

Back-end разработка

Автоматическая привязка данных

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

Автоматическая привязка данных	4
Структура автоматической привязки данных	4
Алгоритм работы автоматической привязки данных	5
Добавить пользовательскую структуру для привязки объекта	7
Вызвать автоматическую привязку данных	7

Автоматическая привязка данных



Автоматическая привязка данных позволяет в автоматическом режиме создать привязки данных объектов со сложной объектной моделью. В Creatio автоматическая привязка данных доступна к использованию для [каталога продуктов](#) в продуктах линейки Financial Services Creatio.

Структура автоматической привязки данных

Составляющие автоматической привязки данных представлены в таблице.

Составляющие автоматической привязки данных

Название	Назначение	Классы и интерфейсы
Сервис	По заданной структуре запускает генерацию привязок данных. Генерация выполняется в фоновом режиме .	<code>Service</code> — интерфейс для вызова автоматической привязки данных из front-end части.
Контроллер	Обращение к классам <code>Obtainer</code> (получение структуры связанных записей) и <code>Manufacturer</code> (генерация привязок данных).	<code>Controller</code> — класс для управления созданием привязок данных.
Генератор привязок	Является оберткой ядрового класса <code>PackageElementUtilities</code> , представляет собой удобный интерфейс для выполнения CRUD-операций с привязками данных. Операции с данными описаны в статьях Доступ к данным через ORM и Прямой доступ к данным .	<code>Manufacturer</code> — промежуточный класс для взаимодействия между контроллером, классом <code>PackageElementUtilities</code> и классом <code>PackageValidator</code> .
Получатель структуры	Возвращает в контроллер структуру связанных записей для основной сущности.	<code>Obtainer</code> — при формировании привязок данных позволяет гибко изменять структуру связанных объектов для основной сущности. Для определения требуемой реализации использует шаблон <code>Strategy</code> .

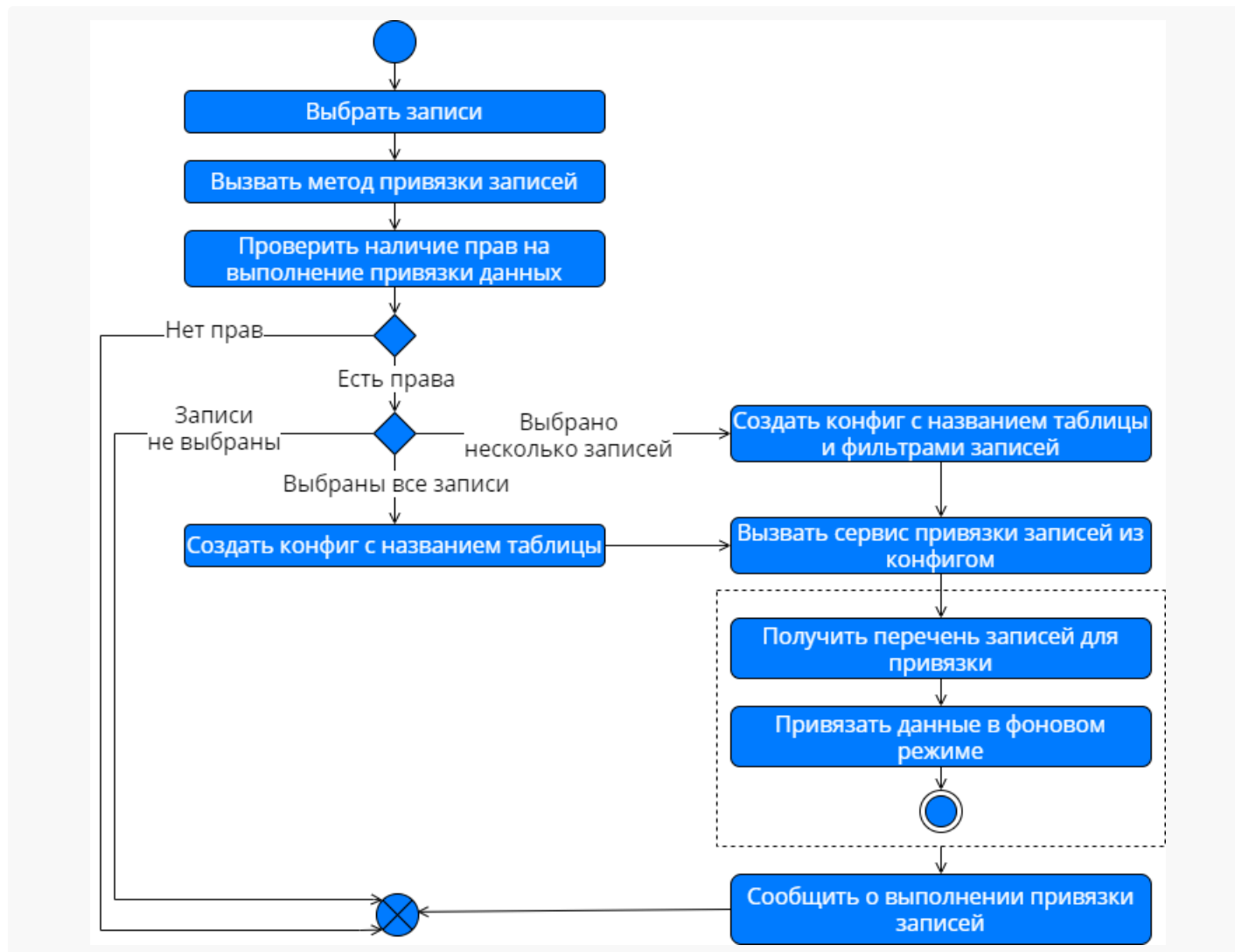
В системе реализована **базовая структура**, в которой возвращается только текущая сущность со всеми своими колонками, кроме системных. Свойства базовой структуры представлены в таблице ниже.

