

Разработка решений в Creatio

Back-end (C#)

Версия 8.0



Эта документация предоставляется с ограничениями на использование и защищена законами об интеллектуальной собственности. За исключением случаев, прямо разрешенных в вашем лицензионном соглашении или разрешенных законом, вы не можете использовать, копировать, воспроизводить, переводить, транслировать, изменять, лицензировать, передавать, распространять, демонстрировать, выполнять, публиковать или отображать любую часть в любой форме или посредством любые значения. Обратный инжиниринг, дизассемблирование или декомпиляция этой документации, если это не требуется по закону для взаимодействия, запрещены.

Информация, содержащаяся в данном документе, может быть изменена без предварительного уведомления и не может гарантировать отсутствие ошибок. Если вы обнаружите какие-либо ошибки, сообщите нам о них в письменной форме.

Содержание

Back-end (C#)	4
Направления back-end разработки	4
Инструменты и утилитные возможности back-end разработки	6

Back-end (C#)

Основы

Разработка решений в Creatio предполагает разные уровни кастомизации в зависимости от сложности или типа бизнес-задачи. При этом следует учитывать, что уровень ядра является неизменяемым компонентом системы, а разработка в Creatio реализуется на уровне конфигурации.

Кастомизация приложений на уровне back-end реализуется в направлениях работы с данными, веб-сервисами, настройкой бизнес-логики объектов и т. д.

Для этих целей Creatio предоставляет набор различных инструментов — от разработки схемы [*Исходный код*] ([*Source code*]) или [*Действие процесса*] ([*User Task*]) во встроенной IDE до создания завершенных пользовательских проектов, подключаемых к приложению в качестве внешних библиотек.

Направления back-end разработки

На уровне back-end реализуются следующие направления разработки:

- ORM-модель данных и методы работы с ней.
- Реализация прямого доступа к базе данных.
- Создание и использование системных веб-сервисов.
- Настройка интеграции с внешними сервисами.
- Работа с компонентами системы и микросервисами.
- Расширенная настройка бизнес-процессов и встроенных процессов объектов приложения.
- Разработка бизнес-логики объектов.

ORM-модель данных и прямой доступ к базе данных

ORM-модель данных реализована в системе классами пространства имен `Terrasoft.Core.Entities` :

- `Terrasoft.Core.Entities.EntitySchemaQuery` — построение запросов выборки записей из таблиц базы данных с учетом прав доступа текущего пользователя.
- `Terrasoft.Core.Entities.Entity` — доступ к объекту, который представляет собой запись в таблице базы данных.

В большинстве случаев для доступа к данным рекомендуется использовать именно объектную модель, хотя прямой доступ к базе данных также реализован в back-end компонентах ядра.

Прямой доступ к базе данных предоставляет группа классов back-end ядра приложения из пространства имен `Terrasoft.Core.DB` :

- `Terrasoft.Core.DB.Select` — построение запросов выборки записей из таблиц базы данных.
- `Terrasoft.Core.DB.Insert` — построение запросов на добавление записей в таблицы базы данных.
- `Terrasoft.Core.DB.InsertSelect` — построение запросов на добавление записей в таблицы базы данных.

- `Terrasoft.Core.DB.Update` — построение запросов на изменение записей в таблице базы данных.
- `Terrasoft.Core.DB.Delete` — построение запросов на удаление записей в таблице базы данных.
- `Terrasoft.Core.DB.DBExecutor` — возможность создания и выполнения сложных запросов (например, с несколькими вложенными фильтрациями, различными комбинациями join-ов и т. д.) к базе данных.

Веб-сервисы

Сервисная модель Creatio предоставляет базовый набор веб-сервисов, с помощью которых может быть организована интеграция Creatio с внешними приложениями и системами.

Примеры **системных сервисов**:

- [odata](#) — обмен данными с Creatio по протоколу OData 4.
- [ProcessEngineService.svc](#) — запуск бизнес-процессов Creatio из внешних приложений.

Кроме системных, в Creatio реализованы **конфигурационные веб-сервисы**, предназначенные для вызова из клиентской части приложения.

Дополнительно к базовым сервисам разработчик имеет возможность создать **пользовательский веб-сервис**, предназначенный для решения узких бизнес-задач конкретного проекта.

