



7a35fa818f269890da1440dd39150643d0106017

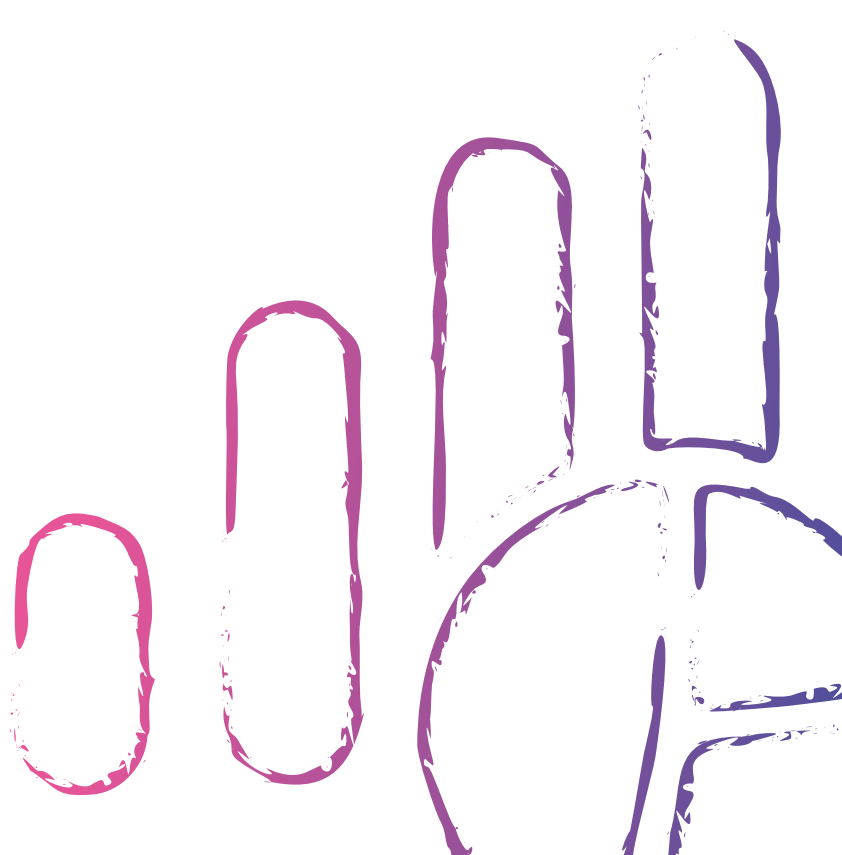


Luxms BI. Руководство по конфигурации дэшлетов

JSON КОНФИГУРАЦИИ

LPE ВЫРАЖЕНИЯ

2024-04-02



Оглавление

1	Расширенные настройки виджетов Luxms BI	1
1.1	Типы дэшей в Luxms BI	1
1.2	JSON конфиг дэша	12
1.2.1	Поле title	12
1.2.2	Поле description	15
1.2.3	Секция frame	15
1.2.4	Поле backgroundImage	16
1.2.5	Секция dataSource	16
1.2.5.1	Опции секции dataSource	17
1.2.6	Секция display	24
1.2.6.1	Опции секции display	25
1.2.6.2	Визуализация тултипов	26
1.2.7	Секция echart	28
1.2.8	Секция forecast	30
1.2.9	Опция url	31
1.2.10	Секция options	32
1.2.10.1	Списки опций, доступные для разных типов дэшей	34
1.2.11	Секция hierarchy	39
1.2.12	Секция onClickDataPoint	39
1.2.13	Секция children	46
1.2.14	Опции для визеля Pie	46
1.2.15	Опции для визеля koob-table-simple (Плоская таблица)	47
1.2.16	Опции для визеля tableP (Таблица с группировкой)	49
1.2.16.1	Редактирование данных через koob-table-simple	51
1.2.17	Конфигурационные опции для визеля “Значение”	54
1.2.18	Конфигурация lookup-таблицы	56
1.2.18.1	Настройка ширины столбцов	56
2	Примеры конфигурации дэшей в LuxmsBI	57
2.1	Поле title	57
2.2	Поле view_class	58
2.3	Секция frame	59
2.4	Секция dataSource	62
2.4.1	Управление данными	62
2.4.2	Стилизация дэшей	70
2.5	Секция display	72
2.5.1	Стилизация типа дэша “Значение”	75
2.5.2	Объект stoplights / массив range	82
2.6	Секция options	89
2.7	Пример конфигурации визеля map	90
2.7.1	Доп. опции в блоке display для слоя “Точки”	94
2.8	Дэш axes-selector и работа с его конфигурационным файлом	95
2.9	Конфигурация визеля “Вкладки” (tabs)	97

2.10	Конфигурация управляющего дэша	101
2.11	Конфигурация дэша what-if	107
3	Использование LPE-выражений для стилизации дэшей	108
3.0.1	Функции, доступные внутри выражений if и when	111
4	Руководство по конфигурации дэшборда	112
5	Таблица изменений	116

1 Расширенные настройки виджетов Luxms BI

Дэшлет (дэш) - зарезервированное место на дэшборде какого-либо из датасетов, являющееся контейнером для конкретного визеля, но не равный ему

Визель - компонент React, по умолчанию рисующий график на основе входных данных и характеристик и особенностей самого типа графика. В одном дэше может быть отрисовано более одного визеля.

Настройки дэша включают в себя тип дэша и JSON структуру, описывающую содержимое дэша.

1.1 Типы дэшей в Luxms BI

Тип дэша (view_class) - это строка, указывающая тип визеля, который нужно отобразить в данном дэшлете. Ниже представлен список типов дэшей, доступных “из коробки”.

1. Аналитические:

1.1 Столбики горизонтальные (bar)

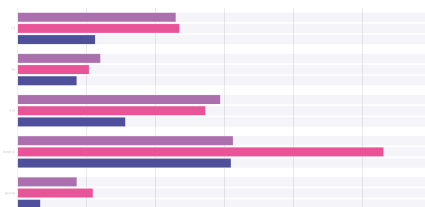


Рис. 1.1 Столбики горизонтальные

Представляет данные в виде горизонтального столбчатого графика.

1.2 Линия (line)

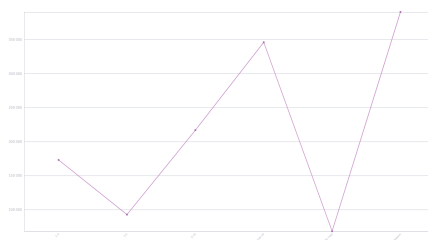


Рис. 1.2 Линия

Представляет данные в виде линейного графика.

1.3 Столбики вертикальные (column)

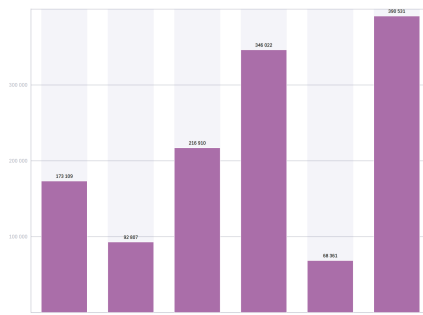


Рис. 1.3 Столбики

Представляет данные в виде вертикального столбчатого графика.

1.4 Области-штабели (stacked-area)

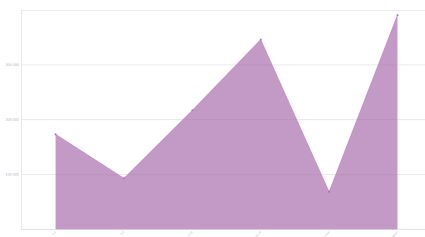


Рис. 1.4 Области

1.5 Штабели горизонтальные (stacked-bar)

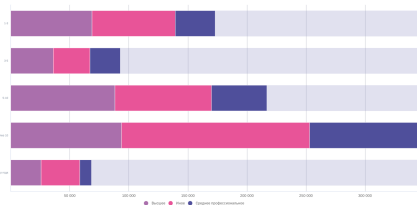


Рис. 1.5 Штабели горизонтальные

Представляет данные в виде горизонтального столбчатого графика с накоплением.

1.6 Штабели вертикальные (stacked-column)

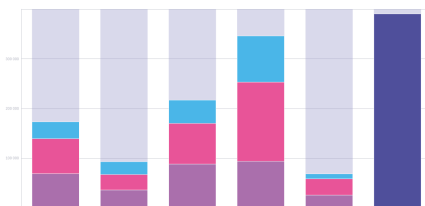


Рис. 1.6 Штабели вертикальные

Представляет данные в виде столбчатого графика с накоплением.

1.7 Точки (scatter)

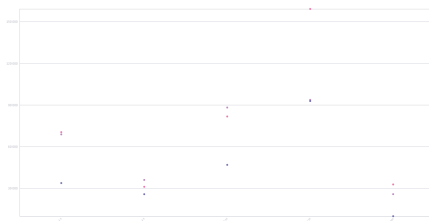


Рис. 1.7 Точки

Точечный график (множество точек, не соединенных между собой).

1.8 Сплайн (spline)

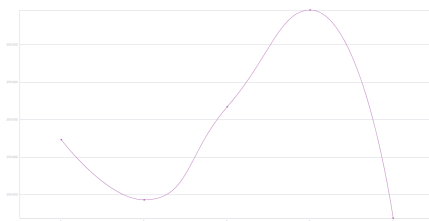


Рис. 1.8 Сплайн

Представляет данные в виде плавного линейного графика.

1.9 Водопад (waterfall)



Рис. 1.9 Водопад

Визуальное представление для факторного анализа.

1.10 Спидометр (gauge)

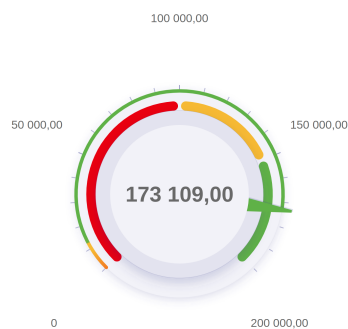


Рис. 1.10 Спидометр

Представляет данные на текущий момент в соответствии с нормативными значениями.

1.11 Спидометр полукруглый (semicircle)

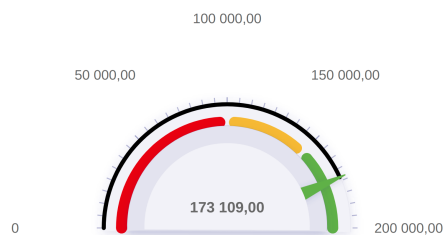


Рис. 1.11 Полукруглый спидометр

Представляет данные на текущий момент в соответствии с нормативными значениями.

1.12 Термометр (thermometer)



Рис. 1.12 Термометр

Представляет данные на текущий момент в соответствии с нормативными значениями.

1.13 Пирог (pie)

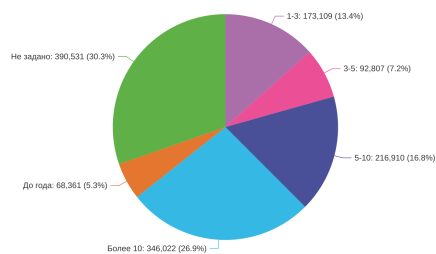


Рис. 1.13 Пирог

Показывает доленое отношение нескольких значений на текущий момент.

1.14 Пончик

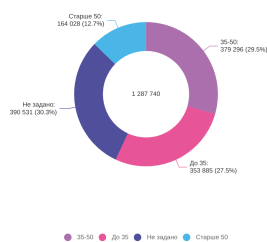


Рис. 1.14 Пончик

Показывает долевое отношение нескольких значений на текущий момент. В центре отображена сумма по отображаемым сегментам.

1.15 Весы (scales)



Рис. 1.15 Весы

Показывает отношение двух значений на текущий момент.

1.16 Воронка (funnel)

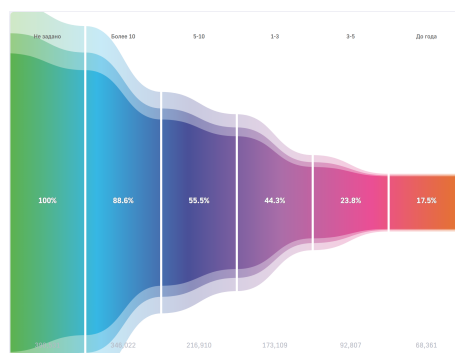


Рис. 1.16 Воронка

Представляет данные на текущий момент в виде воронки.

1.17 Радар (radar)

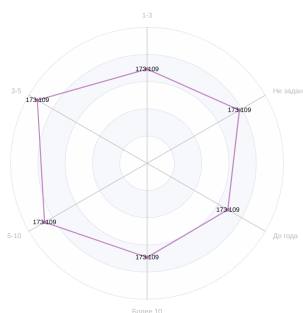


Рис. 1.17 Радар

Представляет данные на текущий момент в виде радиальной диаграммы.

1.18 Значение (label)

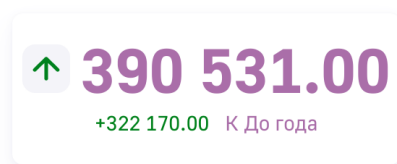


Рис. 1.18 Значение

Показывает одно значение на текущий момент в виде цифры. По умолчанию выводится с символом роста/падения и абсолютным отклонением от прошлого периода.

1.19 Древовидная карта (treemap)

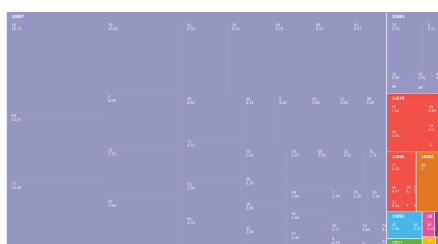


Рис. 1.19 Древовидная карта (treemap)

Отражает иерархические данные как набор вложенных прямоугольников

1.20 Санкей (sankey)

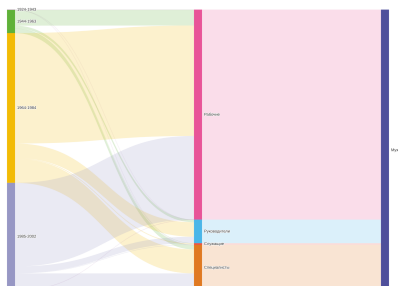


Рис. 1.20 Диаграмма Санкей (sankey)

Подчеркивает поток/движение/переход от одного состояния к другому или от одного времени к другому, в которой ширина стрелок пропорциональна скорости потока изображаемого обширного свойства.

1.21 ABC анализ

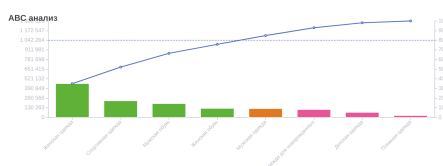


Рис. 1.21 ABC анализ

Метод анализа, позволяющий классифицировать ресурсы фирмы по степени их важности.

1.22 Сетчатая диаграмма



Рис. 1.22 Сетчатая диаграмма

Подобна точечной, за исключением того, что значения измерений откладываются на осях, а символы графика определяются с помощью выражения

1.23 Таблица с группировкой (tableP)

ФВВ	2003						2004
	Продажи	Продажи	Налие	Налие	Продажи	Продажи	Продажи
Ред Карсон	-	-	-	-	-	-	39 233.61
Фрэнк Рокс	-	-	-	-	-	-	38 233.65
Мэри Карсон	-	-	-	-	-	-	18 078.52
Том Карсон	-	-	-	-	-	-	-
Лард Шайк	1 734.05	1 734.05	30 000.00	30 000.00	5.78	5.78	35 000.00
Леонард Селл-Пул	-	-	-	-	-	-	15 001.42
Дэвид Карсон	-	-	-	-	-	-	12 734.38

Рис. 1.23 tableP

Представляет данные в табличном виде. В основном требуется для отображения агрегированных данных.

1.24 Список (list)

Не задано	407 288
Рабочие	2 509
Руководители	243
Служащие	130
Специалисты	828

Рис. 1.24 Список

Показывает несколько значений на текущий момент в виде списка.

1.25 What-if



Рис. 1.25 What-if анализ

Тип анализа, который исследует эффект изменения в параметре на результат

2. Кастомизируемые:

2.1 Карта (map)

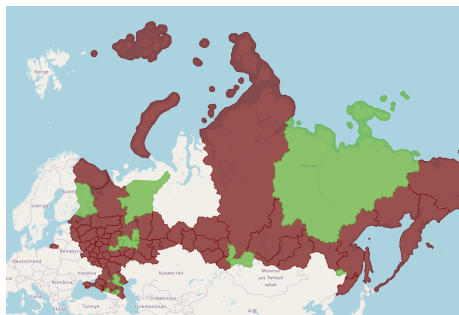


Рис. 1.26 Карта (map)

Позволяет визуально сравнить географические области по значению показателя или по выполнению норматива (требуется добавить в конфиг даша типа map опции “fillAreasByData” - окрашивание, “fillLegend” - легенда для него).

2.2 Внешний - используется для встраивания iframe в дэш (подробности в руководстве по кастомизации фронтенда)

2.3 Внутренний - дэш, для отображения кастомных React-компонентов (подробности в руководстве по кастомизации фронтенда)

2.4 Изображение (image)



Рис. 1.27 Изображение

Статичная картинка.