Свойства базовой структуры связанных сущностей

Свойство	Описание
EntitySchemaName	Название сущности.
FilterColumnPathes	Массив путей к первичной колонке основной сущности (при использовании фильтрации выполняется объединение через логический оператор <code>OR</code>).
IsBundle	<p>Способ сохранения записей.</p> <p>Способы сохранения записей:</p> <ul style="list-style-type: none"> • Сохранение записей в одну привязку. • Создание по одной привязке на каждую запись, связанную по фильтру с одной записью основной сущности.
Columns	Перечень колонок для привязки.
InnerStructures	Перечень структур вложенных объектов (например, детали, которые относятся к основной сущности).
DependsStructures	Перечень структур, от которых зависит текущая структура (например, справочники, на которые через справочные поля ссылается запись в текущей структуре).

Алгоритм работы автоматической привязки данных

Алгоритм работы автоматической привязки данных представлен на рисунке ниже.



Обработка структуры начинается с основной сущности и выполняется для каждого экземпляра класса структуры `SchemaDataBindingStructure`.

Алгоритм обработки структуры:

1. Рекурсивно обрабатывается каждая структура перечня структур, которые содержатся в свойстве `DependsStructures`. Это выполняется, чтобы при привязке основной сущности все объекты, на которые ссылается текущий, уже были привязаны.
2. По фильтру, который содержится в свойстве `FilterColumnPathes`, создается привязка к записи основной сущности.
3. Рекурсивно обрабатывается каждая структура из перечня структур, которые содержатся в свойстве `InnerStructures`. Поскольку структуры ссылаются на основную сущность, то их необходимо привязывать после основной сущности.

Способы именования автоматических привязок данных:

- `agb_EntitySchemaName`, если свойство `IsBundle` установлено в значение `true`. `EntitySchemaName` — наименование сущности текущей привязки.
- `agb_EntitySchemaName_PrimaryEntityId`, если свойство `IsBundle` установлено в значение `false`.

