

Описание подсистемы управления сущностями

Подсистема управления сущностями направлена на работу с сущностями: добавление, удаление, изменение, установка связей.

1. Доступ к определенной системе осуществляется на базе сущностей.

Архитектура системы – модульная. Модули могут динамически добавляться и/или обновляться. Каждый модуль может поставлять в ядро описание своего набора сущностей, с которыми он работает, и набор связей между ними, а для хранения самих объектов, модули имеют свои БД.

2. Ядро составляет, хранит и обслуживает схему связей и зависимостей между всеми зарегистрированными сущностями в ИНКА 4.0

В ядро поставляются только описания сущностей, за хранение данных отвечает функциональные модули

Определение сущности

Каждая сущность описывается следующими свойствами.

1. Имя сущности

Имя сущности уникально в пределах одной экосистемы.

Имя сущности состоит из неймспейса и имени, разделенных точкой. Неймспейс сущности совпадает с именем модуля. Примеры:

- *склад.наличие_на_складе*
- *склад.поступление_на_склад*
- *планирование.задание_на_стан_хп*
- *нси.размер_обрези*

2. Поля сущности

Каждое поле имеет название, тип данных

Поле может быть помечено как:

РК(Primary Key)-идентификатор сущности в экосистеме, уникальное значение в пределах одной экосистемы. Предлагается использовать внешний идентификатор сущности из области бизнес логики, а не внутренний Id в базе данных модуля. Например, номер заказа, номенклатурный номер изделия и т.д.

FK(Foreign Key)-ссылка на другую сущность в экосистеме

Помимо простых типов, поле может иметь тип ранее описанной сущности, обеспечивая возможность создания вложенных сущностей.

Функционал подсистемы управления сущностями

1. Добавление и изменение сущностей

В сущности, могут динамически добавляться новые поля, в последствии эти новые поля могут быть изменены или удалены.

Изменение может происходить двумя способами:

- При обновлении версии модуля. Разработчики модуля добавляют/изменяют поля сущностей и при обновлении модуля изменения будут зарегистрированы в системе.
- Динамическое изменение аналитиками или администраторами системы. Динамически могут быть изменены только сущности, помеченные как расширяемые (Extensionable). Новые поля могут добавляться к существующим, изменяться или удаляться. Все поля, изначально поступающие при регистрации модуля, считаются фиксированными и динамически изменяться не могут т.к. код модуля зависит от них.

Техническая часть работы по хранению новых полей ложится на модули, которым принадлежит сущность. Графический интерфейс по изменению сущностей также предоставляет модуль.

2. Предоставление данных

Запросы к данным могут приходиться из нескольких источников:

- Функциональные модули. Функциональные модули могут запрашивать необходимые сущности для выполнения своих функций
- Графический интерфейс. Элементы графического интерфейса могут запрашивать данные для отображения в виде, например, таблиц.
- Бизнес-процессы. Во время выполнения бизнес процессоры также могут быть запрошены данные и сохранены во временное хранилище для последующего изъятия другими задачами бизнес-процесса.
- Администраторы системы. Администраторы системы могут запрашивать данные для анализа и экспорта.

3. Проверка целостности схемы и контрактов

Для обеспечения целостности данных применяются следующие механизмы:

- Бизнес-процессы включают в себя действия по обеспечению целостности данных, пример: модуль управления заказами не сможет удалить заказа, пока не получит подтверждение от модуля склада.
- Периодический запуск тестов, проверяющих целостность данных в системе.
- При динамическом изменении сущностей необходимо обеспечить целостность схемы данных. Для этого, при каждом изменении сущности модуль проходит процедуру перерегистрации, во время которой будут выполнены все тесты на соответствие схеме.

4. Управление и редактирование карты связей.

Функционал система позволяет выстраивать карты связей:

- по ключу (primary key)
- импорт-экспорт (для этого у сущностей есть дополнительно свойства: флаг экспорта и флаг импорта)

Флаг экспорта (Export) показывает, что сущность принадлежит модулю, он ответственен за ее хранение.

Флаг импорта (Import) показывает, что сущность не принадлежит модулю, он не хранит ее в своей базе. Такие сущности нужны для выполнения функций модуля. Модуль рассчитывает на получение этих сущностей от других модулей с помощью ядра. У импортируемых сущностей также должны быть описаны поля, с которыми модуль будет работать. Это необходимо для обеспечения целостности схемы данных. Если ядро не найдет в системе сущность с указанными полями будет выдано сообщение о проблеме.

4. Контроль схемы данных

Система управления сущностями поддерживает и проверяет данную схему при добавлении и изменении сущностей.

Т.к. система позволяет изменять и добавлять поля к сущностям, то совпадение имен сущностей не означает полное совпадение структуры (полей) сущностей. Может оказаться, что разные модули описывают одну сущность, но с разными полями.

Пример:

Модуль "Планирование" версии 1.0 требует наличие сущности "склад.наличие_на_складе" с полями: "сплав", "толщина". Эту сущность поставляет другой модуль - Склад 1.0 В новой версии модуля планирования 1.2 требуется, чтобы сущность

"склад.наличие_на_складе" имела дополнительное поле "ширина". Однако, модуль Склад 1.0 не обновлен и не предоставляет новое поле "ширина" в систему.

Ядро по идентификатору сущности "склад.наличие_на_складе" определяет зависимость между двумя модулями, однако, чтобы обеспечить полное соответствие необходимо проверить соответствие структуры сущностей. Для этого после каждого изменения сущностей ядро производит тестирования схемы сущностей, по результатам которого будет выдано сообщение об ошибке "Несоответствие структур сущностей между модулем Планирование 1.2 и модулем Склад 1.0"

5. Кроме перечисленного функционала, подсистема управления сущностями предоставляет возможность:

- Использования сущностей в конструкторе интерфейсов
- Использования сущностей в конструкторе бизнес-процессов
- Поиска по зарегистрированным сущностям в системе