2.5 HTML (html)

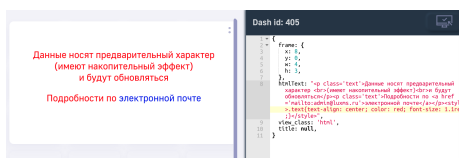


Рис. 1.28 Отображение html-дэша

Дэш, используемый для отображения статичного текста с использованием html-верстки. В поле `htmlText` указывается html-верстка. Верстка прописывается в кавычках. В примере ниже продемонстрировано использование `mailto`, стилей, и тега `p` в дэше `html`:


```

1      htmlText: "<p class='text'>Данные носят предварительный характер
2      <br>(имеют накопительный эффект)<br>и будут обновляться</p><p class=
'text'>Подробности по
3      <a href='mailto:admin@luxms.ru'>электронной почте</a></p>
4      <style>.text{text-align: center; color: red; font-size: 1.1rem;}</style>",
5      view_class: 'html',

```

2.6 Плоская таблица

Страна заказчика	Город заказчика	Себестоимость	Прибыль
Швейцария	Берн	962.61	16.09
Франция	Марсель	2 699.82	16.29
Германия	Берлин	51 965.76	16.54
США	Вала-Вала	211	23.88
Австрия	Грац	7 009.39	4.71
США	Юджин	61 926.56	11.49
Аргентина	Буэнос Айрес	8 443.62	23.83
Дания	Копенгаген	45 329.49	14.53

Рис. 1.29 Таблица “Данные”

Простая плоская таблица

2.7 Сводная таблица

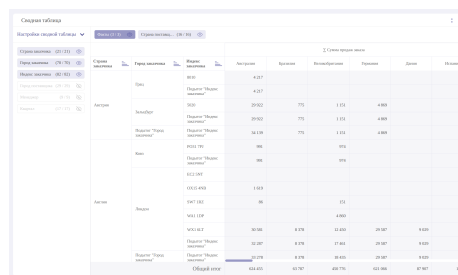


Рис. 1.30 Сводная таблица

Таблица с возможностью манипулирования отображением данных и группировкой данных по размерностям.

2.8 Схема (plan)



Рис. 1.31 Схема (plan)

Представляет данные в виде кастомизированной графики с возможностью детализации.

3. Группирующие:

3.1 Доска (board)

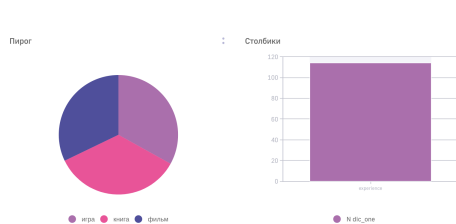


Рис. 1.32 Доска (board)

Позволяет задать для отображения внутри даша более одного визеля через список оных в поле children.

3.2 Вкладки (tabs)

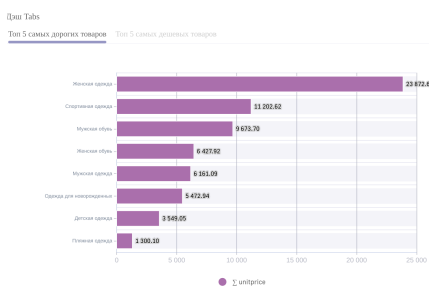


Рис. 1.33 Вкладки (tabs)

Позволяет выбирать какой из дочерних визелей будет отображен.

4. Статистика:

4.1 Гистограмма (histogram)

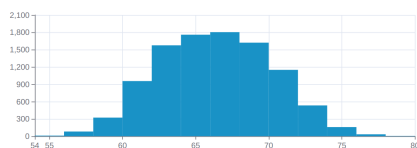


Рис. 1.34 Гистограмма (histogram)

При указании пустого объекта forecast в конфигурации на верхнем уровне даша histogram будет построено наилучшее приближение для данного распределения (распределением Гаусса):

1 forecast: {}

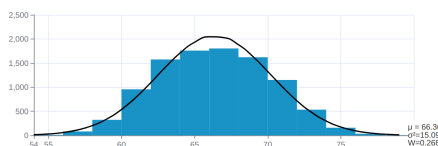


Рис. 1.35 Гистограмма с отображением нормального распределения



Дэш “Гистограмма” работает корректно только с фактами без агрегации

4.2 Ящик с усами (boxplot)

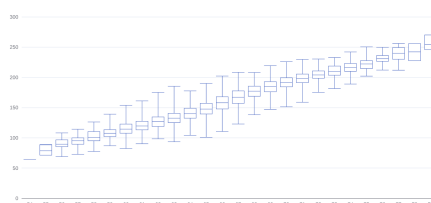


Рис. 1.36 Ящик с усами

График, использующийся в описательной статистике, компактно изображающий одномерное распределение вероятностей

4.3 Пузырьки

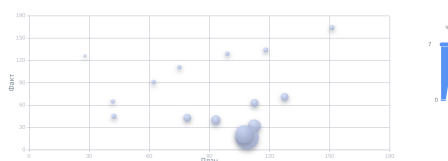


Рис. 1.37 Пузырьки

Визуальное представление корреляции данных.

Также в Luxms BI можно создать следующие визуализации, не представленные в режиме редактирования:

1. Комбинированный график

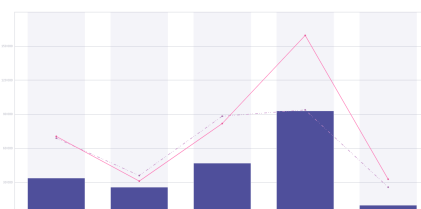


Рис. 1.38 Комбинирование

Объединяет два типа визуализации на одном графике, например, столбики и линии (для каждой величины задается свой тип графика через настройку widgetType в блоке style секции dataSource).

2. lookup-table

Name	Business category	Business category	Date name	Date name	Business category	Business category	Business category
1001	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1002	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1003	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1004	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1005	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1006	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1007	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1008	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1009	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1010	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1011	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1012	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1013	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1014	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1015	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1016	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1017	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1018	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1019	Business	Business	2020-01-01	2020-01-01	Business	Business	Business
1020	Business	Business	2020-01-01	2020-01-01	Business	Business	Business

Рис. 1.39 lookup-таблица

Таблица, отображающая результат запрос в кастомную таблицу БД.

3. axes-selector



Рис. 1.40 axes-selector

Дэш, с возможностью переключения осей, кубов и фактов для различного варианта отображения.

1.2 JSON конфиг дэша

JSON структура описания дэша на верхнем уровне состоит из следующих секций и опций:

- frame
- dataSource
- display
- url
- bgImage
- options
- hierarchy
- onClickDataPoint
- children
- view_class
- title

1.2.1 Поле title

В данном поле указывается заголовок дэшей на дэшборде.

Формат:

```
1 title: 'Название дэша'
```

Стилизация заголовка

В title можно указать lpe-выражение с использованием функций **bold** и **italic**.

```
1 title: 'lpe:italic("Сравнение") + " минимума и" + bold(" 0.5-квантиля")',
2   display: {
3     headerStyle: {
4       color: 'white',
5       fontSize: 30,
6       fontFamily: 'Times New Roman',
7       backgroundColor: '#E0AB9F',
8     },
9   },
```

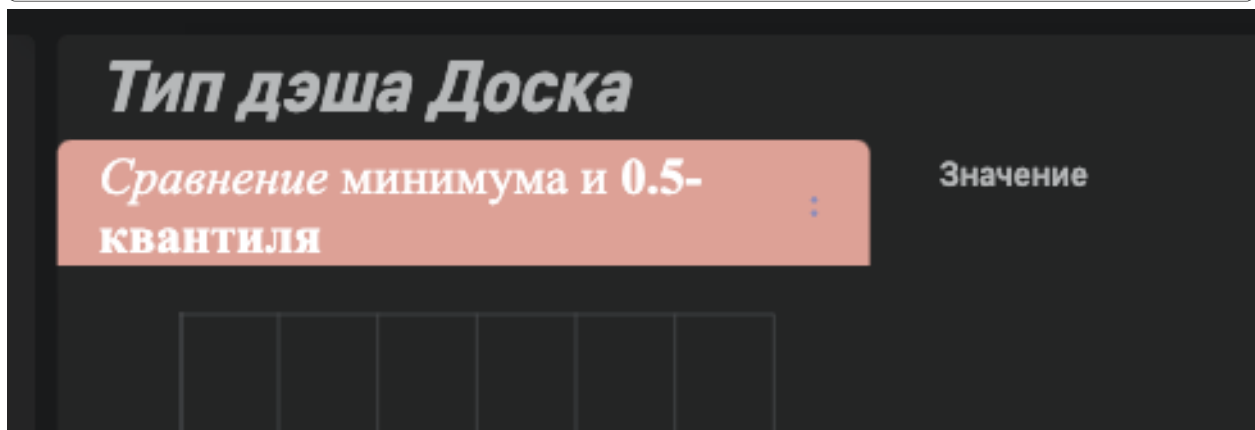


Рис. 1.41 title_stylization.png

Использование **getFilter** в заголовке

Обычно, в заголовке дэша указывается строка, но для динамического изменения заголовка можно использовать LPE-выражение с функцией **getFilter**. Ниже представлен пример использования данной функции:

```
1 title: 'lpe:"Количество " + if(getFilter("sex") && getFilter("sex").1 = 
    "Женщины", "гинекологов", "урологов") + " (чел)"',
```

В данном примере через **+** производится объединение строк. Во втором слагаемом данного выражения подставляется строка в зависимости от установленных фильтров (логика работы фильтров описана в блоке **dataSource** подразделе **filters**).

Функция **getFilter** принимает на вход размерность, и возвращает **true** если данная размерность была отфильтрована, в противном случае **false**.

В случае, если необходимо проверить наличие конкретного показателя в выборке после фильтрации, добавляется дополнительный параметр число, который проверяет n-ый элемент выборки.

Например `getFilter("sex").1 = "Женщины"` вернет **true** если, показатель "Женщины" не отфильтрован, присутствует в выборе, и этот показатель выбран в единственном числе или является первым в списке отфильтрованных.

Ниже приведен пример работы вышеописанного LPE-выражения:

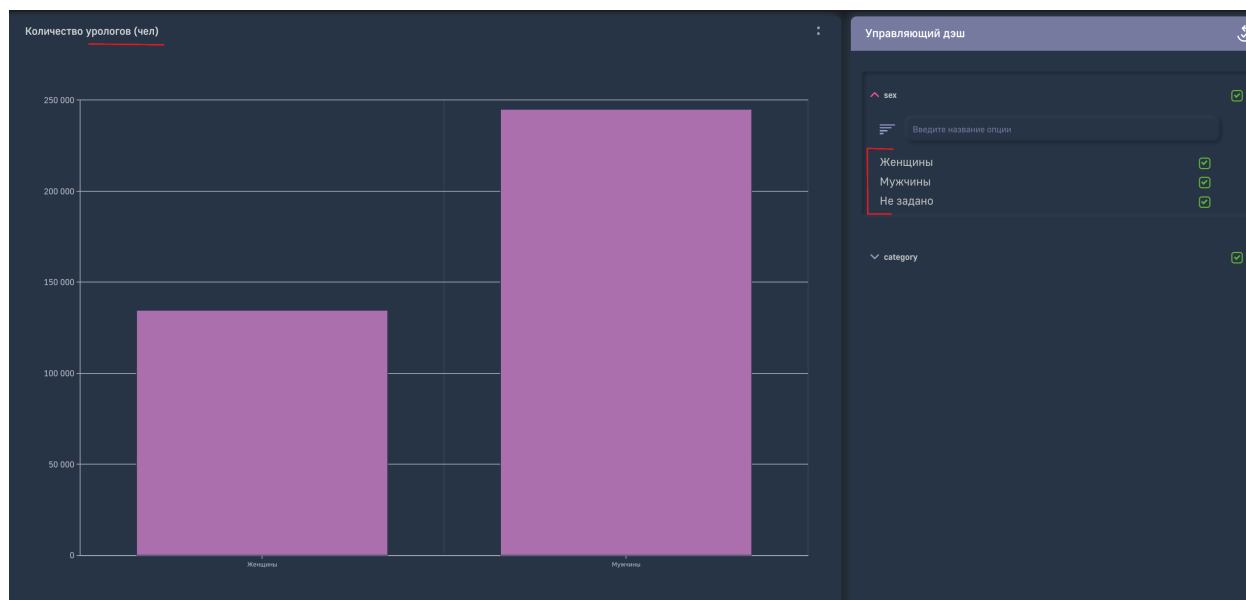


Рис. 1.42 Отображение заголовка до фильтрации

Если отфильтровать сначала все, затем выбрать показатель "Женщины" - заголовок изменится на следующий:

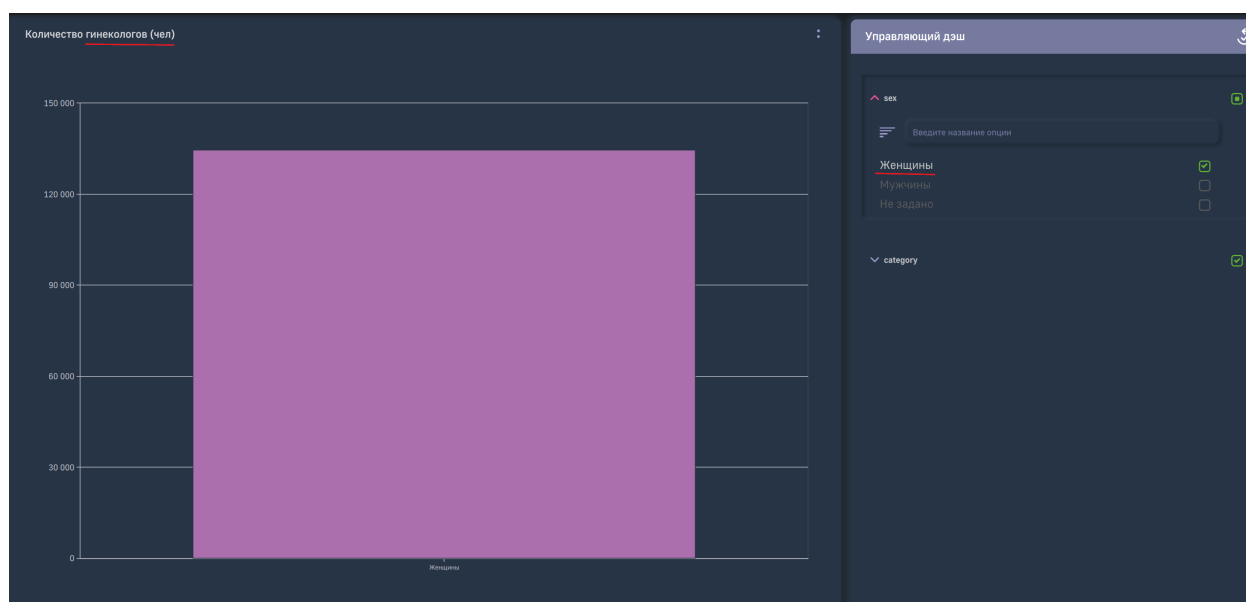


Рис. 1.43 Изменение заголовка после фильтрации

Данный функционал особенно полезен, в случае указания **defaultValue** для размерностей.

1.2.2 Поле description

В случае необходимости написания подробного описания дэша, который не поместится в заголовок вы можете указать описание дэша используя опыт **description**

Пример

```
1 description: 'Активы банка за 2016 календарный год'
```

После добавления данного поля у меню дэша появится иконка, при нажатии на которую отобразится описание:

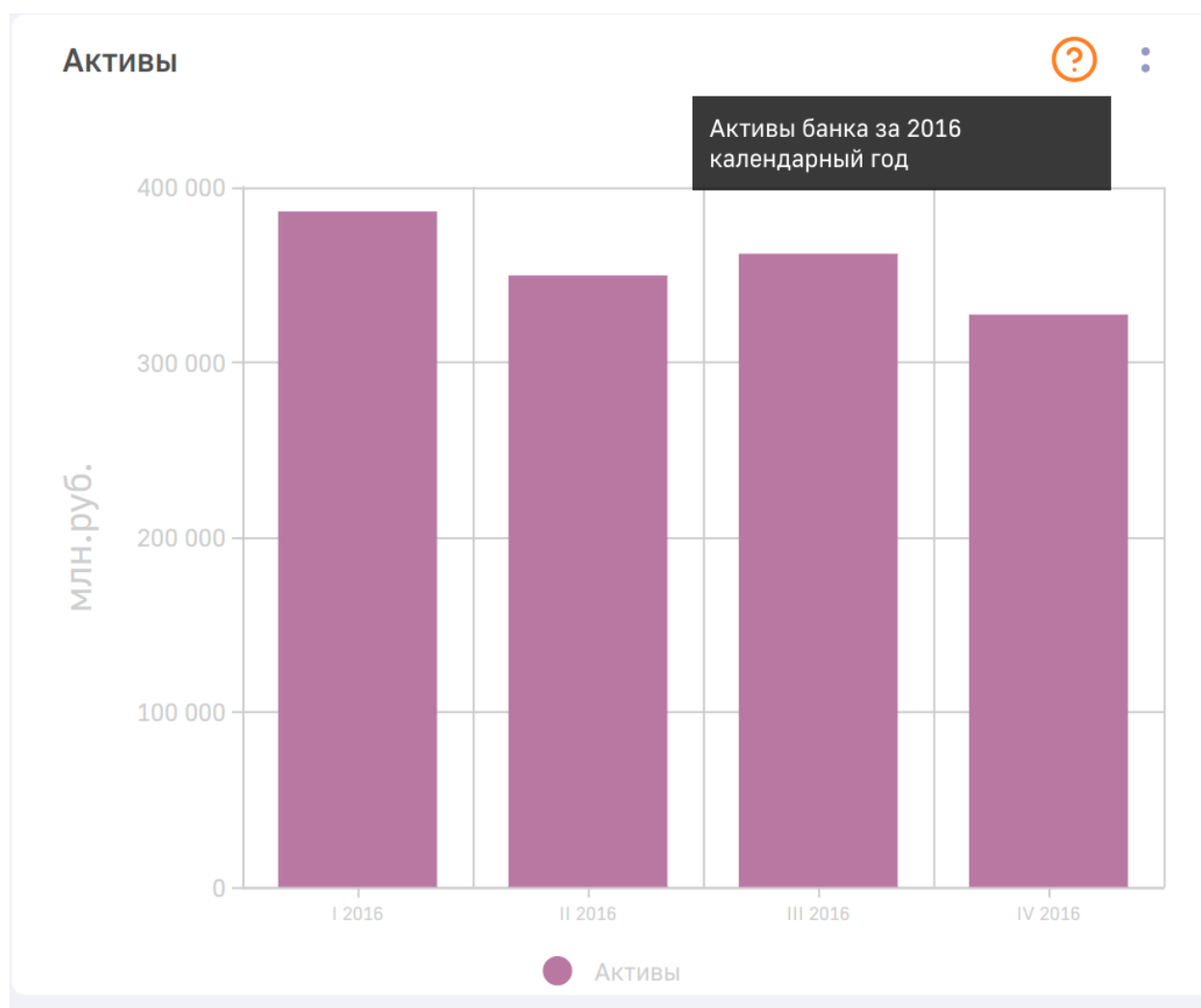


Рис. 1.44 Описание дэша

1.2.3 Секция frame

Данная секция используется для размещения создаваемых дэшей на дэшборде

Формат:

```

1  "frame": {
2    "h": Integer,
3    "w": Integer,
4    "x": Integer,
5    "y": Integer,
6  }

```

Где

- **h** - указание высоты в относительных единицах;
- **w** - указание ширины блока в относительных единицах;
- **x** - указание расположения дэша по оси X в относительных единицах;
- **y** - указание расположения дэша по оси Y в относительных единицах;

Дэш занимает все свободное пространство, в случае указания

```

1  "frame": {
2    "h": 0,
3    "w": 0,
4    "x": 0,
5    "y": 0,
6  }

```

По мере увеличения количества дэшей на дэшборде, расположение будет меняться, маневрируя показателями в секции **frame**

1.2.4 Поле bgImage

Используется для добавления изображения из ресурсов атласа на задний фон дэша.

