# Модуль генерации тестовых множеств доменных объектов.

## Общая информация

Модуль генерации тестовых множеств доменных объектов представляет собой отдельный Java-bean. Команды модулю отправляются удаленно с помощью примитивного Java-приложения. Это приложение осуществляет чтение данных из xml-файла, содержащего инструкции по созданию тестовых множеств, получает удаленный доступ к bean’у на сервере jboss с помощью jndi-механизма и вызывает его рабочий метод, передавая в качестве аргумента содержимое xml. Модуль осуществляет разбор инструкций и, в случае отсутствия в них ошибок, создает множество (или множества) объектов с указанными характеристиками, используя вызовы интерфейсных методов ядра системы. Модуль возвращает краткий отчет о времени выполнения и успешно созданных объектах и, в случае ошибок в процессе выполнения, информацию об ошибках.

## Краткое описание функционала

Предполагается, что модуль будет выполнять следующие действия:

создание указанного числа (точного или произвольного из диапазона) доменных объектов заданного типа и заполнение их полей значениями (также точными или из диапазона) в соответствии с заданными правилами;

объявление глобальных правил, применяемых, когда для конкретного множества соответствующего правила явно не задано.

Для ссылочных полей:

создание объекта требуемого типа на лету с последующим заполнением поля ссылкой на него;

создание объекта по «обратной» ссылке (объект создается и в одном из его полей сохраняется ссылка на текущий);

создание иерархии однотипных объектов заданной глубины с возможностью определить шаблон заполнения только один раз.

## Генерация случайных значений.

Для разных случаев генерируются случайные величины с разными распределениями. Для полей типа **long**, **decimal** и **dateTime**, а также для длин строковых полей используется равномерное распределение, полностью задаваемое диапазоном значений (maxValue, minValue и maxLength, minLenght). Для полей типа **string** используется непараметризируемый генератор, выдающий псевдотекст (псевдослова из случайных букв алфавита и числа, разделенные пробелами и знаками препинания). Распределение равномерное с учетом ограничений. Для полей типа **children**, значений глубины однотипных деревьев и количества создаваемых документов используется биномиальное распределение, задаваемое диапазоном и средним значением.

## Структура xml-инструкций, используемых для управления.

**template** – базовый элемент, определяет параметры заполнения для всех множеств. **template** допускает единственный вложенный элемент – **object**.

**set** – базовый элемент, инструкция создать множество объектов.

Атрибуты:

* id – идентификатор множества. Используется, когда объекты другого множества должны ссылаться на объекты этого множества.
* quantity – количество объектов, которое нужно создать. Используется, когда необходимо указать точное количество объектов.
* maxQuantity, minQuantity – максимальное и минимальное количество объектов, которое требуется создать. Будет создано случайное количество объектов из диапазона. Обязательны, если quantity отсутствует.
* averageQuantity – математическое ожидание количества объектов. При достаточно большом количестве вызовов среднее значение будет близко к МО. По умолчанию равно (maxQuantity – minQuantity)/2, должно быть значением между ними.

**set** допускает единственный вложенный элемент – **object**.

**object**, в случае вложения в **set,** определяет параметры заполнения полей для элементов данного множества, а в случае вложения в **template** для элементов всех множеств вообще (глобальное). Правила заполнения поля, определенного во вложенном элементе, имеют для своего множества больший приоритет, чем глобально объявленные. Если для соответствующего поля во множестве правило не определено, то используется глобальное.

**object** имеет единственный атрибут:

* type – тип объекта, должен быть одним из типов, уже зарегистрированных в системе. Обязателен, если **object** входит в **set**. Если объявлен для **object** вне **set**, то параметры заполнения будут задаваться для всех множеств данного типа.

**object** допускает следующие вложенные элементы:

**string** – строковое поле. Поддерживает следующие атрибуты:

* name – имя поля. Имя поля должно совпадать с именем одного из полей типа указанного в атрибуте domain-object-type элемента **set**. Если имя не указано, правило заполнения будет применяться ко всем строковым полям множества (или группы множеств), для которых явно не заданы правила. Глобальное правило для *конкретного* поля (с именем) имеет больший приоритет, чем локальное правило для типа полей.
* length, maxLength, minLength – по аналогии с quantity, maxQuantity, minQuantity – точная длина строки, которая будет присвоена в качестве значения, либо диапазон допустимых длин. Длина в любом случае ограничивается максимальной длиной зарегистрированного поля.
* value – фиксированное значение, которое будет присвоено всем объектам множества (атрибуты ограничения длины будут проигнорированы).
* noValueChance – дробь от 0 до 1, определяет вероятность того, что значение не будет заполнено вообще.

