

Функциональные характеристики программного обеспечения «Фреймворк для создания высоконагруженных e-commerce-проектов»

Назначение

Фреймворк предназначен для быстрого создания, запуска, развития и сопровождения интернет-магазинов, B2B-порталов и сложных digital-проектов электронной коммерции. Платформа применяется как технологическая основа проекта, на которой специалисты ООО "Интаро Софт" или уполномоченная команда заказчика реализуют бизнес-логику, пользовательские сценарии, интеграции и frontend-интерфейс под требования конкретного заказчика.

Основная цель фреймворка – сохранить преимущества привычной административной панели и структуры данных Bitrix, одновременно устранив ограничения типовой разработки на Bitrix за счёт современной backend-архитектуры Symfony, модульной структуры, API-first-подхода и возможности использовать современные frontend-технологии.

Ключевые принципы

Фреймворк построен на следующих принципах:

- Модульность. Функциональность разделена на независимые composer-пакеты и Symfony-бандлы, каждый из которых отвечает за отдельную область: пользователи, каталог, поиск, корзина, избранное, контент, меню, магазины, баннеры, фиды и вспомогательные возможности.
- API-ориентированность. Публичный пользовательский интерфейс взаимодействует с серверной частью через REST API. Это позволяет независимо развивать backend и frontend.
- Использование Bitrix как административной системы. Bitrix применяется для управления контентом, структурой данных, инфоблоками, пользователями, заказами и стандартными административными возможностями.
- Использование Symfony как прикладного backend-слоя. Symfony обеспечивает современную структуру кода, маршрутизацию, DI-контейнер, валидацию, сериализацию, безопасность, тестируемость и удобное развитие бизнес-логики.
- Независимый от состояния API. API-запросы проектируются без зависимости от серверной PHP-сессии. Для авторизации используются JWT-токены и refresh-токены.
- Расширяемость. Платформа предназначена для доработки под уникальные бизнес-процессы заказчика, включая интеграции с внешними системами.

Основные функциональные возможности

Фреймворк обеспечивает следующие группы функций.

Управление пользователями и авторизация

- регистрация пользователей;
- авторизация пользователей через API;
- выпуск и проверка JWT-токенов;
- обновление токенов через refresh-token-механизм;
- поддержка личного кабинета;
- интеграция с пользовательской моделью Bitrix;
- разграничение доступа к защищённым API-разделам;
- возможность расширения сценариев авторизации под требования проекта.

Каталог, поиск и фильтрация

- работа с товарным каталогом на основе структур данных Bitrix;
- использование настроек инфоблоков, свойств и умного фильтра Bitrix;
- индексация каталожных данных в Elasticsearch при использовании соответствующего модуля;
- быстрый поиск по каталогу;
- фильтрация товаров;
- подготовка данных каталога для frontend-приложения через API;
- возможность оптимизации каталога для проектов с большим ассортиментом и высокой нагрузкой.

Корзина и оформление заказа

- работа с корзиной пользователя;
- поддержка сценариев для авторизованных и неавторизованных пользователей;
- хранение данных корзины в соответствии с логикой проекта;
- базовая логика оформления заказа;
- передача выбранных параметров доставки, оплаты, промокодов и иных данных через API;
- возможность расширения процесса оформления заказа под бизнес-требования заказчика;
- интеграция с механизмами заказов Bitrix и внешними системами при необходимости.

Контент и административное управление

- использование Bitrix как административной панели;
- управление контентными страницами;
- управление баннерами;
- управление меню;
- работа с дополнительными типами свойств;
- использование инфоблоков и иных стандартных механизмов Bitrix для хранения и редактирования данных;

- сохранение привычных процессов работы для контент-менеджеров и администраторов, знакомых с Bitrix.

География, магазины и складская информация

- выбор города или местоположения;
- работа с магазинами через складские механизмы Bitrix;
- предоставление данных о магазинах и связанных сущностях через API;
- возможность адаптации логики доступности, остатков и региональных настроек под требования проекта.

Избранное, сравнение и пользовательские сценарии

- добавление товаров в избранное;
- поддержка избранного для авторизованных и неавторизованных пользователей;
- возможность реализации сравнения товаров;
- хранение пользовательских данных в базе данных или на клиентской стороне в зависимости от сценария;
- расширение пользовательских сценариев под конкретный интернет-магазин.

Фиды и интеграции

- формирование фидов на основании каталожных данных;
- подготовка данных для внешних маркетинговых, аналитических и товарных площадок;
- возможность интеграции с CRM, ERP, платёжными системами, службами доставки, email- и SMS-провайдерами;
- подключение проектных интеграций через сервисный слой Symfony;
- расширение состава интеграций без изменения базовой архитектуры платформы.