Пример использования:

```

1  bgImage: 'res:image.svg',

```

1.2.5 Секция dataSource

Описывает источник данных для дэша, измерения (dimension) и меры (measures)

Формат:

```

1  {
2    "dataSource": {
3      "koob": "String",
4      "dimensions": [],
5      "measures": [],
6      "xAxis": "String",
7      "yAxis": "String",
8      "filters": {} или [],

```



```

9      "sortBy": "String",
10     "style": {
11       "String": {
12         "String": {
13           "title": "String",
14           "color": "String"
15         }
16       },
17       "measures": {
18         "String": {
19           "title": "String",
20           "color": "String",
21           "format": "String",
22           "vizelType": "String"
23         }
24       }
25     }
26   }
27 }

```

1.2.5.1 Опции секции dataSource

“koob”:

- тип STRING, имя куба. Указание куба происходит с указанием источника данных и куба. Например:

```
1 koob: "название_источника.название_куба"
```

“dimensions” : []

- Принимает массив STRING через запятую, имена столбцов у куба (должны быть известны заранее)
- Обязателен, для загрузки данных по этим столбцам
- Через двоеточие можно задать псевдоним. Если у id столбца в конце поставит двоеточие и условное слово, к примеру, newId. Дальнейшее обращение можно будет сделать через этот идентификатор.
- Принимает формулы или функции, где обязательно нужно будет задавать новый id через двоеточие в конце
- Пример: "dimension":["dt", "sex", "age", "concat(age, sex):id2"]
- У размерности типа “период” есть возможность указания формата отображения. Для этого вы можете использовать функции форматирования даты своего источника. в Postgres для этого используется TO_CHAR. Пример: "dimension":["TO_CHAR(dt, MM.DD):new_date", "sex", "age", "concat(age, sex):id2"]

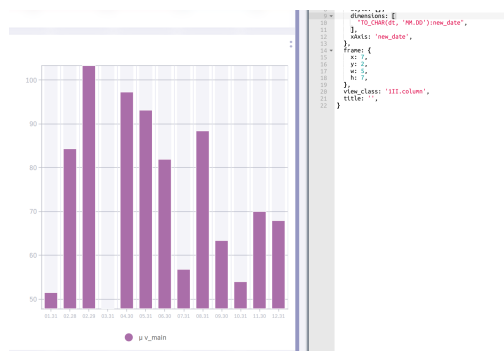


Рис. 1.45 Форматирование даты

“measures” : []

- Тип массив STRING через запятую, формула которая применяется к столбцам куба.
- Обязателен, для получения значений
- Если у id столбца в конце поставить двоеточие и условное слово, к примеру, newId. Дальнейшее обращение можно будет сделать через этот идентификатор.
- Пример: `"measures": ["sum(v_main):id1", sum(v_main):newId, "count(v_main)"]`.

Также в подсецию **“measures”** можно включить условие ее формирования, например, `"if (sum(pers)=0, 0, sum(fin)/sum(pers)):d"`. В результате на сервере сформируется SELECT. Для данного примера - `CASE WHEN sum(pers) = 0 THEN 0 ELSE sum(fin) / sum(pers) END as "d"`.

- Для сводной таблицы (pivot/table) реализована возможность отображения по умолчанию не всех фактов. Для этого в конфигурации необходимо прописать массив `innerMeasures`. Те факты, которые будут указаны в данном массиве будут включены по-умолчанию.
- Пример: `"innerMeasures": ["id1", "newId"]`.

“xAxis”, “yAxis” : ““

- Тип String, набор id dimensions
- id записываются в строку через точку запятой
- Есть спец id **‘measures’**, при котором на оси будут находиться все id measures



Пример: `"xAxis": "sex;age", "yAxis": "measures;dt"`.

“filters” : { } или []

- Тип OBJECT содержит ключи id измерений (dimension)
- Значения id ключа могут быть:

1. **true** - данные по этому фильтру будет подтягиваться из управляющего деша 2. массив STRING, установка определенного фильтра, значения из управляющего деша будут проигнорированы



```
filters: { country: true, },
```



```
filters: { sex: [ '!=', 'Мужчины', ], },
```

- Если необходимо исключить один из показателей размерности, но чтобы размерность реагировала на действия в управляющем дэше



```
filters: { "": [ '!=', [ 'column', 'sex', ], 'Мужчины', ],
sex: true, },
```

Пример использования нескольких размерностей в filters:

```

1 filters: {
2   '': [
3     'and',
4     [
5       '!=',
6       [
7         'column',
8         'country',
9       ],
10      'United Kingdom',
11    ],
12    [
13      'ilike',
14      [
15        'column',
16        'productname',
17      ],
18      'Hot Baths',
19    ],
20    [
21      '=',
22      [
23        'column',
24        'date',
25      ],
26      '2018-12-01',
27    ],
28  ],
29  country: true,
30 },
```

- Для сводной таблицы также присутствует возможность предустановки фильтрации для размерностей с помощью **innerFilters**. Задается аналогично **filters**

Пример:



```
innerFilters: { "sex": ["!=", "Мужчины"] }
```

“sortBy”

- тип STRING, в строке поочередно указываются id размерностей с указанием знака сортировки



Пример: `"sortBy": "+region;-city;+district"`

“style” : {}

- тип OBJECT, принимает id размерностей и спец. ключ **“measures”**
- **ключ id “dimension”** :
 1. тип OBJECT
 2. принимает id ключ из массива **dataSource.dimensions**

- тип OBJECT - ключ **“title”**: тип STRING, меняет заголовок dimension - ключ **“color”**: тип STRING, меняет цвет dimension

3. Пример : `style:{ "sex":{ "Мужчина":{ "title": "мужчина" } } }`

- **спец ключ “measures”**:

1. тип OBJECT
2. Принимает id ключ из массива **dataSource.measures**

- тип OBJECT

- ключ **“title”**: тип STRING, меняет заголовок measure. Для данного ключа также можно пользоваться LPE-выражением и выводить различные заголовки в зависимости от установленных фильтров (функция `getFilters` описанна выше, в секции `title`).

```
1 title: 'lpe:"Количество " + if(getFilter("sex") && getFilter("sex").1 = (↔)
  "Женщины", "гинекологов", "урологов") + " (чел)"',
```






```
1 - ключ **"color"**: тип STRING, меняет цвет measures. Используется значение # (↔)
  HEX для указания цвета
```

```
3 - ключ **"format"**: ниже подробное описание**
```

```
5 - ключ **"vizelType"**: тип STRING, меняет тип графика для выбранного факта / p (↔)
  азмерности. Имеет следующие типы значений:
```

```
7 a. `bar` - Отображение данного показателя в виде столбиков
```

```
9 b. `line` - Отображение данного показателя в виде линии
```

- 11 c. `spline` - Отображение данного показателя в виде плавной линии (сплайн)
- 13 - ключ `**"strokeStyle"**` в случае, если для дэша выбран тип отображения 
 "Линии", "Сплайн" или "Области", то дэш меняет свое отображение в зависимости 
 от выбранного значения. Ниже представлен список доступных значений:
- 15 a. `Solid` - сплошная линия
- 16 b. `ShortDot` - линия отображена точками, расположенными близко относительно 
 друг друга
- 17 c. `ShortDash` - линия отображена черточками небольшой длины
- 18 d. `ShortDashDot` - линия отображена черточками небольшой длины и точками
- 19 e. `ShortDashDotDot` - линия отображена черточками небольшой длины и двумя т 
 очками
- 20 f. `Dot` - отображение точками
- 21 g. `Dash` - линия отображена черточками
- 22 h. `LongDash` - линия отображена длинными черточками(тире)
- 23 i. `DashDot` - линия отображена черточками и точками
- 24 j. `LongDashDot` - линия отображена длинными черточками(тире) и точкой
- 25 k. `LongDashDotDot` - линия отображена длинными черточками(тире) и двумя то 
 чками

Пример:

```

1  "style": {
2    "measures": {
3      "newId": {
4        "title": "мой новый заголовок",
5        "color": "#f2f2f2",
6        "format": "# ###",
7        "vizeType": "spline"
8      }
9    }
10  }
```

Для указания стилей всем фактам разом, в конфигурации вместо названия факта указывается символ `*`. Пример:

```

1  "style": {
2    "measures": {
3      "*": {
4        "title": "мой новый заголовок",
5        "color": "#f2f2f2",
6        "format": "# ###",
7      }
8    }
9  }
```

Аналогичным способом можно указать стили для всех показателей размерности:

```

1  "style": {
2    "category": {
3      "*": {
```

```

4      "color": "#f2f2f2",
5      "format": "# ###",
6    }
7  }
8 }

```

Цвет в одномерных и двумерных дэшах можно задать в зависимости от значения. Для этого необходимо окрашиваемому показателю указать опцию **ColorX** и указать LPE-выражение, по которому столбцы будут окрашиваться. Ниже представлен пример конфигурации:

```

1    style: {
2      measures: {
3        '*': {
4          options: [
5            'ColorX',
6          ],
7          color: "lpe:if(avg_v_main < 15000, 'green', 'red')",
8        },
9      },
10   },

```

В случае необходимости сортировки данных по размерности, которую выводить не нужно можно размерности прописать **title** с пустой строкой.

Пример:

```

1  "style": {
2    "month_id": {
3      "": {
4        "title": ''
5      }
6    }
7  }

```

Заголовок для подытогов задается внутри объекта, имеющего в названии символ суммы Σ (U+2211) и наименование размерности:

Пример:

```

1  "style": {
2    "category": {
3      "Σcategory": {
4        "title": 'Подытог '
5      }
6    }
7  }

```

“limit”

- тип NUMBER, ограничение по количеству выводимых данных
- Обрезает ось-X на заданное зн-ие



Пример: `"limit": 3`

Только для **koob-table-simple**:

1. Принимает id ключ из массива **dataSource.measures**
2. Создаем любые переменные в виде ключ : выражение
3. **backgroundColor**: будет либо цвета:
 - a. если `myConst1 = true`, `backgroundColor: '#00b9ac'`
 - b. если `myConst2 = true`, `backgroundColor: '#fed450'`
 - c. если `(myConst1 && myConst2) = false`, `'#93CAFE'`
 - d. Пример: `"newId": {"myConst1": "done_date != null && delivery_target_date > due_date"}`

Пример:

```

1  "style": {
2    "measures": {
3      "newId": {
4        "myConst1": используется lpe-выражение
5                      (true || false),
6        "myConst2": используется lpe-выражение
7                      (true || false),
8        "backgroundColor": "lpe:when(lpe(myConst1), '#00b9ac',
9                               lpe(myConst1), '#fed450', '#93CAFE')"
10     }
11   }
12 }
```

“subtotals” : “dimensions” Только для таблицы **tableP** и сводной таблицы (**pivot/table**)

- тип **STRING**, принимает список размерностей, для которых необходимо выводить подытог
- работает для источника данных **PostgreSQL**

Пример : `subtotals: "sex;age;degree"`

“format” :

- тип **STRING**
- Используется в типах дэша: **text**
- формат задается исходя из документации ([по данной ссылке](#))
- Примеры:
 1. `"format": "-# ###,0%"` - значение будет выводиться с один знаком после запятой с символом “%”
 2. `"format": "#[/ 2]"` - отображаемое значение в тултипе и подписях будет поделено на 2

3. "format": "# ### [тыс, млн, млрд, тера]" - значение будет отображаться в зависимости от количества цифр в числе. Пример: число "220 000" будет выводиться как "220 тыс", число "200 000 000" будет выводиться как "220 млн" и т.д.
4. "format": "△ #" округляет значение в большую сторону, а "▽ #" в меньшую



Для виджета "Значение" и значения внутри виджета "Бублик" `format` необходимо указывать в поле `display`, для остальных случаев `format` прописывается в поле `style` рассматриваемого факта/размерности.

Если два и более фактов имеют различные единицы измерения и их необходимо отобразить в одном дэше на разных осях, необходимо каждому из фактов указать `format`. Например:

```
1 style: {
2   measures: {
3     avg_naselenie: {
4       format: '# ### человек',
5     },
6   },
7 }
```

После этого на дэше будет отображено две оси:

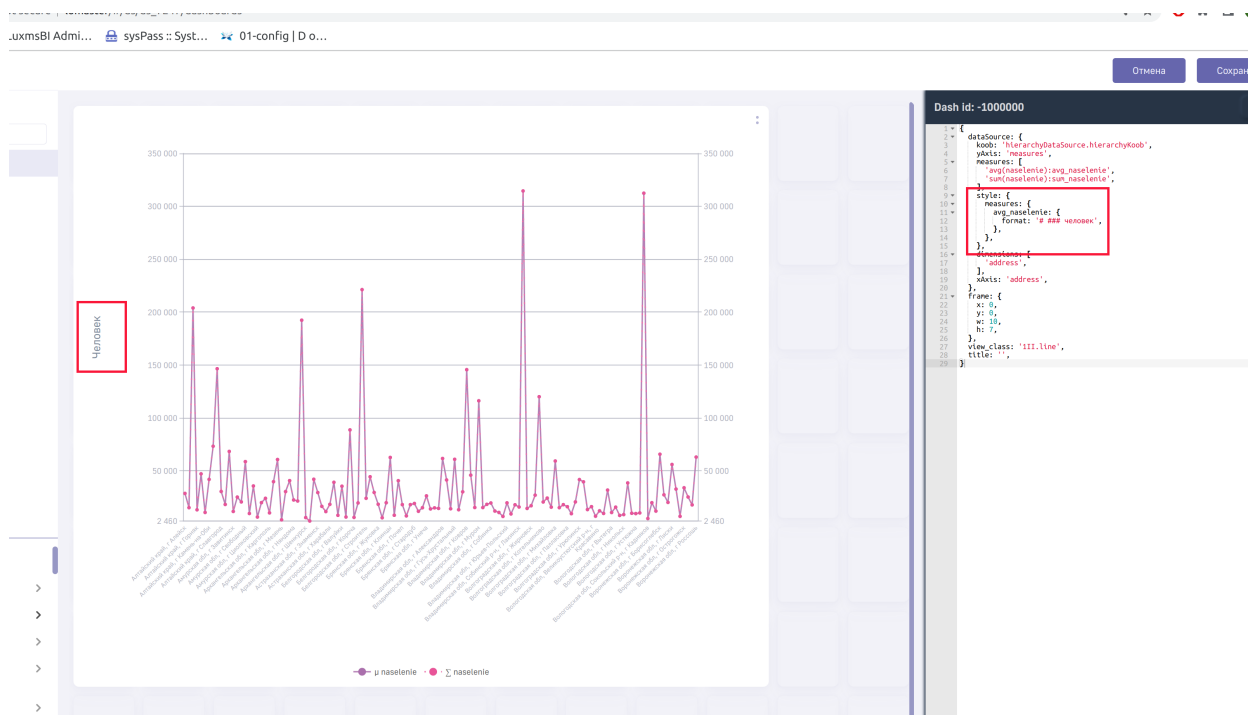


Рис. 1.46 Отображение двух осей в дэше

1.2.6 Секция display

Описывает отображение дэша

Формат:


```

1 {
2   "display": {
3     "limit": Integer,
4     "range": Integer,
5     "format": "String",
6     "maxFontSize": Integer,
7     "minFontSize" : Integer,
8     "bgColor": "String",
9     "color": "String",
10    "customValue": "String",
11    "stoplight": {
12      "lights": [
13        {
14          "name": "String",
15          "color": "String",
16          "limit": [
17            Integer, ...
18          ],
19          "bgColor": "String"
20        }
21      ]
22    }
23  }
24 }

```

1.2.6.1 Опции секции display

“limit” :

- тип NUMBER
- Обрезает ось-X на заданное зн-ие
- Пример `"limit":20`

“range” :

- Принимает массив значений, типа NUMBER
- Применяется на типе визеля с лимитами
- Пример: `"range": [0,30000]`
- range можно также указать для конкретной единицы измерения. Для этого необходимо перейти в раздел единиц измерения, редактировать или создать новую единицу измерения, и прописать в ее конфигурации `range`

“stoplight”:

- тип OBJECT

“lights”:

- Принимает массив OBJECT, ключи которого:

- **name** : имя лимита (STRING)
- **color** : цвет (STRING)
- **limit** : промежуток цвета (массив NUMBER)
- **bgColor** : цвет фона у показателя (STRING)

