

Система МЕТА

Введение

<https://sirius-systems.ru/documentation.html> - очень полезная информация о мете, пожалуйста, прочитайте.

Понятие "мета" используется для описания нескольких вещей: а) наша основная библиотека доступа к данным - lib_meta.jar; б) родной взб-интерфейс доступа к данным (без реализации бизнес-логики); в) подход к представлению, хранению, доступу к данным.

Данные лежат в множестве реляционных баз, библиотека lib_meta.jar (классы пакета arpt.meta3.*) обеспечивает доступ к ним в виде объектов с атрибутами. Каждый объект имеет уникальный идентификатор, сквозной для всех баз, номер типа, дату/время создания, смерти и автора объекта.

- Тип объекта - это число от 1 до 999, определяет набор атрибутов для всех объектов типа, параметры прав доступа к объектам, а так же расположение объектов типа в множестве реляционных баз.

- Уникальный идентификатор объекта (ID объекта) может быть 12- или 18-значным, id объекта однозначно определяет тип объекта (первые четыре знака минус 1000).

- Атрибуты объектов бывают 4 типов: строковый, численный, дата со временем, ссылка на другой объект. Для обращения к атрибутам объекта используется ID атрибута типа, он уникален в контексте типа, id атрибута типа имеет длину 10 знаков, id однозначно определяет номер типа, атрибутом объектов которого он является (первые четыре знака минус 1000), и тип атрибута (пятый знак). С атрибутами каждого типа лучше знакомиться на конкретных примерах объектов.

Все параметры для работы с базами и параметры проекта мета (и библиотека, и интерфейс) берет из главного конфигурационного property файла, который разворачивается в памяти явамашины через объекты класса

ResourceBundle. ResourceBundle является параметром всех вызовов api библиотеки lib_meta.

ResourceBundle mains = ResourceBundle.getBundle(mainResourceFile, new Locale("ru", "RU")); - получение ресурсбандла

mainResourceFile - это файл intern_main, но имя проперти файла может отличаться в зависимости от проекта и того, как он попадает в код.

Поля объекта класса Obb:

Obb ob= new Obb(ob.id_type); - тестовый объект определенного типа ob.id_type

ob.id - это идентификатор объекта,

ob.id_type - это тип объекта,

ob.id_user - это пользователь, который создал объект,

ob.data_n- это момент времени, когда объект создали,

ob.data_k - это дата, по наступлении которой объект логически перестанет существовать, если она 31.12.4172 - объект никто не удалял,

ob.zn - это HashMap, в котором лежат все заполненные атрибуты объекта).

Важная информация о хранении данных в базах меты - все операции редактирования, удаления - логические, то есть из базы никакие данные не удаляются, а лишь считаются "живыми" в определенный промежуток времени от data_n до data_k. Это справедливо как для объектов целиком, так и для отдельных атрибутов объекта.

1000127 Название англ:	ГАГРА ЛАЗУРНАЯ, гостевой дом Новый [22.11.2019 16:03:13] 3172884:I.KOTKOVA [22.11.2019 16:04:30] 0: 191122160313610001 ГАГРА ЛАЗУРНАЯ, гостевой дом [22.11.2019 16:04:30] 3172884:I.KOTKOVA [31.12.4712 00:00:00] 0: 191122160430750003
---	---

Как вы видите, в правом столбце (это значение атрибута) написано 3 строки:

1. это текущее название,
2. это изначальное название, которое было с 22.11.2019 16:03:13 по 22.11.2019 16:04:30:

Новый - установленное значение,

[22.11.2019 16:03:13] - дата, когда значение стало таким,

3172884 : I.KOTKOVA- это id_user : логин пользователя, кто написал такое значение

[22.11.2019 16:04:30] - дата, когда значение перестало быть таким,

0: - это id_user : логин пользователя, кто поменял изначальное значение, но здесь написано 0, потому что в коде не прописано сохранение пользователя,

который поменял атрибут - это плохо,

191122160313610001 - это id значения атрибута.