API и документация

- предоставление REST API для frontend-приложения;
- маршрутизация API-запросов через Symfony;
- настройка CORS для взаимодействия с frontend;
- сериализация и валидация данных;
- возможность генерации API-документации в формате OpenAPI;
- единый подход к обработке запросов, ответов и ошибок.

Frontend-взаимодействие

Фреймворк не ограничивает заказчика одной frontend-технологией. Платформа может использоваться совместно с:

- Vue.js;
- Nuxt.js;
- React;
- другим frontend-стеком, согласованным в рамках проекта.

Поддерживаются SPA- и SSR-подходы. SPA обеспечивает быстрый интерактивный пользовательский интерфейс, а SSR может применяться для улучшения первичной загрузки страниц, SEO и пользовательского опыта.

Архитектурные характеристики

Серверная часть фреймворка использует Symfony-структуру приложения:

- контроллеры принимают HTTP-запросы, выполняют первичную обработку и формируют ответы;
- сервисы содержат бизнес-логику;
- репозитории отвечают за получение и изменение данных;
- сущности отражают доменные объекты и модели хранения;
- DTO используются для передачи данных между слоями приложения;
- консольные команды применяются для фоновых и административных задач;
- миграции используются для управляемого изменения структуры данных.

Bitrix интегрирован в проект как ядро, административная панель и источник стандартных механизмов управления данными. Symfony-приложение подключается к Bitrix через интеграционный слой ядра фреймворка и обрабатывает API-запросы, поступающие на маршруты /api/*.

Производительность и масштабируемость

Фреймворк ориентирован на проекты электронной коммерции, где важны производительность, устойчивость к росту ассортимента и возможность дальнейшего развития без полной переработки архитектуры.

Для повышения производительности применяются:

- разделение frontend и backend;
- API-ориентированный подход;
- кеширование на уровне Bitrix и Symfony;
- использование Elasticsearch для поиска и фильтрации каталога;
- модульная структура, позволяющая подключать только необходимые возможности;
- возможность горизонтального и вертикального масштабирования инфраструктуры;
- независимое развитие backend, frontend и интеграционных сервисов.

Фактические показатели производительности зависят от инфраструктуры, объема данных, качества проектных доработок, настроек кеширования, поискового индекса и внешних интеграций.

Безопасность

Фреймворк использует возможности Symfony, Bitrix и проектной инфраструктуры для обеспечения безопасности веб-приложения. В платформе предусмотрены:

- разграничение доступа к API;
- JWT-аутентификация;
- refresh-token-механизм;
- подход без сохранения состояния для API;
- настройка CORS;
- использование механизмов Symfony Security;
- хеширование паролей средствами компонентов безопасности;
- валидация входящих данных;
- разделение публичных и защищенных маршрутов;

- возможность журналирования ошибок и событий;
- применение стандартных механизмов Bitrix для административной части.

Дополнительные меры безопасности, включая систему мониторинга сетевого трафика, сетевые ограничения, резервное копирование, мониторинг, защиту персональных данных и контроль доступа к инфраструктуре, определяются проектом и договором сопровождения.

Преимущества для заказчика

Использование фреймворка даёт заказчику следующие преимущества:

- ускорение запуска e-commerce-проекта за счёт готовой архитектуры и набора модулей;
- сохранение привычной административной панели Bitrix для контент-менеджеров и администраторов;
- более гибкая и поддерживаемая backend-разработка на Symfony;
- возможность использовать современные frontend-технологии и развивать интерфейс независимо от backend;
- масштабируемость для проектов с растущим ассортиментом, трафиком и числом интеграций;
- снижение зависимости от монолитной разработки на Bitrix-компонентах;
- возможность быстро внедрять уникальные бизнес-функции;
- повторное использование проверенных модулей на разных проектах;
- повышение качества кода за счёт современных практик разработки, тестирования и статического анализа.

Типовые сценарии применения

Фреймворк применяется в следующих сценариях:

- запуск нового интернет-магазина;
- модернизация существующего интернет-магазина на Bitrix;
- разработка B2B-портала;
- создание мультисайтовой e-commerce-платформы;
- развитие проекта с большим каталогом товаров;
- перенос части публичной логики на современный API и frontend;
- внедрение индивидуальных checkout-сценариев, интеграций и личных кабинетов;
- построение долгосрочной технологической основы для развития цифрового направления заказчика.

Ограничения и условия использования

Фреймворк является платформой для внедрения и разработки проекта, а не универсальным коробочным интернет-магазином, полностью готовым к эксплуатации без настройки. Для запуска конкретного проекта требуются установка, настройка, подключение модулей, разработка бизнес-логики, настройка frontend, интеграции и проверка пользовательских сценариев.

Состав доступных функций в конкретном внедрении зависит от технического задания, договора, выбранных модулей, инфраструктуры заказчика и объёма проектных доработок.