- Пример:

```

1 "lights": [ {
2   "name": "red",
3   "color": "#ff4d4d",
4   "limit": [20000,3000],
5   "bgColor" :#d4f6dc
6   },
7   {...},{...} ],

```

“fontSize” :

- тип NUMBER || STRING
- Используется в типах даша: **text**
- Размер текста
- Пример: `"fontSize": "30"`

“cmpTitle” :

- тип STRING
- Используется в типах даша: **gauge**
- Указание текстовой подписи под значением
- Пример: `"cmpTitle": "Нижний заголовок"`

“gap” : - тип NUMBER || STRING - Используется в типах даша: **board** - Указание отступов между дочерними элементами - Пример: `gap: "5"`

“selectedColor” : - тип STRING - Используется в типах даша: **map** - Указание цвета обводки при включенной множественной фильтрации на карте - Пример: `selectedColor: "#000000"`

“stripes” : - тип BOOLEAN - Используется в типах даша: **koob-table-simple** - Отображение таблицы с чередующейся покраской строк - Пример: `stripes: true`

“borderRadius” : - тип NUMBER - Используется в типах даша: **bar, column, stacked-bar, stacked-column** - скругление столбиков - Пример: `borderRadius: 30`

1.2.6.2 Визуализация тултипов

Визуализация тултипов возможна с использованием опции `display`. В данном разделе будет рассмотрено два варианта конфигурации отображения тултипов: отображение в тултипе визеля и отображение в тултипе математических формул. Для данных действий используется опция `tooltip`. Для отображения визеля внутри тултипа необходимо `tooltip` указать как объект и прописать внутри него конфигурацию. Пример:

```

1 "display": {
2   "tooltip": {
3     "dataSource": {
4       "xAxis": "measures",
5       "yAxis": "sex",
6       "dimensions": ["sex"],
7     },
8     "view_class": "pie",
9   },
10 },

```

Отображение данного тултипа продемонстрировано ниже:



Рис. 1.47 Отображение визеля в тултипе

Для отображения формул `tooltip` необходимо задавать как строку с формулой LaTeX. Ниже представлен пример указания формулы:

```

1 "display": {
2   "tooltip": "latex:%y: $f_x(%x) = %v^2$",
3 }

```

где `%y` - наименование показателя, отложенного на оси Y, `%x` - наименование показателя, отложенного на оси X, `%v` - значение

Ниже пример отображения тултипа с использованием данной функции:

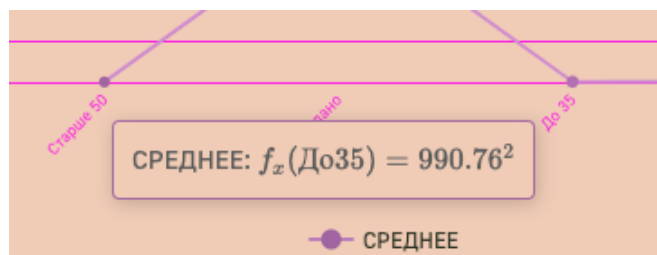


Рис. 1.48 Математические формулы в тултипе

В тултипах также можно указать собственную подпись, выводить значения и наименования показателей используя HTML верстку. В случае необходимости указания общего тултипа для всех показателей, его необходимо прописать в блоке `display`:

```
1 "display": {
2   "tooltip": "html:общий формат тултипов с заменой %x %y %v <b>%percent %marker"
3 },
```

где **%x** и **%y** - наименования показателей, отложенные на **xAxis** и **yAxis** соответственно, **%v** - значение показателя, **%percent** - Вывод показателя в процентах (для дэшлетов “Пирог” и “Донат”), **%marker** - отображения цвета показателя в круге

1.2.7 Секция echart

Дэши в LuxmsBI отрисовываются с использованием библиотеки **Apache Echarts**. Для ряда дэшей существует возможность наследования конфигурационных полей из данной библиотеки. Ниже представлен список команд, которые можно указать в секции **echart**:

1. Указание ширины и цвета линий дэша:

```
1 echart: {
2   series: {
3     lineStyle: {
4       color: 'green',
5       width: 40,
6     },
7   },
8 },
```

2. Указание ширины столбцов:

```
1 echart: {
2   barWidth: 50,
3 },
```

3. Указание цвета и размера подписей для оси:

```
1  echart: {  
2    xAxis: {  
3      axisLabel: {  
4        textStyle: {  
5          color: 'gold',  
6          fontSize: 42,  
7        },  
8      },  
9    },  
10   yAxis: {  
11     axisLabel: {  
12       textStyle: {  
13         color: 'gold',  
14         fontSize: 42,  
15       },  
16     },  
17   },  
18 },
```

4. Для изменения варианта отображения точек для дэшей “Линии”, “Области”, “Точки”:

```
1  echart: {  
2    series: {  
3      symbol: 'diamond',  
4    },  
5  },
```

Возможные значения:

1. `diamond` - ромбики,
2. `circle` - круги (по-умолчанию),
3. `triangle` - треугольники,
4. `rect` - квадраты
5. Указание цвета лейбла на графике

```
1  echart: {  
2    series: {  
3      label: {  
4        color: 'red',  
5        position: 'inside',  
6      },  
7    },  
8  },
```

1.2.8 Секция forecast

Используется для указания параметров для расчета статистических данных и прогнозирования. Реализовано для дэшей типа “Линии”, “Точки”.

В данный момент в LuxmsBI представлено три алгоритма расчетов: линейная регрессия (**LINEAR_REGRESSION**), экспоненциальное сглаживание (**EXPONENTIAL_SMOOTHING**), метод Холт-Винтерса (**HOLT_WINTERS**).

Ниже представлен набор ключей для каждого из алгоритмов:

Линейная регрессия:

```
1 forecast: {  
2   steps: 10,  
3   degree: 2,  
4   algorithm: 'LINEAR_REGRESSION',  
5 }
```

где - **steps** - количество рассчитываемых значений. Допустимые значения: целые положительные числа, - **degree** - возводимая степень. Допустимые значения: целые положительные числа.

Экспоненциальное сглаживание:

```
1 forecast: {  
2   steps: 10,  
3   alpha: 0.1,  
4   gamma: 0.1,  
5   algorithm: 'EXPONENTIAL_SMOOTHING',  
6 }
```

где - **steps** - количество рассчитываемых значений. Допустимые значения: целые положительные числа, - **alpha** - коэффициент альфа. Допустимые значения: от 0 до 1, - **gamma** - коэффициент гамма. Допустимые значения: от 0 до 1.

Метод Холт-Винтерса:

```
1 forecast: {  
2   steps: 10,  
3   alpha: 0.1,  
4   gamma: 0.1,  
5   delta: 0.2,  
6   trend: 'MULT' | 'ADD',  
7   algorithm: 'HOLT_WINTERS',  
8   seasonLength: 7 | 12 ...,  
9 }
```

где - **steps** - количество рассчитываемых значений. Допустимые значения: целые положительные числа, - **alpha** - коэффициент альфа. Допустимые значения: от 0 до 1, - **gamma** - коэффициент гамма. Допустимые значения: от 0 до 1, - **delta** - коэффициент дельта. Допустимые значения: от 0 до 1, - **trend** - вариант расчета: мультипликативный (**MULT**), либо

аддитивный (ADD) - `seasonLength` - сезонность. Рекомендуемые значения: кратные рассматриваемому периоду (неделя - 7, год - 12 и т.д.)

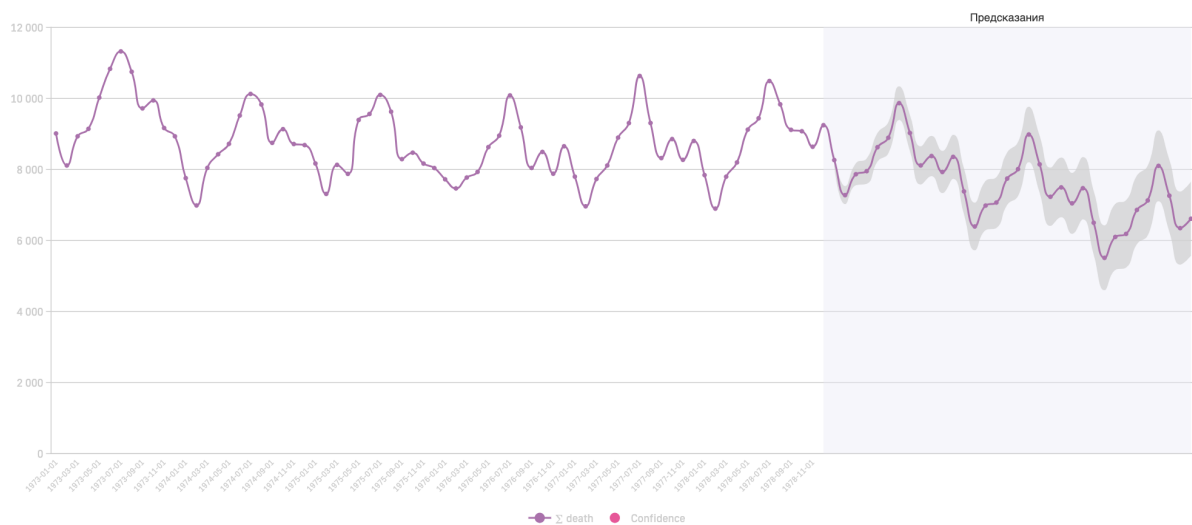


Рис. 1.49 Пример отображения дэша “Линии” с прогнозированием с использованием линейной регрессии

При указании пустого объекта **forecast** в конфигурации дэша `histogram` будет построено наилучшее приближение для данного распределения (распределением Гаусса). Выводятся параметры нормального распределения, такие как: мат.ожидание, ср.квдр.отклонение, и точность аппроксимации.

При указании пустого объекта **forecast** в конфигурации дэша `correlation` будет построена линейная аппроксимация.

1.2.9 Опция url

- тип STRING
- Используется в типах дэша: **external**, **internal**
- Является ссылкой на html файл (для **external**) или на js файл с компонентом написанном на React и содержащем `export default` (для **internal**)

url: путь к файлу из раздела `resources` датасета (текущего или любого другого существующего).

Пример: ссылка на файл `template.html`, который находится в разделе `resources` того же датасета, где и дешлет:

```
"res:template.html "
```

Ссылка на тот же файл, но если он находится не в текущем датасете, а в датасете с `schema_name = "ds_20"`:

```
"res:ds_20:template.html "
```

1.2.10 Секция options

Описывает различные дополнительные настройки при отображении дэша

“options” : [],

- массив STRING;
 - Пример "options": ["DisplayAllBadges", "HideLegend"].
1. **Fixed** - при включенной опции дэш закреплен на экране при скроллинге
 2. **DisableLoadData** - запретить загрузку у кубовых дэшей, в тех случаях, когда данные либо не нужны (например, управляющий дэш, what if), либо они загружаются иным запросом (это поможет избежать лишних запросов)
 3. **DisplayAllBadges** - показать все подписи на графиках без учета пересечений значений
 4. **DisplayAllVeryBadges** - показать все подписи принудительно
 5. **!TopBarMenu** - скрыть меню визеля
 6. **!TopBar** - скрыть заголовок дэша
 7. **HideLegend** - скрыть легенду на графике
 8. **!DisplayOverall** - для дэшей таблиц убирает итоговые строки внизу таблицы
 9. **DisplayOverall** - для дэшей таблиц добавляет итоговые строки внизу таблицы. (По умолчанию выводятся суммы по столбцам, для использования другой формулы используйте поле formula, описанную ниже.)
 10. **EastPanel** - для дублирования дэша в правой выдвигающейся панели (рекомендуется использовать только для управляющего дэша)
 11. **!DisplayAxisYMarks** - скрыть ось Y у дэша (работает для двумерных дэшей)
 12. **!DisplayAxisXMarks** - скрыть ось X у дэша (работает для двумерных дэшей)
 13. **TooltipXAxisTitle** - отображение полного наименования показателя размерности в всплывающей подсказке (тултипе)
 14. **!DisplayAxis** - скрытие сетки и осей у двумерных дэшей
 15. **!ShowBackground** - скрытие теней у столбцов
 16. **!DisplaySplitLines** - скрытие разделительных линий у двумерных дэшей
 17. **!DisplayTicks** - скрытие рисок/тиков осей у двумерных дэшей

18. `Transparent` - фон визелей “Значение” и “Внутренний” соответствует фону подложки дэшборда
19. `XAxisValue` - переключение оси X с категориальной на численный вариант отображения (применяется для двумерных дэшей)
20. `MergeCells` - опция, включающая объединение ячеек по вертикали в таблице `koob-table-simple`
21. `ModalDrilldown` - подключение старого варианта иерархии для дэша (Модальное окно с горизонтальными столбцами. Для подключения данного варианта на всю систему, необходимо в файле `/opt/luxmsbi/web/settings/settings.js` в массиве **features** указать данную опцию и перезапустить компонент)
22. `FullWidth` - растягивает все колонки таблицы на всю ширину (включено по умолчанию). Если указать, например, у одного столбика значительно большую длину, то может появиться скролл. При отключенной опции (`!FullWidth`) пользователь может задать своё значение ширины (`width`) для всех столбцов.
23. `!Sort` - выключение сортировки у дэша Плоская таблица
24. `!Expanded` - скрытие возможности разворачивания дэша на весь экран
25. `ResetButton/ !ResetButton` - опция управляющего дэша, которая позволяет показывать/скрыть иконку “Очистить все фильтры” и “Сбросить фильтр” в зависимости от уровня, на котором опция указана. По умолчанию у управляющего дэша и у размерности установлено `ResetButton`, если у размерности указан тип `radio` - `!ResetButton`
26. `ResetButtonToDefault/!ResetButtonToDefault` - опция управляющего дэша, которая позволяет показать/скрыть иконку “Сбросить фильтры по умолчанию” в зависимости от уровня, на котором опция указана. По умолчанию у управляющего дэша установлено `ResetButtonToDefault`, у размерности `!ResetButtonToDefault`, если у размерности указан тип `radio` - `ResetButtonToDefault`

Пример указания опций `ResetButton` и `ResetButtonToDefault`, который убирает из интерфейса кнопку “Сбросить фильтр” у конкретной размерности, кнопка “Сбросить фильтр по умолчанию” присутствует.

```
1  dataSource: {  
2    koob: 'a.b',  
3    style: {  
4      category: {  
5        type: 'radio',  
6        options: [  
7          'ResetButtonToDefault',  
8          '!ResetButton',  
9        ],  
10     },  
11  },  
12 }
```

Пример указания опций `ResetButton` и `ResetButtonToDefault`, который убирает из интерфейса кнопки “Сбросить фильтры по умолчанию” и “Очистить все фильтры” у всего управляющего дэша.

```

1 options: [
2   '!ResetButtonToDefault',
3   '!ResetButton',
4 ],

```

Помимо общих опций, есть опции для конкретный пользователей. Указываются данные опции в блоке style конкретной размерности. Ниже представлен пример указания опции для конкретной размерности:

```

1 "dataSource":{
2   "koob":"ch.max_example",
3   "style":{
4     "category":{
5       "Специалисты":{
6         "color":"red",
7         "options":[
8           "AlwaysLast",
9         ],
10      },
11    },
12  },
13 }

```

Ниже представлены опции для конкретных размерностей:

1. **AlwaysLast** - Независимо от включенной сортировки набора данных, данный показатель будет отображаться последним. Реализовано для двумерных (столбики) и одномерных дэш (бублик и пирог)
2. **DisableLegend** - По умолчанию данный показатель будет отображаться после нажатия на легенду. (реализовано для двумерных дэш)
3. **Pin** - Закрепление столбца при скролле плоской таблицы (реализовано только для плоской таблицы)

1.2.10.1 Списки опций, доступные для разных типов дэш

1. Опции, доступные для всех дэш:

- **Fixed**
- **TopBarMenu**
- **TopBar**
- **Hidden**
- **ModalDrilldown**

2. Опции, доступные для дэш "Линия", "Области-Штабели" и "Слайн":

- **DisplayAllBadges**
- **DisplayAllVeryBadges**

- EastPanel
- !DisplayAxisYMarks
- !DisplayAxisXMarks
- TooltipXAxisTitle
- !DisplayAxis
- !DisplaySplitLines
- !DisplayTicks
- Transparent
- DisableLegend
- YLogAxis
- LinesWithoutDots
- DisplayDataZoom

3. Опции, доступные для дэша “Точки”:

- DisplayAllBadges
- DisplayAllVeryBadges
- EastPanel
- !DisplayAxisYMarks
- !DisplayAxisXMarks
- TooltipXAxisTitle
- !DisplayAxis
- !DisplaySplitLines
- !DisplayTicks
- Transparent
- DisableLegend
- YLogAxis
- DisplayDataZoom

4. Опции, доступные для дэшэй “Столбики горизонтальные” и “Столбики вертикальные”:

- DisplayAllBadges
- DisplayAllVeryBadges
- EastPanel
- DisplayAxisYMarks
- DisplayAxisXMarks
- TooltipXAxisTitle
- DisplayAxis
- DisplaySplitLines
- DisplayTicks
- Transparent
- DisableLegend
- YLogAxis
- LinesWithoutDots
- DisplayDataZoom

5. Опции, доступные для дэшэй “Штабели горизонтальные” и “Штабели вертикальные”:

- DisableLoadData
- DisplayAllBadges
- DisplayAllVeryBadges
- EastPanel
- DisplayAxisYMarks
- DisplayAxisXMarks
- TooltipXAxisTitle
- DisplayAxis
- DisplaySplitLines
- DisplayTicks
- Transparent
- DisableLegend
- YLogAxis
- LinesWithoutDots
- DisplayDataZoom
- DisplayBadgesPercent

6. Опции, доступные для дэша “Пирог”:

- DisplayAllBadges
- DisplayAllVeryBadges
- EastPanel
- !DisableLegend
- DisplayBadgesPercent
- DisplayBadgesValue