3. это новое значение атрибута, которое заменило предыдущее 22.11.2019 в 16:04:30:

ГАГРА ЛАЗУРНАЯ, гостевой дом - установленное новое значение,

[22.11.2019 16:04:30] - дата, когда значение стало таким,

3172884 : I.KOTKOVA- это id_user : логин пользователя, кто написал такое значение - хотя во второй строчке не было видно, кто заменил старое значение, зато в этой строчке видно и можно однозначно понять, кто исправил значение,

[31.12.4712 00:00:00] - дата, когда значение перестанет быть таким, по умолчанию для актуальной записи это дата будет равна 31.12.4712 00:00:00,

0: - это id_user : логин пользователя, кто поменял значение, но здесь написано 0, потому что это текущее значение и его никто не менял,

191122160430750003 - это id значения атрибута.

API для работы с объектами.

Получить из базы объект можно двумя способами:

- Получение конкретного объекта по его уникальному идентификатору при помощи метода `Ob0.getOb`.

```
Obb obHotel = Ob0.getOb(mains, id, null, true);
```

id - id объекта, который достаем из базы,

null - это значение `datafis` - то есть дата, на которую хотим достать объект, так как объект меняется со временем, то в зависимости от `datafis` значения атрибутов у объекта может измениться и мы можем достать значение на определенную секунду, если null - то `datafis` = сейчас,

true - это значение `isOb`, то есть доставать ли поля типа `.data_n`, `.data_k` - если true, значит доставать.

- Поиск объекта/объектов по известному идентификатору атрибута типа и его значению при помощи методов `Ob0.addCondition` и `Ob0.getSrc`.

//Чтобы найти объекты по известным значениям атрибутов необходимо создать в памяти объект `appt.meta3.Obb`, в котором будут указано, на какие атрибуты и какие условия будут накладываться. Создаем объект `filter`

```
Obb filter = Ob0.createFilter(26);
```

//задаем условие, что значение атрибута `1000117 = flt2`, поскольку система БГ разрабатывается уже очень много лет, то в объектах могут еще существовать старые id атрибутов, которые были короткими

```
Ob0.addCondition(filter, 1000117, Ob0.ComparisonType.EQ, flt2);
```

//задаем условие, что значение атрибута 1026100353 = Да или 1026100353 = Нет

```
Ob0.addCondition(filter, 1026100353, Ob0.ComparisonType.IN, "Да#Нет");
```

//достаем все объекты из базы, которые удовлетворяют условиям

```
Obb[] hotelObs = Ob0.getSrcObs(mains, connname, filter, 0, 0);
```

connname - имя соединения с БД (имя коннекшна к БД), по-умолчанию

Ob0.metaconnname

filter - наш фильтр

первый 0 - это параметр m, отвечает за то, начиная с какого объекта мы начинаем доставать их из базы, нумерация с 0

второй 0 - это параметр entr, количество записей, которые достаем, если 0 - значит все

Пример: нам нужно достать 1000 объектов по 50 штук на странице, для первой страницы вызов getSrc мы сделаем с параметрами m=0 и entr=50, для второй страницы m=50, entr=50 и т.д, а для последней страницы m=950, entr=50.

В примере с фильтром приведено всего 2 условия выборки:

EQ - значение атрибута равно значению справа от условия,

IN - значение атрибута равно одному из значений справа от условия, разделенных решеткой,

Но есть и другие:

EQD - значение атрибута (дата dd.MM.yyyy) равно значению даты справа от условия,

EQDT - значение атрибута (дата со временем dd.MM.yyyy HH:mm:ss) равно значению даты со временем справа от условия,

GT - значение атрибута больше значения справа от условия,

GTD - значение атрибута(дата dd.MM.yyyy) больше значения даты справа от условия,

GTDT - значение атрибута(дата со временем dd.MM.yyyy HH:mm:ss) больше значения даты со временем справа от условия,

GE - значение атрибута больше или равно значению справа от условия,

GED - значение атрибута(дата dd.MM.yyyy) больше или равно значению даты справа от условия,