**dateTime** – поле, содержащее дату и время. Поддерживает атрибуты:

* name – аналогично **string**.
* value – аналогично **string**, если значение не удастся распознать как дату, оно будет проигнорировано. Это относится ко всем атрибутам, ограничивающим значение не-**string** полей.
* maxValue, minValue – аналогично **string**.
* relativeValue – точная дата, вычисляется как смещение относительно текущей.
* relativeMaxValue, relativeMinValue – максимальное и минимальное допустимые смещения относительно текущей даты.
* noValueChance – аналогично **string**

**long** – поле, содержащее целое число. Поддерживает атрибуты name, value, minValue, maxValue, noValueChance.

**decimal** – поле, содержащее десятичную дробь с фиксированным количеством знаков до и после запятой. Поддерживает атрибуты name, value, minValue, maxValue, noValueChance.

**reference** – поле, содержащее ссылку на другой доменный объект системы в виде его уникального идентификатора. Поддерживает атрибуты:

* name – аналогично **string**.
* type – тип объекта, должен быть одним из типов, уже зарегистрированных в системе. В качестве значения поля будет присвоен идентификатор произвольного объекта данного типа.
* setId – идентификатор уже созданного множества. В качестве значения поля будет присвоен идентификатор произвольного объекта данного множества.
* parentReference – булево значение (по умолчанию – false), определяет, должно ли поле содержать родительский объект (см. **children**).
* treeReference – булево значение (по умолчанию – false), определяет, должно ли поле использоваться для ссылки на следующий объект в цепи (см. treeDepth).
* treeDepth – глубина дерева. Если reference содержит этот атрибут, будет создана цепь объектов длиной n, где n – значение атрибута. Предыдущий объект будет ссылаться на следующий посредством поля **reference**, в котором объявлен этот атрибут.
* minTreeDepth, maxTreeDepth – максимальное и минимальное допустимые возможной глубины дерева. Аналогичны другим атрибутам диапазона.
* averageTreeDepth – математическое ожидание глубины дерева (аналогично averageQuantity).

**reference** может содержать вложенный элемент object. В этом случае будет создан объект данного типа, а ссылка на него помещена в reference.

**children** – псевдополе. В результате обработки этого элемента будет создан доменный объект, ссылающийся на текущий. Поддерживает атрибуты:

* quantity – количество объектов, которое нужно создать. Используется, когда необходимо указать точное количество объектов.
* maxQuantity, minQuantity – максимальное и минимальное количество объектов, которое требуется создать. Будет создано случайное количество объектов из диапазона. Обязательны, если quantity отсутствует.
* averageQuantity – МО, аналогично одноименному атрибуту **set**.
* treeReference – булево значение (по умолчанию – false), определяет, должно ли поле использоваться для ссылки на предыдущий объект в цепи (см. treeDepth).
* treeDepth – глубина дерева. Если reference содержит этот атрибут, будет создано дерево объектов глубиной n, где n – значение атрибута. Следующий объект будет ссылаться на предыдущий посредством поля reference, в котором объявлен этот атрибут.
* minTreeDepth, maxTreeDepth – максимальное и минимальное допустимые возможной глубины дерева. Аналогичны другим атрибутам диапазона.

**children** может содержать вложенный элемент **object**. В этом случае будет создан объект данного типа, а ссылка на него помещена в reference. Во вложенном **object** должен присутствовать один элемент **reference** с атрибутом parentReference, установленным в true.

## Пример xml с командами создания объектов.

<template>

<object>

<string name="Name" minLength="4" maxLength="32"/>

</object>

</template>

<template>

<object type="Outcoming\_Document">

<reference name="Attachment">

<object type="Attachment>

<string name="Attachment\_path" minLength="10" maxLength="100"/>

</object>

</reference>

</object>

</template>

<set quantity="10000" id="genericIncoming">

<object type="Incoming\_Document">

<string length="3"/>

<string name="Registration\_Number" length="10"/>

<string name="Description" minLength="0" maxLength="20"/>

<dateTime name="Registration\_Date" minValue="01.01.2010" maxValue="31.12.2013"/>

<reference name="Author" type="Employee" />

<reference name="Attachment">

<object type="Attachment>

<string name="Attachment\_path" minLength="10" maxLength="100"/>

</object>

</reference>

<children minQuantity="6" maxQuantity="8">

<object type="Visa">

<string name="Attachment\_path" minLength="10" maxLength="100"/>

<reference name="parent" parentReference="true"/>

</object>

</children>

<children minQuantity="0" maxQuantity="3" minTreeDepth="2" maxTreeDepth="3"/>

<long name="Long\_Field" value="0"/>

<long name="Another\_Long\_Field" minValue="0" maxValue="777.777"/>

<decimal name="something" value="88" />

<decimal name="Decimal\_Field" minValue="0" maxValue="777.777" />

</object>

<set>