7. Опции, доступные для дэша “Пончик”:

- DisplayAllBadges
- DisplayAllVeryBadges
- EastPanel
- !DisableLegend
- DisplayBadgesPercent
- DisplayBadgesValue
- !DisplayCenterLabel

8. Опции, доступные для дэша “Весы”:

- EastPanel
- Transparent
- !DisplayLegend
- DisplayBadgesPercent

9. Опции, доступные для дэша “Радар”:

- EastPanel
- Transparent
- !DisplayLegend

10. Опции, доступные для дэша “Список”:

- `EastPanel`
- `Transparent`

10. Опции, доступные для дэша “Воронка”:

- `Transparent`

11. Опции, доступные для дэша “Древовидная карта”:

- `EastPanel`

12. Опции, доступные для дэшэй “Термометр”, “Спидометр” и “Спидометр полукруглый”:

- `DisableLoadData`
- `EastPanel`

13. Опции, доступные для дэша “Значение”:

- `DisableLoadData`
- `EastPanel`
- `Transparent`

14. Опции, доступные для дэша “Водопад”:

- `DisableLoadData`
- `DisplayAllBadges`
- `EastPanel`
- `!DisplayAxisYMarks`
- `!DisplayAxisXMarks`
- `TooltipXAxisTitle`
- `!DisplayAxis`
- `!DisplaySplitLines`
- `!DisplayTicks`

15. Опции, доступные для дэша “Пузырьки”:

- `DisableLoadData`
- `EastPanel`

16. Опции, доступные для дэша “Санкей”:

- `DisableLoadData`
- `!ColoredPath`
- `EastPanel`
- `GradientPath`
- `TooltipPercent`

- `LabelsLeft`
- `!Labels`
- `!Category`
- `AllCategory`

17. Опции, доступные для дэша “Сетчатая диаграмма”:

- `DisableLoadData`
- `!ColorByCategory`
- `EastPanel`
- `!xAxisSplitLines`
- `!yAxisSplitLines`

18. Опции, доступные для дэша “Таблица с группировкой”:

- `DisableLoadData`
- `DisplayOverall`
- `FullWidth`

19. Опции, доступные для дэша “Плоская таблица”:

- `bgContain`
- `FullWidth`
- `!Sort`

20. Опции, доступные для дэша “Сводная таблица”:

- `DisableLoadData`
- `DisplayOverall`
- `FullWidth`

21. Опции, доступные для дэшэй “What-if анализ”, “HTML” и “Изображение”:

- `DisableLoadData`
- `Transparent`
- `EastPanel`

22. Опции, доступные для дэша “Управляющий дэш”:

- `DisableLoadData`
- `EastPanel`
- `PanelOpened`
- `ResetButton`
- `ResetButtonToDefault`

23. Опции, доступные для дэша “Доска”:

- `EastPanel`

24. Опции, доступные для дэша “Вкладки”:

- `Transparent`

25. Опции, доступные для дэша “Axes-selector”:

- `DisableLoadData`
- `EastPanel`

1.2.11 Секция hierarchy

Описывает иерархию измерений, например для реализации функции DrillDown

“**hierarchy**” : [],

- тип массив `STRING`;
- создание иерархии у кубов; используется для описания каскадных фильтров в управляющем дэше
- Пример: `"hierarchy": ["sex=>age=>dt", "category=>degree=>sex"]`.



По умолчанию, иерархия будет отображена внутри дэша аналогичным типом дэша. Для отображения иерархии в модальном окне горизонтальными столбцами необходимо подключить у дэша опцию `ModalDrilldown`

1.2.12 Секция onClickDataPoint

Поле `onClickDataPoint` описывает поведение клиентской части LuxmsBI при нажатии на один из указанных показателей в LPE-выражении, используемом для описания поведения.

Для изменения поведения можно пользоваться следующими функциями:

1. `setKoobFilters` - фильтрация дэшей по значению размерности при нажатии на данное значение размерности. Функция имеет три или более аргументов: `название_источника_данных.название_куба`, `название размерности` (по которой необходимо отфильтровать данные), массив с указанием условия фильтрации - первым элементом массива должен быть указан знак условия фильтрации (`=`, `!=`, `<`, `>`, `<=`, `>=`, `between`) после этого указывается размерность, по нажатию на которую, в массив подставится значение размерности. Указать можно как одну размерность для фильтрации, так и несколько: `setKoobFilters('источник.куб', 'ключ', ['=', значение], 'ключ', ['=', значение], ...)` Пример: при нажатии на сегмент дэша со значением размерности “Мужчины” фильтр для дэшей будет установлен в следующее положение:

```

1 "filters": {
2   "указанное_вторым_параметром_название_размерности": [
3     "=",
4     "Мужчины"
5   ]
6 }
```

Пример использования функции:

```

1 "onClickDataPoint": "lpe:setKoobFilters('ch.max_example', 'max_example_category',
  ['=', max_example_category])",
```

2. `setKoobFilter` - работает аналогично `setKoobFilters` с тем исключением, что принимает на вход только три аргумента, задать фильтры для нескольких размерностей разом в этой функции не получится: `setKoobFilter('источник.куб', 'ключ', ['=', значение])` В данной функции можно также указать используемую гранулярность для размерностей типа период, для этого в функцию необходимо передать параметр `data:granularity` и необходимый тип периода:

```
1 onClickDataPoint: 'lpe:setKoobFilter(null, "data:granularity","month")',
```

Возможные значения для третьего аргумента:

- `day` - для отображения календаря по дням
- `week` - для отображения календаря по неделям
- `month` - для отображения календаря по месяцам
- `quarter` - для отображения календаря по кварталам
- `year` - для отображения календаря по годам

Ниже представлен пример, для случая, если при нажатии на дэш необходимо открывать модальное окно с управляющим дэшем с установленной гранулярностью:

```
1 onClickDataPoint: 'lpe:openModal(dashlet(255));setKoobFilter(null, "data:↔granularity","month")',
```

3. `navigate` - переключение на указанный в выражении дэшборд/датасет. Функция имеет минимум два аргумента: `Элемент для перехода` - Возможные значения: `dboard` - для перехода на другой дэшбод текущего датасета, `segmentId` - для перехода на другой датасет (в случае, если в аргументах указан только переход на датасет, то переход будет осуществлен на дэшборд с номером, аналогичным изначальному) `ID элемента для перехода` - Для перехода к датасету необходимо указывать схему датасета, для дэшборда номер.

Пример использования LPE-выражения с вышеописанными функциями представлен ниже:

На изображении ниже представлен дэш “Данные” и его конфигурационный файл:

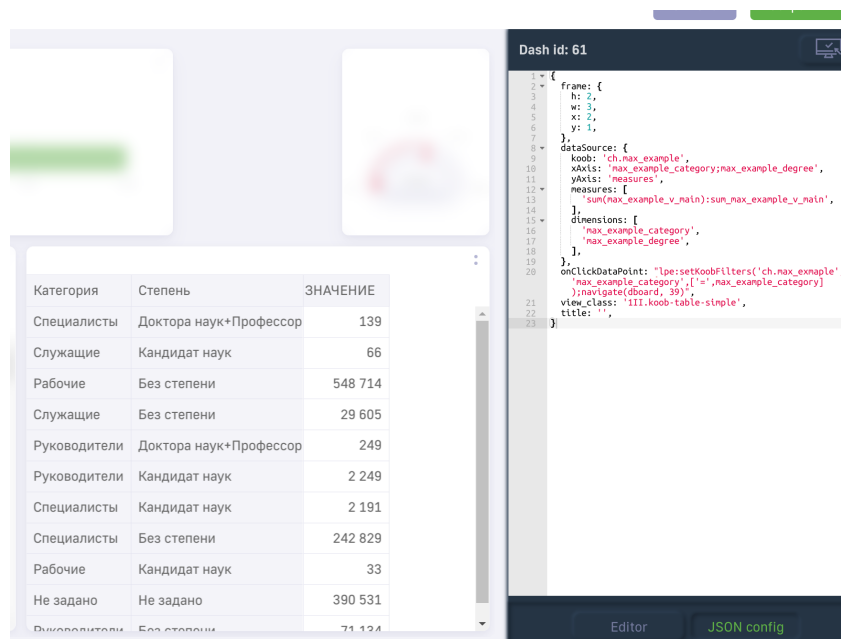


Рис. 1.50 Дэш “Данные” и конфигурационный файл

Ниже представлен сам конфигурационный файл дэша:

```

1 {
2   "frame": {
3     "h": 2,
4     "w": 3,
5     "x": 2,
6     "y": 1,
7   },
8   "dataSource": {
9     "koob": "ch.max_example",
10    "xAxis": "max_example_category;max_example_degree",
11    "yAxis": "measures",
12    "measures": [
13      "sum(max_example_v_main):sum_max_example_v_main",
14    ],
15    "dimensions": [
16      "max_example_category",
17      "max_example_degree",
18    ],
19  },
20  "onClickDataPoint": "lpe:setKoobFilters('ch.max_example',
21    'max_example_category',[ '=',max_example_category]);navigate(dboard,39)",
22  "view_class": "III.koob-table-simple",
23  "title": "",
24 }

```

В данном примере, при нажатии на ячейку таблицы, происходит переход на дэшборд 39 и на данном дэшборде все дэши фильтруются, у которых в конфигурации указан фильтр по размерности, указанной в LPE-выражении и используется указанный куб.

Существует также возможность в функции `navigate` указать 4 аргумента, для перехода к конкретному датасету на конкретный дэшборд. Пример функции представлен ниже:

```
1 navigate(segmentId, ds_demo117, dboard, 1)
```

4. `openModal` - при нажатии на точку (сегмент) дэша откроется модальное окно с дэшем, id которого указано в конфигурационном файле. Пример использования данной функциональности представлен ниже:

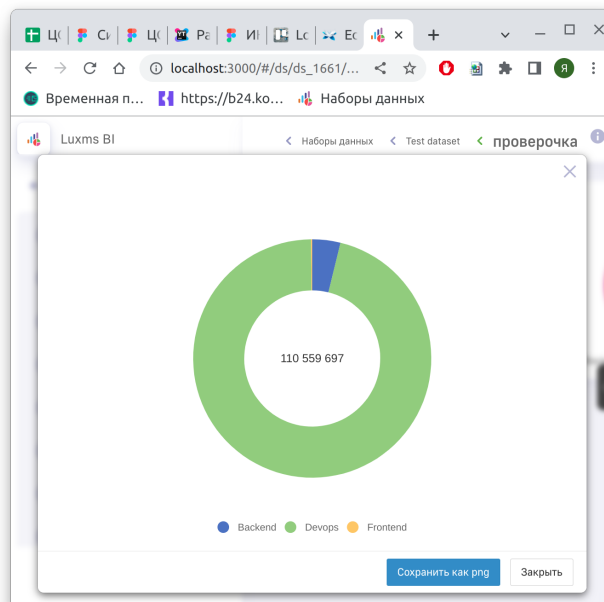


Рис. 1.51 Открытое модальное окно с дэшем “Донат”

Конфигурационный файл с примером использования функции `openModal`:

```
1 {
2   "frame": {
3     "h": 4,
4     "w": 6,
5     "x": 0,
6     "y": 0,
7   },
8   "dataSource": {
9     "koob": "my_new_excels.my_new_excels",
10    "xAxis": "department",
11    "yAxis": "measures;sex",
12    "measures": [
13      "avg(height):avg_height",
14    ],
15    "dimensions": [
16      "department",
17      "sex",
18    ],
19  },
20  "onClickDataPoint": "lpe:openModal(dashlet(210))",
21  "view_class": "1III.stacked-column",
22  "title": "",
23 }
```

Для использования `openModal` необходимо аргументом передать функцию `dashlet` и `id` необходимого для отображения дашлета (в случае на примере выбран дашлет с `id = 210`)

Если данные в модальном окне необходимо фильтровать данные в зависимости от нажатой точки, можно использовать функцию `set` с параметрами `dataSource.filters` и `filters`.

Пример:

```
1 onClickDataPoint: 'lpe:openModal(dashlet(532).set("dataSource.filters", filters))',
```

Вторым аргументом в **openModal** передается заголовок модального окна, который будет выводиться:

```
1 onClickDataPoint: 'lpe:openModal(dashlet(299).set("dataSource.filters", filters), "Количество заказов")',
```

Количество заказов



Франция США Великобритания Австралия Италия Германия Норвегия Швеция Дания Бразилия

Закрыть

Сохранить как png

Рис. 1.52 Указание заголовка для модального окна

5. `navigateUrl` - используется для перехода на внешний источник с подстановкой значения из таблицы. Пример использования данной функциональности представлен ниже:

Возраст	Категория	Σ V_main
35-50	Рабочие	225492
	Руководители	37600
	Служащие	12346
	Специалисты	103858
До 35	Рабочие	
Общий итог		1287740

```

1 {
2   frame: {
3     h: 4,
4     w: 9,
5     x: 0,
6     y: 0,
7   },
8   dataSource: {
9     koob: 'ch.max_example',
10    style: {},
11    xAxis: 'age;category',
12    yAxis: 'measures',
13    measures: [
14      'sum(v_main):sum_v_main',
15    ],
16    dimensions: [
17      'age',
18      'category',
19    ],
20  },
21  onClickDataPoint: 'lpe:navigateUrl("http://google.com/'
22    +v)",
23  view_class: '111.tableP',
24  title: ''
25 }

```

Рис. 1.53 navigateUrl для таблицы tableP

Конфигурационный файл с примером использования функции `navigateUrl`:

```

1 {
2   "frame": {
3     "h": 4,
4     "w": 9,
5     "x": 0,
6     "y": 0,
7   },
8   "dataSource": {
9     "koob": "ch.max_example",
10    "style": {},
11    "xAxis": "age;category",
12    "yAxis": "measures",
13    "measures": [
14      "sum(v_main):sum_v_main",
15    ],
16    "dimensions": [
17      "age",
18      "category",
19    ],
20  },
21  "onClickDataPoint": "lpe:navigateUrl('http://google.com/' +v)",
22  "view_class": "111.tableP",
23  "title": "",
24 }

```

Параметры, необходимые для функции `navigateUrl`:

- в кавычках указывается url для перехода
- `+v` указывается в случае необходимости подстановки значения из таблицы в url и переходе с учетом `id` из таблицы

На примере, продемонстрированном выше, в случае нажатия на ячейку со значением 37600, браузер откроет страницу по следующему адресу `http://google.com/37600`

6. `menuItem` - используется для совместного использования **hierarchy** и функций **onClickDataPoint**.

Ниже представлен пример добавления фильтрации и вызова lookup-таблицы в выпадающий список по клику:

```
1 "onClickDataPoint": "lpe:showDrilldownMenu(menuItem('Доп.пункт.меню', ↵),
  setKoobFilters('luxmsbi.orders_ful', 'country', ['=', 'country'])),attachment(1))",
```

В `menuItem` аргументами являются все команды, которые необходимо указать в списке. Данную команду можно прописать следующим образом:

```
1 "onClickDataPoint": [
2   'showDrilldownMenu',
3   [
4     'menuItem',
5     'Доп.пункт.меню',
6     "setKoobFilters('luxmsbi.orders_ful', 'country', ['=', 'country'])",
7   ],
8   [
9     'attachment',
10    1,
11  ],
12 ],
```

7. `toggleKoobFilters` - множественный выбор для фильтрации на дэше “Карта”. Работает для всех слоев.

Пример:

```
1 "onClickDataPoint": "lpe:toggleKoobFilters('luxmsbi.custom_final', 'region_id', ↵,
  ['=', 'region_id'])",
```



Для выбора цвета обводки выбранных объектов используется функция `selectedColor` в блоке `display`. Для указания ширины используйте функцию `selectedWidth`

Если необходимо выполнять какую-либо функцию исходя из определенных условий, можно использовать оператор `if`, который прописывается следующим образом:

```
1 onClickDataPoint: "lpe:if(условие, действие_если_условие_истинно, действие_если_↵,
  условие_ложно)
```

Пример:

```
1 onClickDataPoint: "lpe:if(dim>0,setKoobFilters('cube', 'dim1', ['=', dim1]),↵,
  null)",
```

Если необходимо выполнять функцию в зависимости от того, какая размерность лежит на оси, можно использовать в условии функцию `get` с первым аргументом `"id"` вторым аргументом указывается ось, которую необходимо проверять (`x` или `y`).

Пример:

```
1 onClickDataPoint: "lpe:if(get(x,'id')=dim1,setKoobFilters('cube', 'dim1', ['=',dim1]),null)",
```

8. `dataReload` - перезапрос данных у указанного куба в аргументе

```
1 onClickDataPoint: 'lpe:dataReload("ch.koob_example")',
```

После нажатия дэши, использующие данные куб будут перезагружены и перезапросят данные.

1.2.13 Секция children

Позволяет задать дочерние дэши

“children” :[]

- Используется в типах дэша: **board**, **tabs**
- Принимает массив ВИЗЕЛЕЙ
- Пример:

```
1 "children": [{
2   "view_class": "text"
3   "frame": {},
4   "display": {},
5   "options": [],
6   "dataSource": {}
7   }, {...},{...}]
```



В случае необходимости отображения меню дэша у дочерних элементов, у них во `view_class` необходимо прописать `dashlet/название_дэшлета`

1.2.14 Опции для визеля Pie

Вывод значений для дэша “Пирог” можно реализовать различными способами, для этого реализованы следующие опции (ввод значения производится в массиве `options`):

```
1 "options": [
2   "DisplayAllBadges",
3   ...]
```

1. `DisplayAllBadges` - Вывод наименования показателя, его абсолютное и относительное значение
2. `DisplayBadgesPercent` - Выводится только относительное значение (в процентах)
3. `DisplayBadgesValue` - Выводится только абсолютное значение

В случае необходимости вывода абсолютного и относительного значений без наименования показателя, необходимо указать опции `DisplayBadgesPercent` и `DisplayBadgesValue`.