GEDT - значение атрибута(дата со временем dd.MM.yyyy HH:mm:ss) больше или равно значению даты со временем справа от условия,

LT - значение атрибута меньше значения справа от условия,

LTD - значение атрибута(дата dd.MM.yyyy) меньше значения даты справа от условия,

LTDT - значение атрибута(дата со временем dd.MM.yyyy HH:mm:ss) меньше значения даты со временем справа от условия,

LE - значение атрибута меньше или равно значению справа от условия,

LED - значение атрибута(дата dd.MM.yyyy) меньше или равно значению даты справа от условия,

LEDT - значение атрибута(дата со временем dd.MM.yyyy HH:mm:ss) меньше или равно значению даты со временем справа от условия,

Список может пополняться в процессе работы с библиотекой.

Получив объекты из базы, с ними можно производить какие-либо действия, например, редактировать значения атрибутов, удалять значения атрибутов, удалять сами объекты.

Достаем атрибуты объекта:

//достаем объект из базы

Obb obHotel = Ob0.getOb(**mains**, **id**);

//достаем значение атрибута 1026423327 у объекта obHotel, вне зависимости от типа атрибута - значение атрибута api всегда вернет строкой

String hotelInfo = Ob0.getAt(obHotel, **1026423327**);

System.**out**.println("**hotelInfo**="+hotelInfo);

//достаем значение множественное атрибута 1026101221 у объекта obHotel, всегда возвращается массив строк

String[] hotelVideos = Ob0.getAts(obHotel, **1026101221**);

System.**out**.println("**hotelVideos**="+Arrays.toString(hotelVideos));

Результат:

hotelInfo=Заезд в 12.00

hotelVideos=[/video/margadina.flv, /video/MARGADINA3.flv]

Очень важно понимать, когда мы достаем объект из базы, мы кладем его копию из базы в виде объекта appt.meta3.Obb в память явамашины, когда же мы достаем атрибуты при помощи getAt, getAts мы работаем уже с копией объекта в памяти, не делаем обращений к базе.

Редактируем атрибуты объекта:

//достаем объект из базы

```

Obb obHotel = Ob0.getOb(mains, id, null, true);
//достаем значение атрибута 1026423327 у объекта obHotel
String hotelInfo = Ob0.getAt(obHotel, 1026423327);
System.out.println("hotelInfo="+hotelInfo);
//в нашем объекте, меняем значение атрибута (еще раз: это копия объекта из базы)
Ob0.addAt(obHotel, 1026423327, "ОТЕЛЬ ЗАКРЫТ НА КАРАНТИН!");
//в объект записываем, кто его изменил
obHotel.id_user = id_user;
//сохраняем изменения, вызывая метод редактирования объекта
Ob0.edtOb(mains, obHotel);
//снова достаем тот же объект
Obb obHotelNew = Ob0.getOb(mains, id, null, true);
//достаем новое значение атрибута 1026423327 у объекта obHotel
String hotelInfoNew = Ob0.getAt(obHotel, 1026423327);
System.out.println("hotelInfoNew="+hotelInfoNew);

```

Результат

hotelInfo=Check in (время заезда) после 14:00, Check out (время выезда) до 12:00

hotelInfoNew=ОТЕЛЬ ЗАКРЫТ НА КАРАНТИН!

Результаты работы кода отображаются и в интерфейсе меты:

1026423327 Расчетный час и важная информация для вачера:	ОТЕЛЬ ЗАКРЫТ НА КАРАНТИН! Check in (время заезда) после 14:00, Check out (время выезда) до 12:00 [22.11.2019 16:04:30] 3172884:I.KOTKOVA [22.05.2020 00:36:42] 2891176:E.DENISOVA 191122160430750002 ОТЕЛЬ ЗАКРЫТ НА КАРАНТИН! [22.05.2020 00:36:42] 2891176:E.DENISOVA [31.12.4712 00:00:00] 0: 200522003642080001
--	--

Текущее значение стало: ОТЕЛЬ ЗАКРЫТ НА КАРАНТИН!

