, поисковый индекс, frontend и внешние интеграции.

Резервное копирование и восстановление

Для надёжной эксплуатации рекомендуется резервировать:

- базу данных Bitrix и проектных модулей;
- каталог пользовательских загрузок;
- конфигурационные файлы окружения;
- исходный код рабочей версии проекта или ссылку на соответствующий Git-коммит;
- поисковые индексы, если их восстановление из базы данных занимает значительное время;
- файлы frontend-сборки, если они размещаются отдельно.

Периодичность резервного копирования определяется требованиями заказчика к допустимой потере данных и времени восстановления. Перед существенными обновлениями резервная копия обязательна.

Требования к квалификации пользователей

Пользователь административной панели должен обладать навыками работы с веб-интерфейсами, пониманием бизнес-процессов интернет-магазина, правил ведения каталога, обработки заказов, управления контентом и пользовательскими данными.

Технический администратор должен обладать знаниями Linux, веб-серверов, PHP, СУБД, Bitrix, Symfony, composer, Git, Docker и базовыми навыками диагностики веб-приложений. Для проектов с высокой нагрузкой дополнительно требуются навыки мониторинга, оптимизации SQL-запросов, работы с кешем, очередями и поисковыми системами.

Разграничение ответственности

ООО "Интаро Софт" отвечает за разработку, поставку, внедрение и сопровождение фреймворка в пределах условий договора. Заказчик отвечает за предоставление достоверных исходных данных, доступов, требований, инфраструктуры и своевременное согласование решений, если иное не предусмотрено договором.

После внедрения эксплуатация может выполняться заказчиком самостоятельно или передаваться ООО "Интаро Софт" по отдельному договору сопровождения. Конкретный состав работ, сроки реакции, регламент обновлений и зоны ответственности фиксируются в договоре или приложениях к нему.