Внешние сервисы

В ядре приложения реализованы классы, обеспечивающие интеграцию с внешними сервисами.

Например, пространство имен `Terrasoft.Social` предоставляет возможность интеграции с различными социальными сетями.

Компоненты системы и микросервисы

В ядре приложения реализованы классы, предоставляющие возможности для работы с компонентами системы и микросервисами. Например, пространство имен `Terrasoft.Sync` ядра приложения предоставляет классы встроенного механизма синхронизации с внешними хранилищами данных ([Sync Engine](#)), который позволяет создавать, изменять и удалять [Entity](#) в системе на основании данных из внешних систем и экспортировать данные во внешние системы. Процесс синхронизации осуществляется с помощью класса `SyncAgent`.

Бизнес-процессы и встроенные процессы объектов

Back-end разработка может потребоваться для **настройки сложных бизнес-процессов** (Элемент процесса [*Задание-сценарий*] ([*Script-task*])) либо для создания пользовательских **повторяющихся операций бизнес-процесса** (Конфигурационная схема [*Действие процесса*] ([*User Task*])).

Встроенные процессы объекта могут быть настроены как при помощи no-code инструментов, так и с использованием back-end разработки. Использование программного кода позволяет более гибко настроить поведение объекта в случае наступления определенных событий.

Бизнес-логика объектов

В Creatio есть возможность разрабатывать [бизнес-логику объектов](#) без использования событийных

подпроцессов. Для этого достаточно создать класс-наследник базового ядрового класса `Terrasoft.Core.Entities.Events.BaseEntityEventListener` и декорировать его атрибутом `EntityEventListener` с указанием имени сущности, для которой необходимо выполнить подписку событий. После чего можно переопределять методы-обработчики нужных событий.

Инструменты и утилитные возможности back-end разработки

Схема исходного кода

Основной возможностью back-end разработки является создание в пользовательском пакете схемы типа [\[Исходный код \]](#) ([*Source code*]).

Все изменения схемы, выполняемые в дизайнера, осуществляются в оперативной памяти. Чтобы изменения были сохранены на уровне метаданных схемы, схему следует сохранить. Для того чтобы изменения произошли на уровне базы данных, схему необходимо опубликовать.

Разработку схемы типа [\[Исходный код \]](#) ([*Source code*]) можно вести в [файловой системе](#) с использованием удобной IDE.

Конфигурирование бизнес-процессов

Включение специфической back-end логики в бизнес-процесс возможно с помощью программного кода, выполняемого элементом [\[Задание-сценарий \]](#) ([*Script-task*]).

Во время работы с бизнес-процессами в Creatio часто возникает необходимость выполнять однотипные операции. Для этих целей лучше всего подходит элемент [\[Выполнить действие процесса \]](#) ([*User task*]), для которого существует возможность выбрать наиболее подходящий в конкретной ситуации тип действия — [\[Пользовательское действие \]](#) ([*User task*]).

По умолчанию в системе доступно множество пользовательских действий, однако могут возникнуть ситуации, когда для выполнения определенного бизнес-процесса необходимо создать новое пользовательское действие.

Создать новое пользовательское действие можно с помощью конфигурационной схемы [\[Действие процесса \]](#) ([*User Task*]). В простой реализации действие процесса частично повторяет логику элемента процесса [\[Задание-сценарий \]](#) ([*Script-task*]), однако созданное однажды действие можно использовать многократно в разных процессах. А при внесении изменений в действие процесса, такие изменения сразу же будут применены ко всем процессам, в которых это действие использовалось.

Внешние библиотеки

В структуру пользовательского пакета могут быть включены внешние библиотеки, созданные пользователем. Это позволяет реализовать сложную логику, механизмы наследования, инкапсуляции при разработке специфического проектного решения.

Пакет-проект

Одним из инструментов Creatio для ускорения back-end разработки приложения является [пакет-проект](#). Это пакет, который позволяет разрабатывать функциональность как обычный C#-проект. Новая

функциональность в виде скомпилированной библиотеки и *.cs-файлов включается в [файловый контент пакета](#). При старте или перезапуске приложения Creatio собирает информацию о том, что в пакетах есть подготовленные библиотеки и сразу же подключает их в приложение.