Видно, что установленное, пользователем 3172884, значение «Check in (время заезда) после 14:00, Check out (время выезда) до 12:00» перестало быть действительным 22.05.2020 в 00:36:42, новое значение было установлено пользователем 2891176.

Добавляем новый объект.

Категорически нельзя создавать объект без единого значения атрибута!

```

//создаем новый объект 26 типа
Obb obHotel = new Obb(26);
//записываем, кто создал объект
obHotel.id_user = id_user;
//добавляем значения атрибутов
Ob0.addAt(obHotel, 1026423327, "Заезд в 15.00");
Ob0.addAt(obHotel, 1000127, "ОТЕЛЬ ДЛЯ ТЕСТА");

```

```
String[] videos = new String[2];
videos[0] = "/video/margadina.flv";
videos[1] = "/video/MARGADINA3.flv";
//добавляем в значение атрибута 1026101221 массив
Ob0.addAtr(obHotel, 1026101221, videos);
//кладем объект в базу, addOb возвращает id созданного объекта
String idOb = Ob0.addOb(mains, obHotel);
System.out.println("idOb="+idOb);
```

Результат:

idOb=102610117613

В итоге создался объект с заполненными атрибутами:

Название, расчетный час и видеоролик. В Видеоролике лежит массив с адресами видео. Объект был создан 25.05.2020 00:56:29 пользователем с логином E.DENISOVA - это прописано в интерфейсе меты.

1000127 Название англ:	Отель для теста
1026423327 Расчетный час и важная информация для ваучера:	Заезд в 15.00
1026101221 Видеоролик:	/video/margadina.flv /video/MARGADINA3.flv

Создан 25.05.2020 00:56:29 E.DENISOVA

Удаляем объект.

```
//достанем объект из базы и проверим, что он действительно существует
Obb obHotel = Ob0.getOb(mains, idHotel);
System.out.println("obHotel="+obHotel.id);
//удаляем объект по id, и передаем того, кто удалил
Ob0.delOb(mains, idHotel, user_id);
//проверяем, что объект удалился
Obb obHotelNew = Ob0.getOb(mains, idHotel);
System.out.println("obHotelNew="+obHotelNew);
```

Результат

obHotel=102610117616

obHotelNew=null

В итоге объект логически удалился:

1000127	Название англ:	
---------	----------------	--

1026423327	Расчетный час и важная информация для ваучера:	
------------	--	--

1026101221	Видеоролик:	
------------	-------------	--

OB IS NULL

Но по истории видно, кто и когда создал и удалил атрибуты. Видно, что удалили все значения атрибутов, потому что дата конца действия значения атрибута это 25.05.2020 01:43:10 и новых значений после этого не было. Объект не может существовать без заполненных атрибутов. Тот, кто удалил последний заполненный атрибут - тот удалил и объект.

1000127	Название англ:	Отель для теста [25.05.2020 00:18:31] 0: [25.05.2020 01:43:10] 2891176:E.DENISOVA 200525001831080004
---------	----------------	--

1026423327	Расчетный час и важная информация для ваучера:	Заезд в 15.00 [25.05.2020 00:18:31] 0: [25.05.2020 01:43:10] 2891176:E.DENISOVA 200525001831080001
------------	--	--

1026101221	Видеоролик:	/video/MARGADINA3.flv [25.05.2020 00:18:31] 0: [25.05.2020 01:43:10] 2891176:E.DENISOVA 200525001831080002 /video/margadina.flv [25.05.2020 00:18:31] 0: [25.05.2020 01:43:10] 2891176:E.DENISOVA 200525001831080003
------------	-------------	---

Задачи

	Название	Английское название	N	Тип
1	Описание	description	1990100063	Строка
2	Площадь	space	1990210007	Число
3	Название	name	1990410000	Строка
4	E-mail	email	1990410010	Строка
5	GPS координаты (x,y)	gps	1990410087	Строка
6	Максимальная вместимость	max_capacity	1990310131	Целое число
7	Адрес	address	1990410028	Строка
8	Город	town_from_hotel	1990100059	Ссылка
9	Категория	category	1990100061	Список
10	Описание отеля	hotel_description	1990922737	Ссылка

Во всех выводах, которые вы будете делать - слева должна быть нумерация. Все действия с объектами и атрибутами должны происходить под вашим логином.