1.2.15 Опции для визуеля koob-table-simple (Плоская таблица)

Для визуеля koob-table-simple существует возможность указывать стили CSS для столбцов таблицы. Реализованы следующие опции:

1. **color** - указание цвета значений в столбце (указывается код цвета #HEX)
2. **fontStyle** - указание стилизации отображения значений столбца (указание значения **italic** отображает текст курсивом)
3. **fontSize** - указание размера шрифта значений в столбце (указываются относительные значения в процентах относительно значения по умолчанию)
4. **fontWeight** - установка насыщенности шрифта значений (указание значения **bold** отображает текст полужирным начертанием)
5. **textDecoration** - добавление оформления текста с использованием линии (указание значения **underline** подчеркивает текст)
6. **backgroundColor** - указание цвета фона значений в столбце (указывается код цвета #HEX)
7. **whiteSpace** - опция, для переноса текста в ячейке, в случае, когда он не помещается в ячейке (указать значение 'wrap')
8. **wordBreak** - указание того, где будет установлен перевод на новую строку в случае превышения текстом границ блока (указать значение **break-word**). Не работает без указанной опции **whiteSpace**
9. **formula** - указывается формула, по которой будет выводиться итог столбца. Требуется указать агрегат **measure** - как указано в массиве **measures**. (Работает только с включенной опцией **DisplayOverall**. при включенной опции выводится сумма значений по столбцу.)
10. **textAlign** - ориентация значений внутри ячейки. Доступные значения: center, left, right.
11. **footerStyle** - опция для настройки нижней строчки таблицы, в которой указывается Общий итог.
12. **Pin** - опция для закрепления столбца (указывается в массиве **options**)
13. **value** - опция для редактирования ячейки плоской таблицы, с ее помощью можно в ячейку добавить ссылку, а так же svg картинку. Кроме того есть возможность добавить условие через lpe-выражение. В примере ниже используются функции **bold(v)** - для выделения текста жирным, **italic(v)** - для выделения значения курсивом, **br()** - для переноса строки, переменная **v** (либо **value**, а также можно указывать название **measure** или **dimension**) - значение в ячейке таблицы, **nbsp** - для пробела, **aHref()** - функция для указания ссылки. Svg файл необходимо добавить в ресурсы текущего атласа.
14. **width** - указание ширины для столбца в таблице.
15. **height** - указание высоты для столбца в таблице.

Для вышеописанных опций присутствует также возможность использования LPE-выражений для различного отображения ячеек в зависимости от условий.

Пример использования данных ключей конфигурации представлен ниже:

```
1 "style": {  
2   "measures": {  
3     "b": {  
4       "color": "#a6c497",  
5       "minWidth": 20,
```

```

6      "backgroundColor": "#771111",
7    },
8    "c": {
9      "color": "#4ab6e8",
10     "fontSize": "lpe:if(c>400000, '120%', '80%')",
11     "minWidth": 10,
12     "fontStyle": "lpe:if(c>400000, 'italic', '')",
13     "fontWeight": "lpe:if(c>400000, 'bold', 'normal')",
14     "textDecoration": "lpe:if(c>400000, 'underline', '80%')",
15     "backgroundColor": "lpe:if(c > 400000, '#ffaabb', 'transparent')",
16     "formula": "sum(usr_amt)/sum(plan_usr_amt)*100:sum_sum_1",
17     "textAlign": "center"
18     "options": [
19       "Pin"
20     ]
21   },
22   "d": {
23     value: 'lpe:if(v>2, bold(v) + br() +svg("myTest"),  v + nbsp
+ svg("myTest_1"))',
24   },
25   "e": {
26     value: 'lpe: if(value="123", aHref("http://localhost:3000/",
"myLink"), value)',
27   }
28 },
29 },

```

В плоской таблице предусмотрена возможность визуальной группировки столбцов. Для этого используется поле **groupColumns**. Данный объект является массивом, содержимое в котором необходимо прописывать по следующей схеме:

```

1 groupColumns: [
2   'Название группы:количество столбцов для группировки',
3   ...,
4 ],

```

Первый элемент массива **groupColumns** объединит первые n-столбцов с указанной подписью.

Пример:

```

1 groupColumns: [
2   'Группа 1:2',
3   'Группа 2:1',
4 ],

```

На дэшборде плоская таблица с вышепредставленным содержимым будет отображена следующим образом:

Группа 1		Группа 2	
Агент	Клиент	Запрашиваемая сум...	Процент от добавочн...
Kontragent596	Klient45399	0	0
Kontragent979	Klient25400	0	0
Kontragent617	Klient36497	0	0
Kontragent979	Klient41574	0	0

Рис. 1.54 Плоская таблица с группировкой столбцов

В примере выше “Группа 1” объединяет 2 столбца, так как в массиве указано объединить 2 столбца. По аналогичной логике в “Группа 2” входит 1 столбец. Все последующие столбцы будут отображены без группы.

1.2.16 Опции для виджета tableP (Таблица с группировкой)



Для таблицы `tableP` стилизация строк реализована аналогично стилизации столбцов

1. `color` - Указание цвета значений в столбце (указывается код цвета #HEX)
2. `fontStyle` - Указание стилизации отображения значений столбца (указание значения `italic` отображает текст курсивом)
3. `fontSize` - Указание размера шрифта значений в столбце (указываются относительные значения в процентах относительно значения по умолчанию)
4. `fontWeight` - Установка насыщенности шрифта значений (указание значения `bold` отображает текст полужирным начертанием)
5. `textDecoration` - Добавление оформления текста с использованием линии (указание значения `underline` подчеркивает текст)
6. `backgroundColor` - Указание цвета фона значений в столбце (указывается код цвета #HEX)
7. `width` - Указание значения ширины столбца в относительных единицах
8. `whiteSpace` - Опция, для переноса текста в ячейке, в случае, когда он не помещается в ячейке (указать значение `'wrap'`)
9. `textAlign` - ориентация значений внутри ячейки. Доступные значения: `center`, `left`, `right`.
10. `height` - Указание значения высоты столбца в относительных единицах
11. `transform` - указание функции манипуляции отображения данных. Работает аналогично CSS-стилю `transform`. Пример: `transform: "rotate(90deg)"`. Работает только для заголовков (`headerStyle`)

Для стилизации заголовков таблицы, внутри объекта стилизации факта/размерности необходимо указать объект `headerStyle` и внутри него прописывать необходимые стили. Это может понадобиться для отцентровки заголовка столбца (`textAlign: "center"`). Ниже представлен пример конфигурации для стилизации заголовка таблицы:

Отмена Сохранить

1	2	3	4	5
Возраст	Категория	Образование	Σ V_main	
2	Старше 50	Рабочие	Высшее	7 024
3	Не задано	Не задано	Не задано	390 531
4	Старше 50	Служащие	Иное	2 536
5	Старше 50	Рабочие	Иное	68 703
6	35-50	Рабочие	Высшее	29 276
7	35-50	Специалисты	Высшее	77 733
8	До 35	Служащие	Среднее профессиональное	2 856
	До 35	Рабочие	Среднее профессиональное	52 832
	35-50	Руководители	Иное	1 555
	До 35	Рабочие	Иное	135 373
	Старше 50	Руководители	Среднее профессиональное	3 288
	Старше 50	Служащие	Среднее профессиональное	2 945
	35-50	Служащие	Высшее	4 223
	Старше 50	Специалисты	Высшее	20 558
	До 35	Руководители	Среднее профессиональное	3 032
	До 35	Специалисты	Высшее	73 035
	35-50	Рабочие	Среднее профессиональное	50 140
	Старше 50	Руководители	Высшее	11 708

Dash id: -1000000

```

1 {
2   dataSource: {
3     koob: 'ch_max_example',
4     ykx: 'measures',
5     measures: [
6       'sum(V_main):sum_v_main',
7     ],
8   },
9   style: {
10    education: {
11      minWidth: 500,
12      headerStyle: {
13        textAlign: 'center',
14      },
15    },
16  },
17  dimensions: [
18    'age',
19    'category',
20    'education',
21  ],
22  xkx: 'age;category;education',
23  frame: {
24    x: 0,
25    y: 0,
26    w: 0,
27    h: 0,
28  },
29  view_class: 'koob-table-simple',
30  title: '',
31 }

```

Рис. 1.55 Стилизация заголовка у таблицы koob-table-simple

Указание ширины для столбцов, отложенных на оси X в tableP производится в headerStyle:

```

1 "degree": {
2   "headerStyle": {
3     "width": 300,
4   },
5 },

```

Степень	Опыт	Σ V_main				NV_main
		Рабочие	Руководители	Служащие	Специалисты	
Без степени	1-3		83270	12460	871	22019
	3-5		43791	8014	624	7958
	5-10		107213	12469	622	24089
	Более 10		186132	11492	714	36415
	До года		32244	4832	607	6563
Доктора наук*Профессор	1-3					36
	3-5			36		31
Общий итог			452683	50901	3438	98360

```

1 {
2   frame: {
3     h: 4,
4     w: 12,
5     x: 0,
6     y: 0,
7   },
8   display: {
9     title: 'xch.mbc',
10    fontFamily: 'sans-serif',
11    customValue: '',
12  },
13  options: [],
14  dataSource: {
15    koob: 'ch.max.example',
16  },
17  style: {
18    degree: {
19      headerStyle: {
20        minWidth: 300,
21      },
22    },
23    category: {
24      headerStyle: {
25        minWidth: 300,
26      },
27    },
28  },
29  xaxis: 'degree:experience',
30  yaxis: 'measures:category',
31  filters: {
32    degree: true,
33  },
34  measures: [
35    'sum(v_main):sum_v_main',
36    'count(v_main):count_v_main',
37  ],
38  dimensions: [
39    'category',
40    'degree',
41    'experience',
42  ],
43  view_class: 'III.tableP',
44  title: 'zzz1',
45 }

```

Рис. 1.56 Указание ширины для столбца с размерностями

1.2.16.1 Редактирование данных через koob-table-simple

Таблицу koob-table-simple также можно использовать для редактирования значений в базе данных. Для этого необходимо совершить ряд действий:

1. Открыть список кубов в разделе “Данные”
2. Открыть режим редактирования куба, в котором необходимо проводить редактирование значений
3. Открыть вкладку “Config” и прописать в конфигурацию следующее:

```

1 "table": "maxexample", // таблица и схема, куда пишутся значения
2 "schema": "public",
3 "primary_key": [ "id" ] // список ключей этой таблицы

```

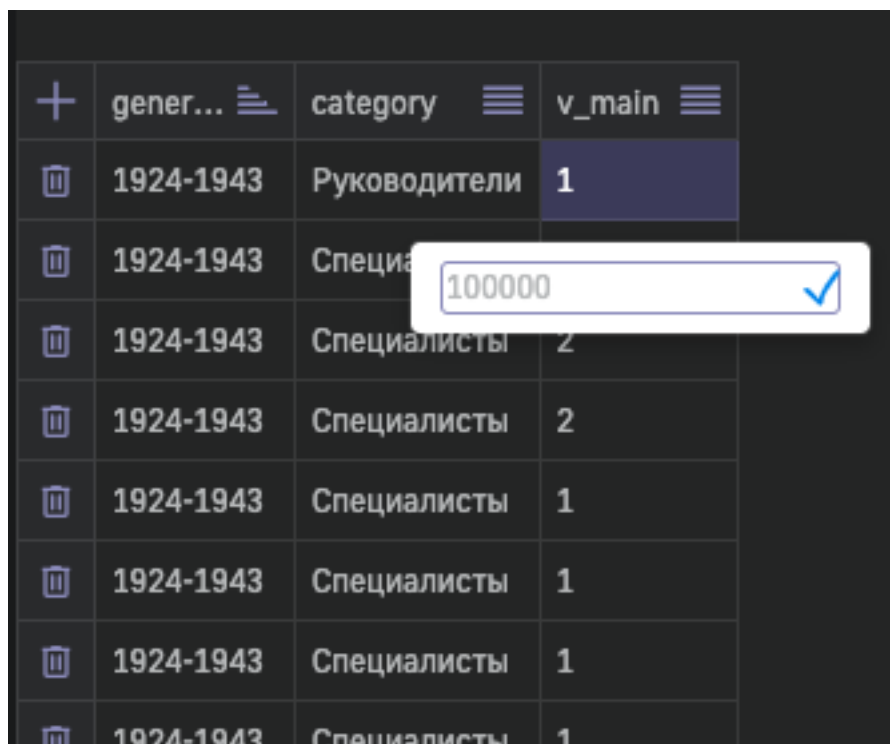
4. Сохранить изменения
5. Перейти в клиентскую часть и в конфигурации дэша **koob-table-simple** прописать:

```

1 "onClickDataPoint": 'edit'

```

Для поля **koob** должен быть указан куб, конфигурация которого была отредактирована на шаге 3. После этого вы можете редактировать значения в таблице нажав на ячейку, которую необходимо редактировать. Ниже представлен пример редактирования таблицы:



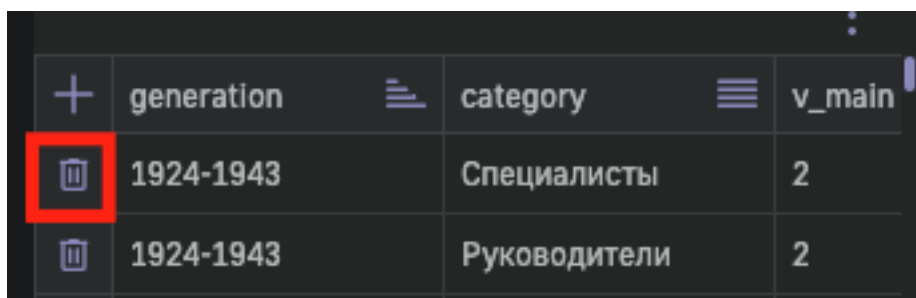
+	gener...	category	v_main
🗑	1924-1943	Руководители	1
🗑	1924-1943	Специа	100000
🗑	1924-1943	Специалисты	2
🗑	1924-1943	Специалисты	2
🗑	1924-1943	Специалисты	1
🗑	1924-1943	Специалисты	1
🗑	1924-1943	Специалисты	1
🗑	1924-1943	Специалисты	1

Рис. 1.57 Редактирование значения в плоской таблице

{width=6.69306in height=3.30486in}

При успешном обновлении таблицы в ответе запроса можно увидеть {"count" : 1}.

С помощью клика по корзине можно удалить из источника всю соответствующую строку.



+	generation	category	v_main
🗑	1924-1943	Специалисты	2
🗑	1924-1943	Руководители	2

Рис. 1.58 Удаление значения в плоской таблице

Также можно создавать сточки, для этого следует кликнуть по "+". После чего появляется окно для ввода значений полей в таблице.

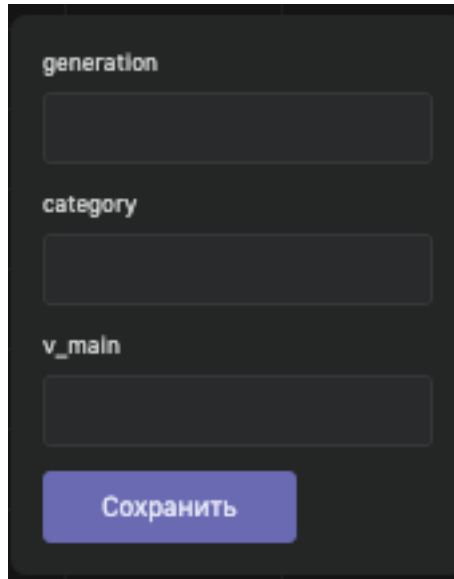


Рис. 1.59 Добавление значения в плоской таблице

Важно помнить, что первичный ключ должен быть корректным.



Помимо этого, можно дать доступ на редактирование только одному из столбцов. Для этого, необходимо указать `"onClickDataPoint": 'edit'` в блоке `style` для конкретного факта/размерности.



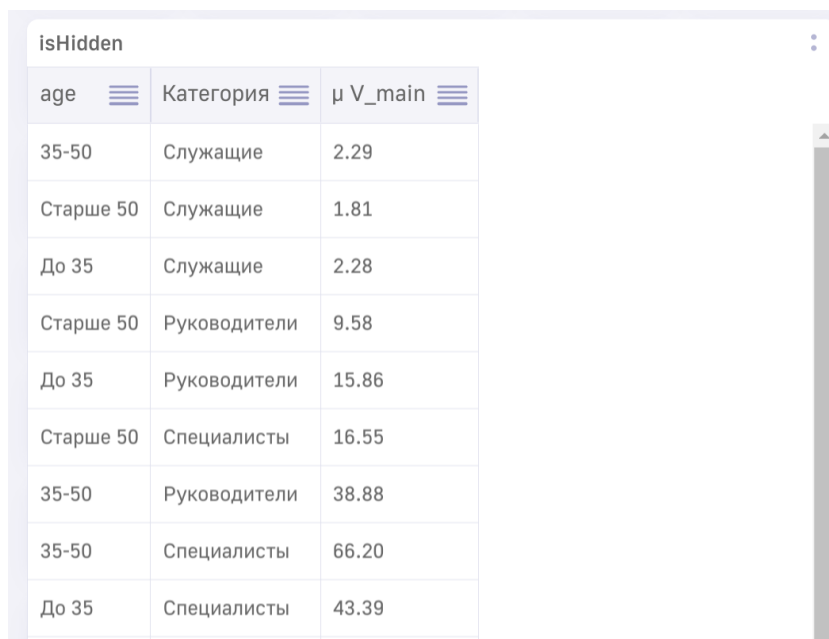
Для корректной работы, в таблице и кубе должен присутствовать столбец `id`, который будет являться `primary_key`



Редактирование значений корректно работает только в случае, если факт отображен без агрегации

Также существует возможность скрытия одного из столбцов в `koob-table-simple`. Для этого необходимо в его стилях прописать `isHidden: true`. Пример:

```
1 style: {  
2   measures: {  
3     sum_v_main: {  
4       isHidden: true,  
5     },  
6   },  
7 },
```



age	Категория	μ V_main
35-50	Служащие	2.29
Старше 50	Служащие	1.81
До 35	Служащие	2.28
Старше 50	Руководители	9.58
До 35	Руководители	15.86
Старше 50	Специалисты	16.55
35-50	Руководители	38.88
35-50	Специалисты	66.20
До 35	Специалисты	43.39

Рис. 1.60 Скрытие столбца в таблице koob-table-simple

Также существует возможность использования LPE-выражений для указания условия скрытия столбца:

```
1 isHidden: 'lpe:if(100>10,true, false)',
```

В данном условии можно также использовать как значения факта (`value`), значение соседнего столбца (необходимо для этого указать его название), функцию `getFilters` описанную выше, в секции `title`.