`PrimaryEntityId` — идентификатор записи основной сущности, с которой связаны записи в текущей привязке.

Важно. Не рекомендуется переименовывать автоматически созданные привязки, поскольку автоматически обновляются только привязки, имена которых соответствуют указанному шаблону.

Добавить пользовательскую структуру для привязки объекта

1. Создайте схему [*Исходный код*] ([*Source code*]). Для создания схемы воспользуйтесь статьей [Разработка конфигурационных элементов](#).
2. Создайте класс сервиса.
 - a. В дизайнере схем добавьте пространство имен `Terrasoft.Configuration`.
 - b. Создайте класс. Шаблон названия класса: `[НазваниеСущности]SchemaDataBindingStructureObtainer` (например, `ProductSchemaDataBindingStructureObtainer`). По указанному шаблону будет выполняться поиск классов при определении используемой реализации получателя (компонент `Obtainer`).
 - c. В качестве родительского класса укажите класс `BaseSchemaDataBindingStructureObtainer`.
 - d. Добавьте пользовательскую реализацию метода интерфейса `ObtainStructure()`. В этом методе сформируйте и верните необходимую структуру сущностей.

Метод `ObtainStructure()`

```
public override SchemaDataBindingStructure ObtainStructure(UserConnection userConnection)
```

Важно. При необходимости модификации реализации структуры, которая содержится в недоступных для редактирования пакетах, в пользовательском пакете создайте класс и используйте атрибут `[Override]`. Описание атрибута содержится в статье [Атрибут `\[Override\]`](#).

3. Опубликуйте схему исходного кода.

Вызвать автоматическую привязку данных

Автоматическую привязку данных можно вызвать из front-end или из back-end части.

Вызвать автоматическую привязку данных из front-end части

1. Вызовите сервис генерации привязок.
2. Установите настройки для выполнения генерации привязок.

Способы установки настроек для генерации привязок:

- Передайте параметры.
 - Пакет для генерации привязки.
 - Название сущности.
 - Перечень идентификаторов для привязки.
- Передайте фильтр.

Ниже представлен пример вызова автоматической привязки данных из front-end части с помощью параметров.

Пример вызова автоматической привязки данных из front-end части с помощью парамет..

```
ServiceHelper.callService("SchemaDataBindingService",
    "GenerateBindings",
    callback,
    {
        "schemaName": this.entitySchemaName,
        "sysPackageUid": sysPackageUid
        "recordIds": ["...", "..."]
    },
    this
);
```

Ниже представлен пример вызова автоматической привязки данных из front-end части с помощью фильтра.

Пример вызова автоматической привязки данных из front-end части с помощью фильтра

```
ServiceHelper.callService("SchemaDataBindingService",
    "GenerateBindingsByFilter",
    callback,
    {
        "schemaName": this.entitySchemaName,
        "sysPackageUid": sysPackageUid
        "filterConfig": ...
    },
    this
);
```

Важно. Автоматическая привязка данных, вызванная из front-end части, запускается в фоновом режиме. По выполнению привязки данных на коммуникационной панели отобразится уведомление, которое будет содержать информацию о количестве успешно и неуспешно привязанных записях продуктов.

Вызвать автоматическую привязку данных из back-end части

1. Создайте экземпляр класса `SchemaDataBindingController`.
2. Вызовите метод `GenerateBindings`, в который передайте:
 - Идентификатор пакета.
 - Название сущности.
 - Перечень идентификаторов для привязки.

Ниже представлен пример вызова автоматической привязки данных из back-end части.

Пример вызова автоматической привязки данных из back-end части

```
var dataBindingController = ClassFactory.Get<IDataBindingController>(
    new ConstructorArgument("userConnection", UserConnection));
var result = dataBindingController.GenerateBindings(schemaName, recordIds, sysPackageUid);
```