1. Создать 10 объектов 990 типа. Заполнить все(кроме описания отеля) атрибуты значениями на ваше усмотрение. Сохранить у себя все id отелей. Города:
100510000863 - Москва
100539140353 - Сочи Имеретинский курорт
100510397198 - Воронеж
100510000892 - Екатеринбург
100510397333 - Саратов
100510000826 - Самара
Категории 1* - 5*
2. Достать все отели, у которых категория 5*. И площадь больше 100. Вывести в консоль все объекты с названием отеля, рейтингом отеля, и все контактные данные по отелю.
3. Достать все объекты, у которых вместимость больше 50, но меньше 150. Вывести id отеля, название, вместимость и координаты.

4. Достать все объекты, у которых заполнен атрибут e-mail, вывести в консоль «постранично», на каждой странице по 3 отеля, номер страницы считывать из аргумента.
5. Во всех своих отелях поменять название отеля на свое имя и фамилию, почту на свой email.
6. Достать все объекты, которые удовлетворяют условию из предыдущего пункта.
7. Удалить все объекты, которые удовлетворяют условию из п.5.

	Название	Английское название	N	Тип
1	Площадь	kvm	1240100946	Строка
2	Сейф на ресепшн	reception-safe	1240101606	Строка
3	Фен	room-hairdryer	1240422204	Да/Нет
4	Кондиционер	central_conditioner_description	1240422948	Строка
5	На оживленной улице	located_on_main_road	1240422118	Да/Нет
6	Мини-бар	room-mini-bar	1240101793	Строка
7	Балкон	room-balcony	1240422218	Да/Нет
8	Холодильник	room_fridge	1240102011	Да/Нет
9	Душ	shower_description	1240422947	Строка
10	Биде	room-bidet	1240101803	Строка
11	Диетическая кухня	dietary_cuisine	1240422732	Строка
12	Микроволновая печь	room_microwave	1240102063	Да/Нет
13	Боулинг	sports_bowling	1240102035	Целое число

8. Создать 5 объектов описания отелей, со всеми заполненными атрибутами. Сохранить себе id объектов.
9. Создать 5 отелей с названием, и положить в каждый отель по 1 ссылке на описание отеля из предыдущего пункта.

10. Достать все отели, которые ссылаются на ваши описания отелей. Вывести всю информацию об отеле в терминал.
11. Достать все отели, в которых есть холодильник и микроволновка, и которые находятся на НЕ оживленной улице, вывести информацию об отелях в терминал.
12. Поменять во всех своих объектах боулинг на 5, и наличие холодильника сделать - да.
13. Для любых отелей (считываем из аргумента id отеля) - вывести в консоль все описание отеля.
14. Удалить все отели и описания отелей, которые вы создали.
15. Для всех атрибутов (990 и 240 типы), длина которых 10, узнать тип атрибута по id атрибута типа.
16. Достать все страны (4 тип), засечь, сколько потребовалось времени на getSrcObs, посчитать количество стран. Найти все страны, у которых валюта EUR (по всем объектам через if), вывести эти страны на экран(Nº id название). Засечь, сколько по времени потребовалось на getSrcObs + вывод на экран. Достать все страны, у которых валюта EUR, засечь getSrcObs, посчитать количество стран. Вывести страны с EUR на экран (Nº id название). Засечь, сколько по времени потребовалось на getSrcObs + вывод на экран. Вывести на экран затраченное на каждую операцию время. Сделать вывод о затраченном времени.

