## Общие положения

Для импорта данных реализована подсистема загрузки данных из CSV файла.

Подсистема работает в 2-х режимах:

1. Загрузка данных с помощью remote вызова метода сервиса ImportDataService.
2. Автоматическая загрузка данных при старте сервера.

### Загрузка данных с помощью remote вызова метода сервиса ImportDataService.

Для загрузки данных необходимо вызвать метод importData сервиса ImportDataService и передать методу как параметр массив байт зачитанного CSV файла. Интерфейс сервиса загрузки данных:

**package** ru.intertrust.cm.core.business.api;

/\*\*

\* Сервис загрузки данных

\* **@author** larin

\*

\*/

**public** **interface** ImportDataService {

**public** **static** **final** String *TYPE\_NAME* = "TYPE\_NAME";

**public** **static** **final** String *KEYS* = "KEYS";

/\*\*

\* Удаленный интерфейс

\* **@author** larin

\*

\*/

**public** **interface** Remote **extends** ImportDataService{

}

/\*\*

\* Метод загрузки данных из файла.

\* **@param** loadFileAsByteArray зачитанный массив данных из файла

\*/

**void** importData(**byte**[] importFileAsByteArray);

}

### Автоматическая загрузка данных при старте сервера

При старте сервера производится поиск всех файлов ru/intertrust/cm/core/importdata/import.xml. Файлы могут быть расположены в разных архивах. В файлах import.xml содержатся имена файлов csv для загрузки и их зависимость от других файлов csv для обеспечения корректной очередности загрузки файлов. Зависимость учитывается иерархически. Так же производится проверка на зацикленность зависимостей, в этом случае формируется исключение. Файлы csv должны так же лежать в директории ru/intertrust/cm/core/importdata. Формат файла import.xml:

<?xml version=*"1.1"* encoding=*"UTF-8"* standalone=*"yes"*?>

<ImportData>

<File name=*"import-organization.csv"*>

<Depend>status.csv</Depend>

</File>

<File name=*"import-department.csv"*>

<Depend>status.csv</Depend>

<Depend>import-organization.csv</Depend>

</File>

<File name=*"import-employee.csv"*>

<Depend>status.csv</Depend>

<Depend>import-department.csv</Depend>

</File>

<File name=*"set-organization-boss.csv"*>

<Depend>status.csv</Depend>

<Depend>import-employee.csv</Depend>

</File>

<File name=*"set-department-boss.csv"*>

<Depend>status.csv</Depend>

<Depend>import-employee.csv</Depend>

</File>

</ImportData>

### Формат файла импорта данных

Файл CSV должен сохранен в кодировке ANSI-1251. Сделано это для совместимости с редактором Excel. Две первые строки файла отведены под метаинформацию. В первой строке задается имя типа для импорта и ключевые поля в виде:

TYPE\_NAME=Department;KEYS=Name,Number;

Во второй строке хранятся имена полей для импорта:

Name;Organization;ParentDepartment

Начиная с третьей строки идут непосредственно данные для импорта

Подразделение 1;"name=""Организация 1""";

Подразделение 2;"name=""Организация 1""";

Подразделение 3;"name=""Организация 1""";

В полях типа Reference необходимо прописывать выражение, для получение значения ссылки на другой доменный объект. Выражения могут быть трех видов:

1. field\_name=”field\_value”. Данное выражение означает, что надо получит конфигурацию поля типа Reference из конфигурации доменного объекта. Далее у поля типа Reference получается тип доменного объекта на который ссылается данное поле, изо всех доменных объектов полученного типа делается запрос с условием field\_name=”field\_value” и из полученного множества берется первый результат.
2. type\_name.field\_name=”field\_value”. Данное выражение означает, что изо всех доменных объектов типа type\_name делается запрос с условием field\_name=”field\_value” и из полученного множества берется первый результат. Данная конструкция используется когда поле Reference может ссылаться на несколько типов доменных объектов.
3. Запрос на внутреннем sql языке. Запрос должен возвращать поле типа Reference. Например: select t.id from department t where t.name='Подразделение 2'.