	Название	Английское название	N	Тип
1	Страна туризма	tour_country	1000606	Да/Нет
2	Валюта**	currency	1000001	Список
3	Шенген	SHENGEN	1004101467	Да/Нет
4	Название	title_rus	1000000	Строка
5	Название англ	title_ing	1000105	Строка
6	Наименьшая широта страны	min_latitude	1004200022	Число
7	Наибольшая широта страны	max_latitude	1004200023	Число
8	Наибольшая долгота страны	max_lougitude	1004200025	Число
9	Наименьшая долгота страны	min_lougitude	1004200024	Число

17. Достать все объекты стран, засечь getSrcObs, посчитать количество объектов. Сложить(+ засечь время) все объекты в HashMap и TreeMap (id, name), вывести(+ засечь время) все объекты на экран. Сравнить порядок отображения объектов. Сделать вывод о затраченном времени.
18. Достать все объекты стран, которые находятся на границе экватора в восточном полушарии. Вывести название и является ли эта страна страной туризма.
19. Достать все города (5 тип) России, сложить(+ засечь) их id в HashSet, LinkedHashSet, TreeSet и вывести(+засечь) все id городов на экран. Сложить(+ засечь) id города + название города в HashMap, LinkedHashMap, TreeMap(+ засечь). Из каждого Set и Map удалить (+ засечь) 300 городов. Вывести на экран сводную таблицу по времени работы каждого Set и каждого Map. Сделать вывод, какой Set и Map эффективнее и в каком случае.

Страна	country_from_town	1000004	Ссылка как список
--------	-------------------	---------	-------------------

20. Заполнить таблицу и внимательно ее изучить. Найти соответствие между типами атрибутов(2 столбец) и типами атрибутов (5 столбец).

Найти повторяющиеся id атрибутов, в чем у них отличия?

id атрибута типа	тип	тип атрибута	id атрибута	тип атрибута
1002922658				Ссылка
1003900009				Ссылка
1003500016				Дата
1003400012				Строка
1003500019				Дата
1304522564				Дата
1023300004				Целое число
1023300005				Целое число
1024223152				Число
1024700003				Дата с временем
1024700004				Дата с временем
1026922588				Ссылка
1026522589				Дата

id атрибута типа	тип	тип атрибута	id атрибута	тип атрибута
1026422636				Строка
1158422562				Файл
1304422562				Файл
1023422562				Файл
1304422563				Файл
1024200049				Число
1393223152				Число

21. Выбрать все НС номера(46 тип), id номера берется из аргумента(102110001662, 102110001660, 102110001440, 102110001823). В файл в своей папке (/w2/srv/bin/ + имя вашей папки) в формате JSON для каждого НСа (тип стоимости НС) вывести: название, id номера, и тариф. В название файла класть id НС, например n1021..

	Название	Английское название	N	Тип
1	Название	title_room_set_cost	1000348	Строка
2	id_номера	id_room_set_cost	1000350	Ссылка
3	Бронировать у партнера	kto_bronir	1046100704	Ссылка
4	Название НС без питания	ns_without_dinner	1046100921	Строка
5	список услуг []	NC_services	1046101058	Строка
6	id по партнеру	id_by_partner	1046101993	Строка
7	Тип стоимости (0/1/2 - С/БНС/НС)	costType	1046222729	Целое число
8	Вид из окна	id_view	1046922839	Ссылка
9	Питание	id_meal	1046922840	Ссылка
10	Тариф проживания	accommodation_rate	1046910245	Ссылка

22. Выбрать все виды касс (89 тип), в которых четный номер. В файл в своей папке (/w2/srv/bin/ + имя вашей папки) в формате XML вывести информацию о кассе: название, номер, название офиса.

Название	Английское название	N	Тип
1 Название	name	1089410000	Строка
2 номер	number	1000633	Строка
3 Офис	link_to_office	1001359	Ссылка

121 тип офиса:

Название	title_rus	1000950	Строка
----------	-----------	---------	--------

23. Написать строку для компиляции проекта в консоли (javac).