1.2.17 Конфигурационные опции для виджета “Значение”

По умолчанию в дэше “Значение” сравнивается 2 значения одного факта по двум размерностям, которые пришли последними из базы.

Также с помощью дэша “Значение” можно сравнивать 2 факта с разбивкой и без по dimensions. Для этого в конструкторе требуется перенести Факты в поле “Категории” или в конфиге внутри dataSource добавить настройку xAxis: ‘measures’.

Для виджета “значение” (`label`) существует ряд конфигурационных опций, работающих только для данного виджета. Ниже представлен список опций с возможными значениями:

- `title` - указание заголовка для дэша. Заголовок указывается над значением, может использоваться для описания значения. Ниже представлено изображение примера работы опции `title`

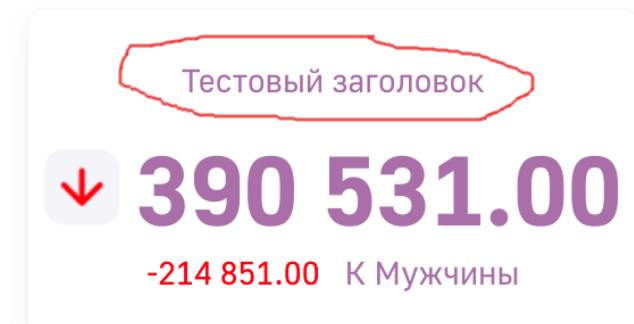


Рис. 1.61 Стилизация дэша “Значение”

В заголовке дэша “Значение” можно также пользоваться LPE-выражением и выводить различные заголовки в зависимости от установленных фильтров (функция `getFilters` описанна выше, в секции `title`).

```
1 title: 'lpe:"Количество " + if(getFilter("sex") && getFilter("sex").1 = "Женщины", "гинекологов", "урологов") + " (чел)"',
```

- `bgColor` - Указание цвета фона дэша (указывается код цвета #HEX)
- `color` - Указание цвета значений в столбце (указывается код цвета #HEX)
- `fontFamily` - Указания конкретного шрифта для дэша. Существует возможность выбрать из следующих вариантов шрифта:
 - Golos UI
 - Arial
 - Times New Roman
 - Helvetica
 - Courier New
 - Courier
 - Verdana
 - Georgia
 - Garamond
 - Bookman
 - Trebuchet
 - Tahoma
 - Arial Black
 - Comic Sans MS
 - Impact
- `fontSize` - указание размера шрифта в относительных единицах. Указывается типом значения NUMBER
- `customValue` - Данная опция используется для указания прописанного текста в значении данного ключа. При установке данного ключа данные не отображаются, отображается только введенный текст в значение `customValue`

1.2.18 Конфигурация lookup-таблицы

1.2.18.1 Настройка ширины столбцов

В конфигурации lookup-таблицы существует возможность использования определенных стилей CSS для оформления отображения столбцов таблицы. Ниже приведены примеры, в которых продемонстрированы ключи для работы lookup-таблицы:

1. `width` - установка ширины столбца в lookup-таблице
2. `white-space` - управление обработкой пробельных символов внутри элемента. Возможные варианты значений:
 - `normal` - Последовательности пробелов объединяются в один пробел. Символы новой строки в источнике обрабатываются, как отдельный пробел. Применение данного значения при необходимости разбивает строки для того, чтобы заполнить строчные боксы.
 - `nowrap` - Объединяет последовательности пробелов в один пробел, как значение `normal`, но не переносит строки (оборачивание текста) внутри текста.
 - `pre` - Последовательности пробелов сохраняются так, как они указаны в источнике.
 - `pre-line` - Последовательности пробелов объединяются в один пробел.

2 Примеры конфигурации дэшей в LuxmsBI

После создания дэша на дэшборде с использованием в административной панели раздела “Дэшборд” или с использованием self-service конструктора LuxmsBI на экране будет отображен созданный дэш. Ниже представлен вариант отображения дэша и его файл конфигурации после создания:

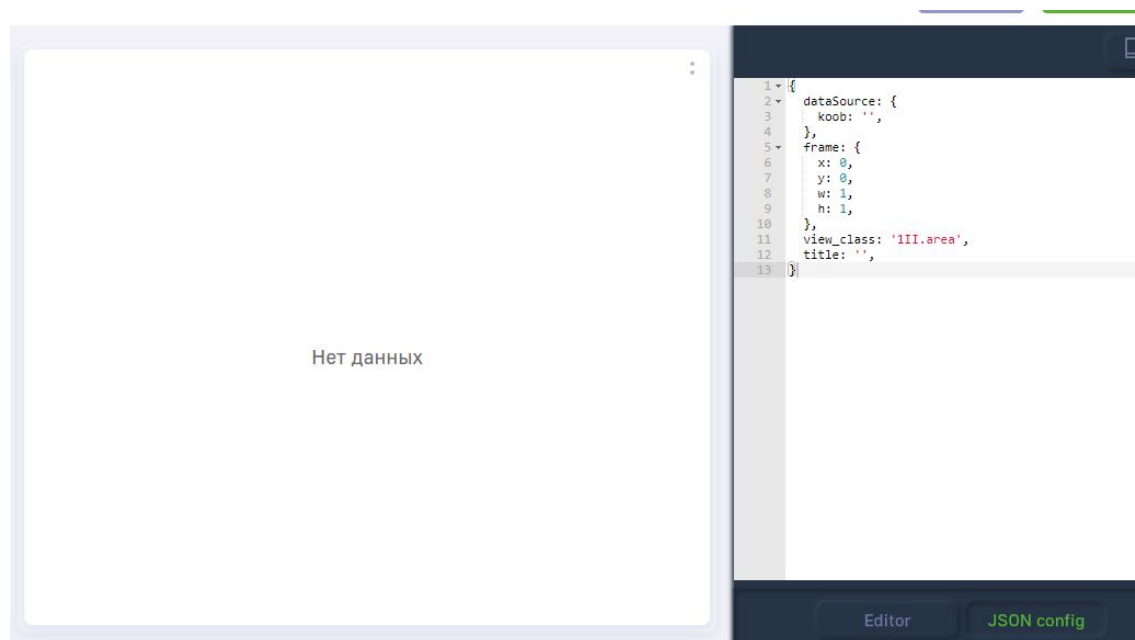
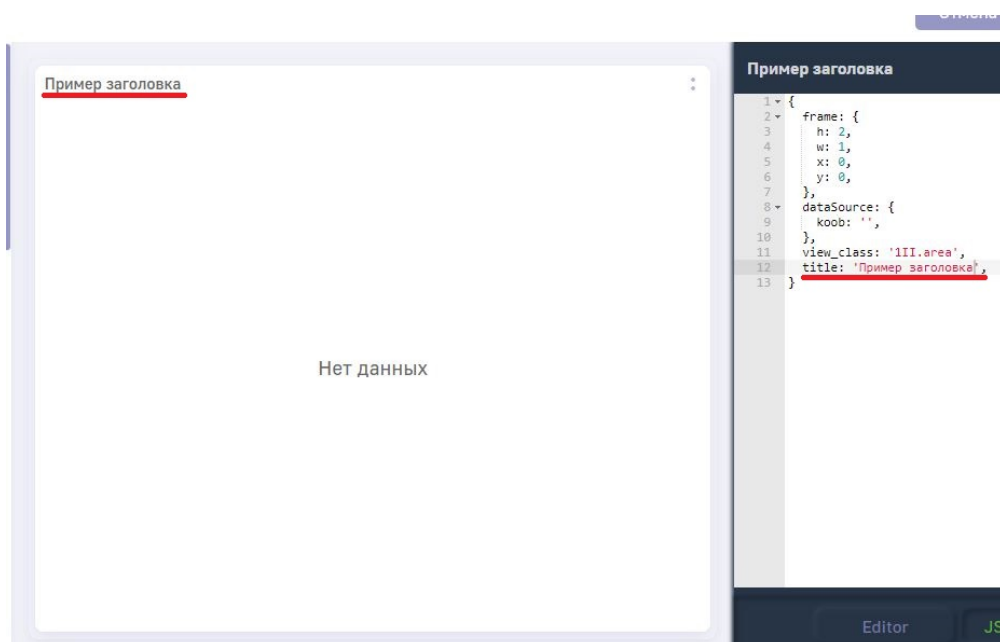


Рис. 2.1 Дэш после создания

2.1 Поле title

В поле `title` указывается заголовок дэша. Заголовок указывается в кавычках. Ниже представлен пример использования `title`:

Рис. 2.2 Поле `title`

2.2 Поле `view_class`

Поле `view_class` используется для отображения выбора варианта отображения данных (визеля). После выбора дэша и переносе его на дэшборд используя административную панель / конструктор self-service существует возможность заменить вариант отображения, заменив значение `view_class`. Список доступных ключей представлен ниже:

1. `111.bar` – Столбики горизонтальные
2. `111.line` – Линия
3. `111.column` – Столбики вертикальные
4. `111.stacked-area` – Области-Штабели
5. `111.stacked-bar` – Штабели горизонтальные
6. `111.stacked-column` – Штабели вертикальные
7. `111.scatter` – Точки
8. `111.spline` – Сплайн (Плавная линия)
9. `111.waterfall` – Водопад
10. `111.correlation` – Пузырьки
11. `111.gauge` – Спидометр
12. `111.semicircle` – Спидометр полукруглый
13. `111.thermometer` – Термометр
14. `111.pie` – Пирог
15. `111.bublik` – Бублик
16. `111.scales` – Весы
17. `111.funnel` – Воронка
18. `111.radar` – Радар
19. `111.label` – Значение
20. `111.treemap` – Древовидная карта

21. `III.sankey` – Санкей
22. `III.abc` – ABC анализ
23. `III.grid` – Сетчатая диаграмма
24. `III.tableP` – Таблица
25. `III.list` – Список
26. `whatif` – What-if анализ
27. `map` – Карта
28. `external` – Внешний
29. `internal` – Внутренний
30. `image` – Изображение
31. `html` – HTML
32. `koob-table-simple` – Плоская таблица
33. `pivot/table` – Сводная таблица
34. `VizelKoobControl` – Управляющий дэш
35. `plan` – Схема
36. `III.board` – Доска
37. `III.tabs` – Вкладки
38. `histogram` – Гистограмма
39. `boxplot` – Ящик с усами

2.3 Секция frame

Секция `frame` используется для указания местоположения и размеров конфигурируемого дэша. По умолчанию, созданные дэши занимают все доступное пространство для отображения. Для отображения дэша на экране используются следующие ключи: 1. `x` - расположение дэша на дэшборде по оси X. При увеличении значения для ключа `x` дэш сдвигается вправо на дэшборде. На изображении ниже представлен пример дэша, с указанным значением для 1:

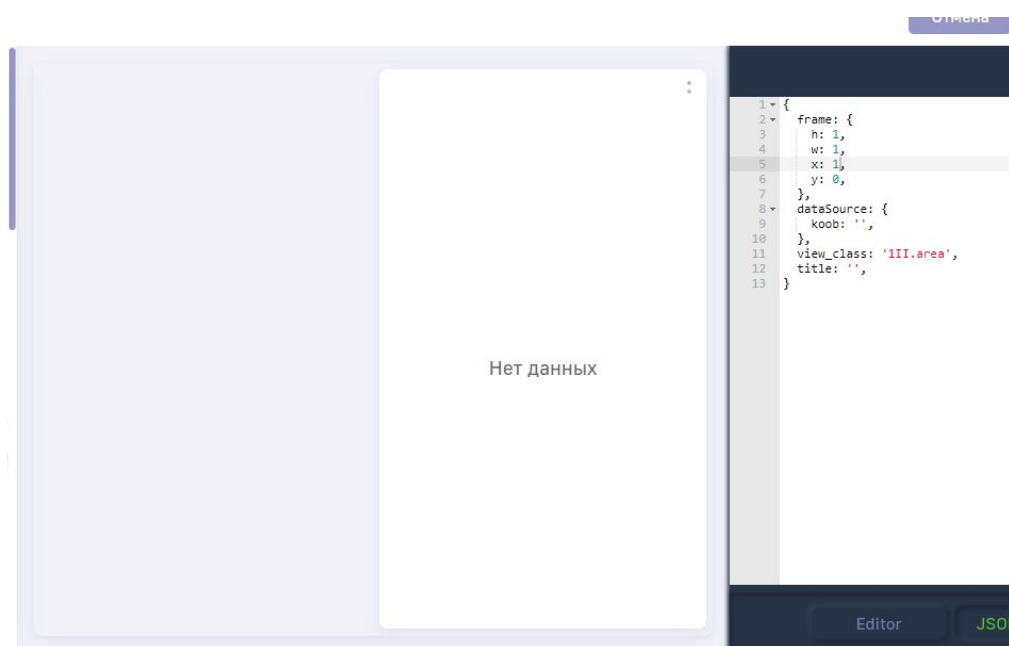


Рис. 2.3 Изменение расположения дэша по оси X

Дэш сдвинут таким образом, что в левой части дэшборда поместится дэш со значениями $w=1/h=1$ и значением $x=0$. Также можно указывать значения с числами после запятой, для более тонкой настройки расположения дэша. Ниже представлен пример отображения дэша со значением $x=3.5$

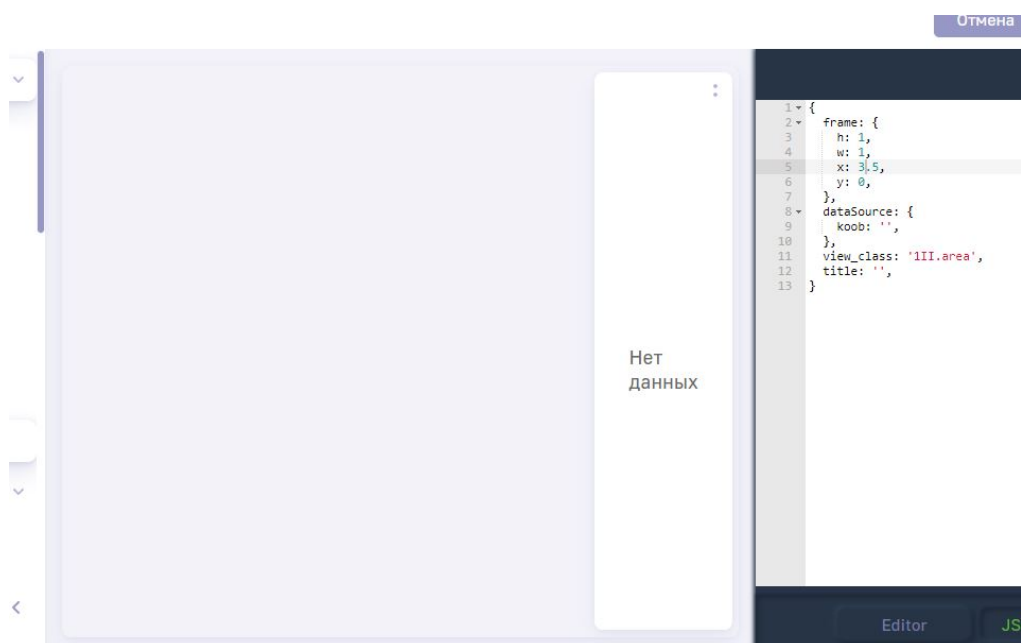


Рис. 2.4 Изменение расположения дэша по оси X

Как можно заметить, в левой пустой части дэшборда, можно разместить три дэша со значениями $w=1$ и один дэш со значением ширины $w=0.5$. Аналогичная логика работы при изменении расположения дэша по оси Y. Ниже представлены изображения для $y=1$:

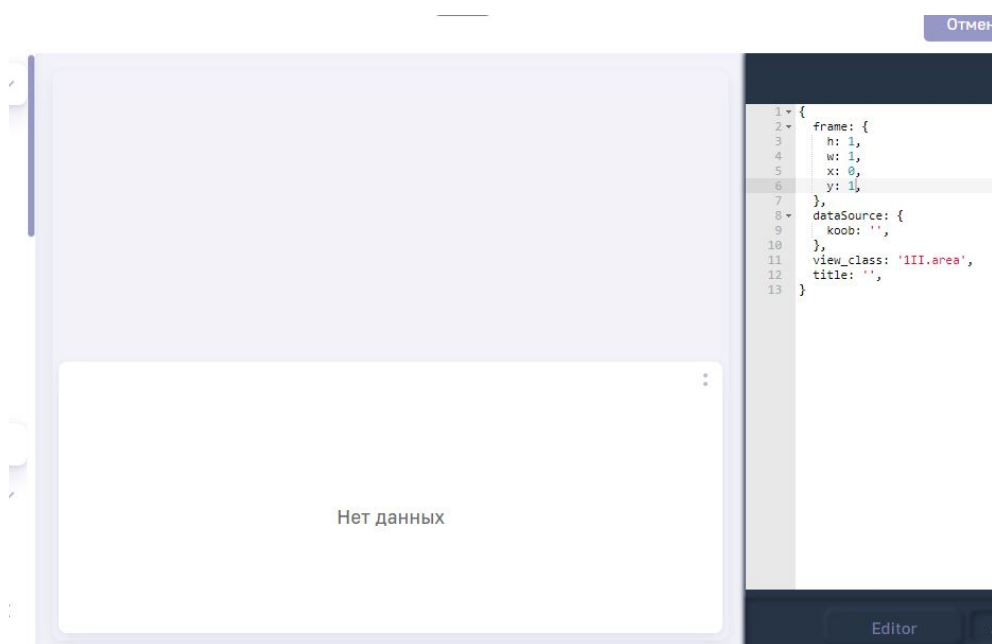


Рис. 2.5 Изменение расположения дэша по оси Y

и $y=3.5$ соответственно:

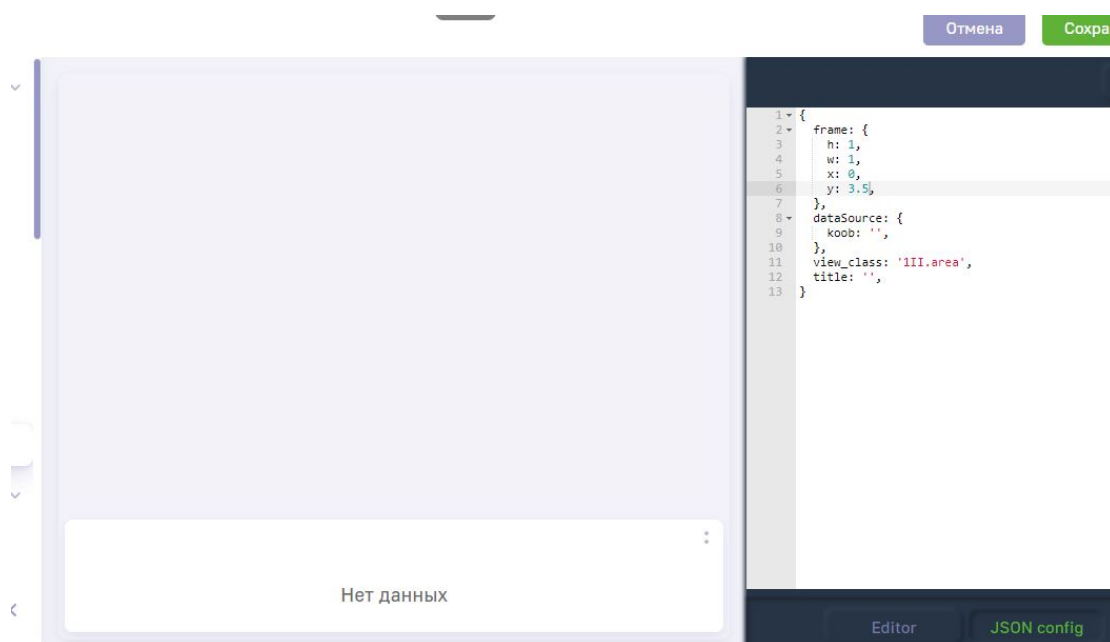


Рис. 2.6 Дэш, с указанной координатой Y

Логика аналогичная как и в случае изменения x - в верхней пустой части дэшборда, можно разместить три дэша со значениями $h=1$ и один дэш со значением ширины $h=0.5$, либо один дэш со значением $h=2$ и один дэш со значением ширины $h=1.5$, ниже представлен пример отображения трех дэшей на дэшборде с вышеописанными примерами:

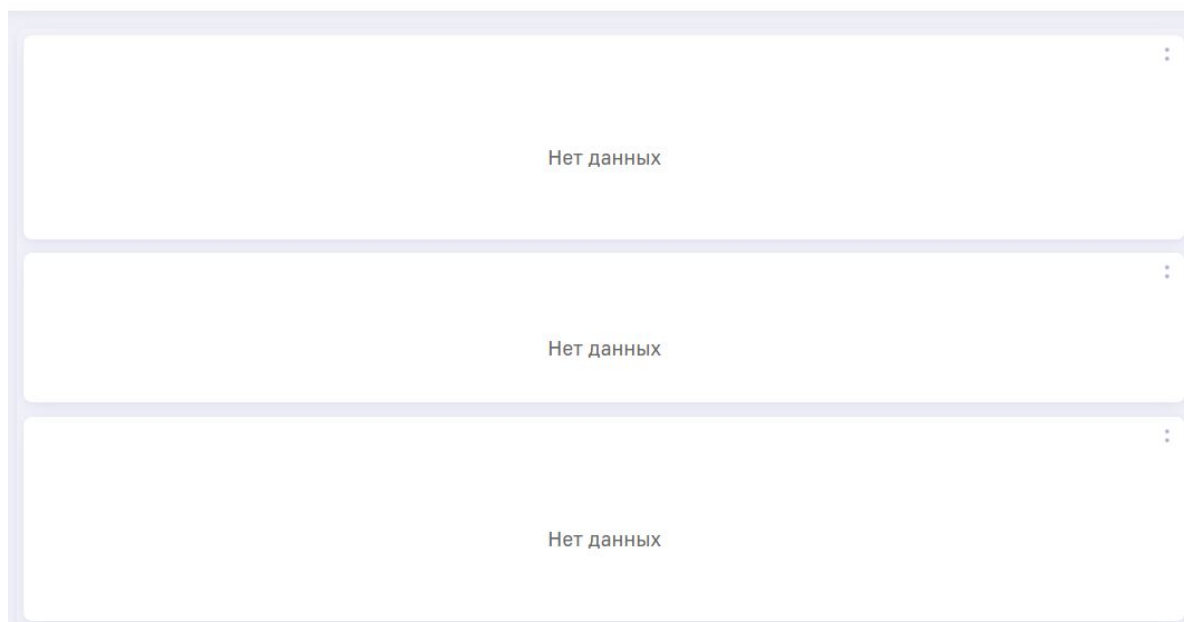


Рис. 2.7 Три дэша, расположенные по вертикали

Секции frame в файле-конфигурации для дэшей, представленных на изображении выше:

- Дэш 1:

```
1 "frame": {  
2   "h": 2,  
3   "w": 1,  
4   "x": 0,  
5   "y": 0,  
6 }
```

- Дэш 2:

```
1 "frame": {  
2   "h": 1.5,  
3   "w": 1,  
4   "x": 0,  
5   "y": 2,  
6 }
```

- Дэш 3:

```
1 "frame": {  
2   "h": 2,  
3   "w": 1,  
4   "x": 0,  
5   "y": 3.5,  
6 }
```

Используя секцию `frame` и изменяя вышеуказанные показатели в относительных единицах, вы сможете отобразить дэши на дэшборде в любом удобном для вас варианте.

`frame` можно также указать и в строковом варианте:

```
1 frame: "(x1,y1),(x2,y2)"
```

где $x1 = x$ $y1 = y$ $x2 = x + w$ $y2 = y + h$

2.4 Секция `dataSource`

Секция `dataSource` используется для указания источника данных, куба, редактирования варианта отображения данных. Ниже будут представлены подробности работы с разделом.

2.4.1 Управление данными

Рассмотрим конфигурацию дэша, только что созданную на дэшборде с использованием административной панели / конструктора:

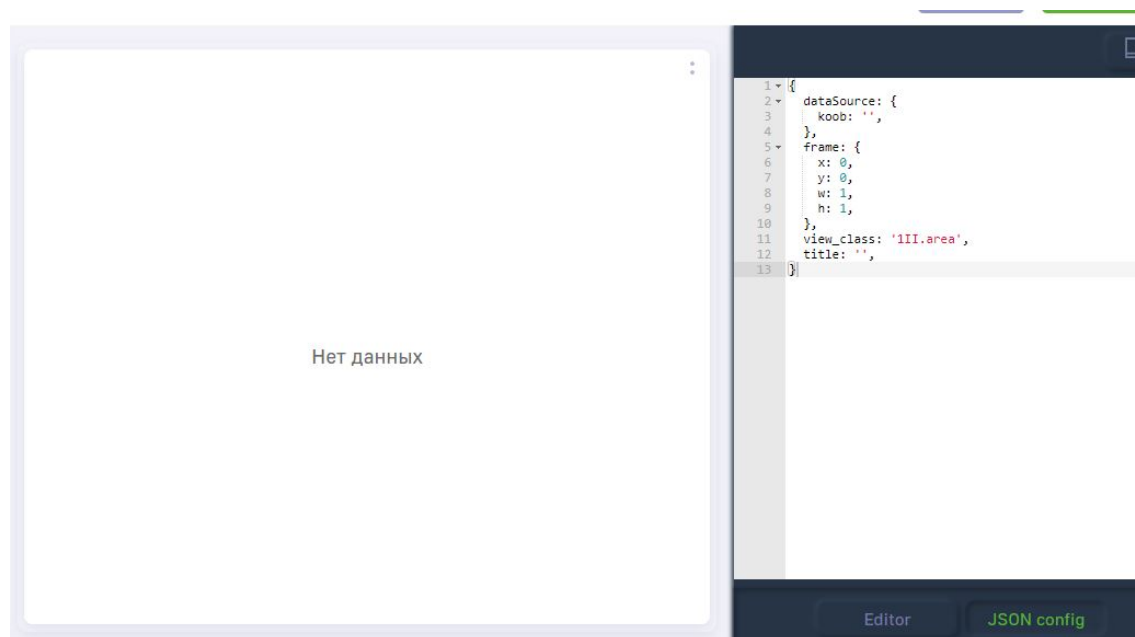


Рис. 2.8 Пустой дэш

В секции `dataSource` автоматически вставлен ключ `koob`, который предназначен для выбора источника данных и куба.

Для указания куба и отображения данных в дэше необходимо в поле `koob` прописать следующее значение:

```
1 "koob": "название_источника.название_куба"
```



Внимание! Для корректной работы LuxmsBI, используйте для названия кубов и источников данных латинские буквы, цифры и нижнее подчеркивание (но не первым символом).

Пример заполнения данного поля представлен ниже:

```
1 "koob": 'clickHouse.test2'
```

где `clickHouse` - название источника данных, `test2` - название куба

После этого, в секции `dataSource` необходимо указать факты (`measures`) и размерности (`dimensions`). Для этого необходимо прописать в массив `measures` - факты следующим образом:

```
1 "measures": [
2   "агрегационная_функция(показатель):название",
3   ...
4 ]
```

Аналогичным образом необходимо прописать и список размерностей:

```
1 "dimensions": [
```

```

2  "название_размерности",
3  ...
4  ]

```

Ниже представлен список доступных агрегационных функций для фактов:

1. sum
2. avg
3. count
4. min
5. max
6. var_pop
7. var_samp
8. stddev_samp
9. mode
10. stddev_pop
11. median

Над фактами можно проводить различные арифметические операции, ниже представлен пример подсчета среднего значения факта:

```

1  "measures": [
2    'sum(v_main)/count(v_main):avg_v_main'
3  ],

```

Ниже представлен пример заполнения массивов `measures` и `dimensions`:

```

1  "dataSource": {
2    "koob": "ch.test2",
3    "measures": [
4      "sum(v_main):sum_v_main",
5      "count(v_main):count_v_main",
6    ],
7    "dimensions": [
8      "degree",
9      "education",
10     "experience",
11     "age",
12   ],
13 },

```

Для отображения выбранных данных из куба, необходимо их отложить на оси X и Y. Для этого используются поля `xAxis` и `yAxis`. Все факты всегда отображаются на одной оси, поэтому для них используется обобщенный заголовок `measures`. Размерности можно отображать на различных осях.



В случае работы с одномерными дэшами (радар, пирог, донат) все данные для отображения необходимо перечислять на оси Y (`yAxis`)

Ниже представлены варианты отображения данных в зависимости от расположения на осях:

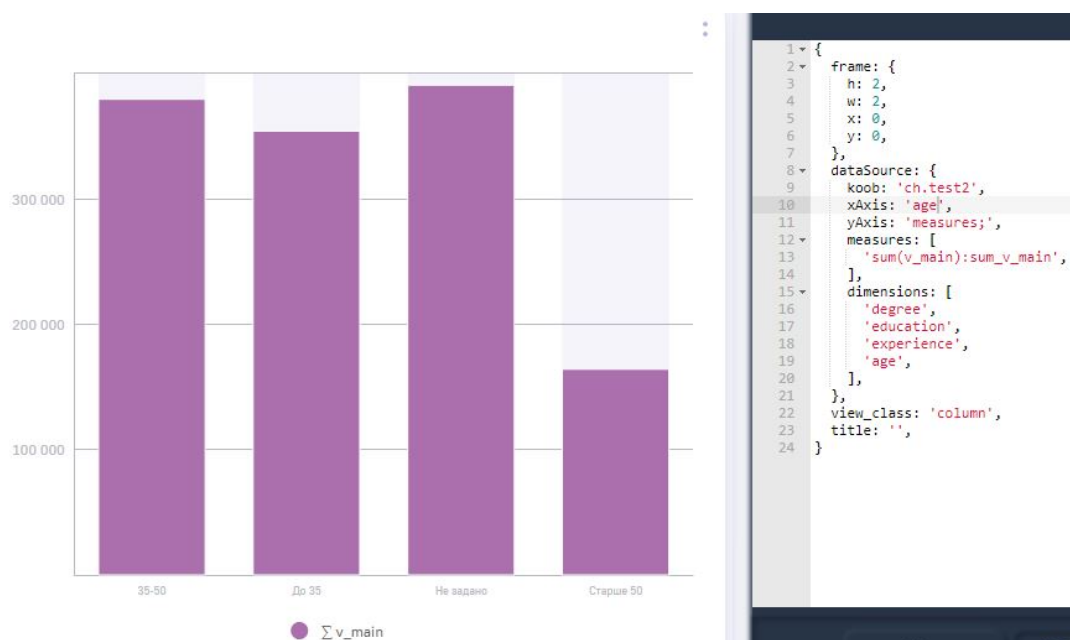


Рис. 2.9 Дэш, с отложенными размерностями на оси X

В случае, если размерность, будет отложена на оси X, то данные по этой размерности будут отображаться как одним цветом.

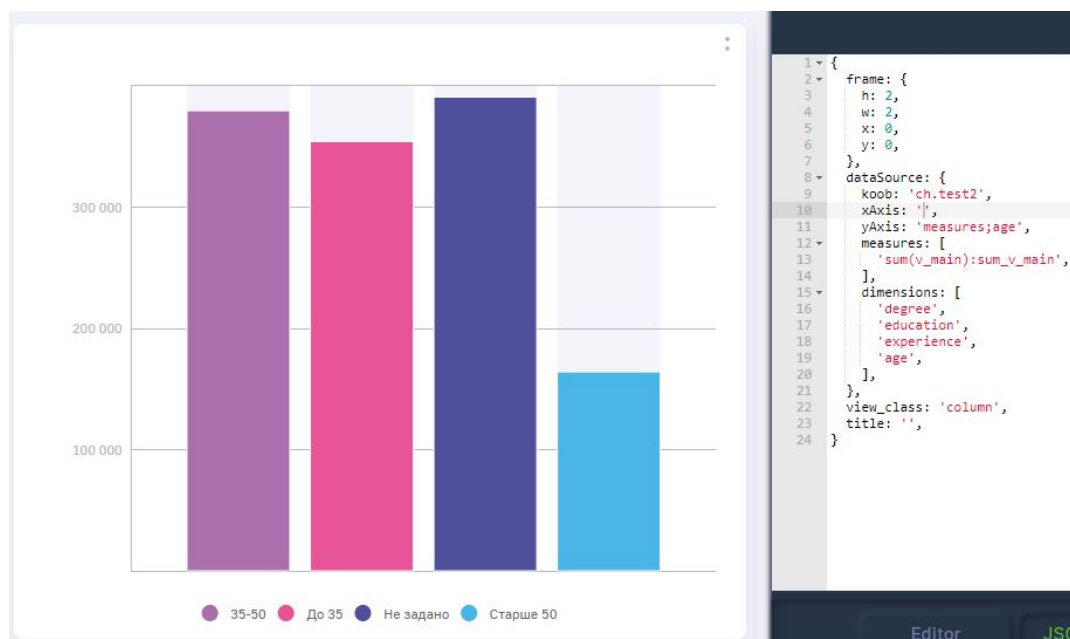


Рис. 2.10 Дэш, с отложенными размерностями на оси Y

В случае, если размерность, будет отложена на оси Y, то каждый показатель данной размерности будет отображен индивидуальным цветом. Ниже представлен диаграмма, у которой один показатель отложен по оси X, а другой по оси Y:



Рис. 2.11 Дэш, с отложенными размерностями на оси X и на оси Y

Из примера видно, что каждый цвет отображает один из рассматриваемых диапазонов возраста, которые сгруппированы по размерности `"experience"` - показателю, отложенному по оси X.

Существуют случаи, когда необходимо отфильтровать значения. На примере выше видно, что присутствует показатель "Не задано". Из-за большого количества значений по данному показателю, дэш отображен некорректно и не передает полноценно аналитические данные. Для фильтрации данных необходимо использовать поле `filters`.

Поле `filters` можно записать в файле в двух представлениях: в виде массива и в виде объекта.

Поле `filters` в формате массива используется для указания размерностей, фильтрация которых будет воспроизводится только с использованием управляющего дэша (о конфигурации управляющего дэша описано ниже) Пример записи поля `filters` в виде массива представлен ниже:

```
1 "filters":[
2     "degree",
3     "education",
4     "experience",
5     "age", ]
```

Внутри массива перечислены размерности, которые будут реагировать на фильтрацию управляющим дэшем. Аналогичное действие можно совершить задав поле `filters` как объект:

```
1 "filters":{
2     "degree": true,
3     "education": true,
4     "experience": true,
```

```

5     "age": true,
6 }

```

В случае, когда нам необходимо отфильтровать какой-то показатель сразу, независимо от управляющего дэша, нам необходимо использовать объект `filters`. Вернемся к вышеописанному примеру:

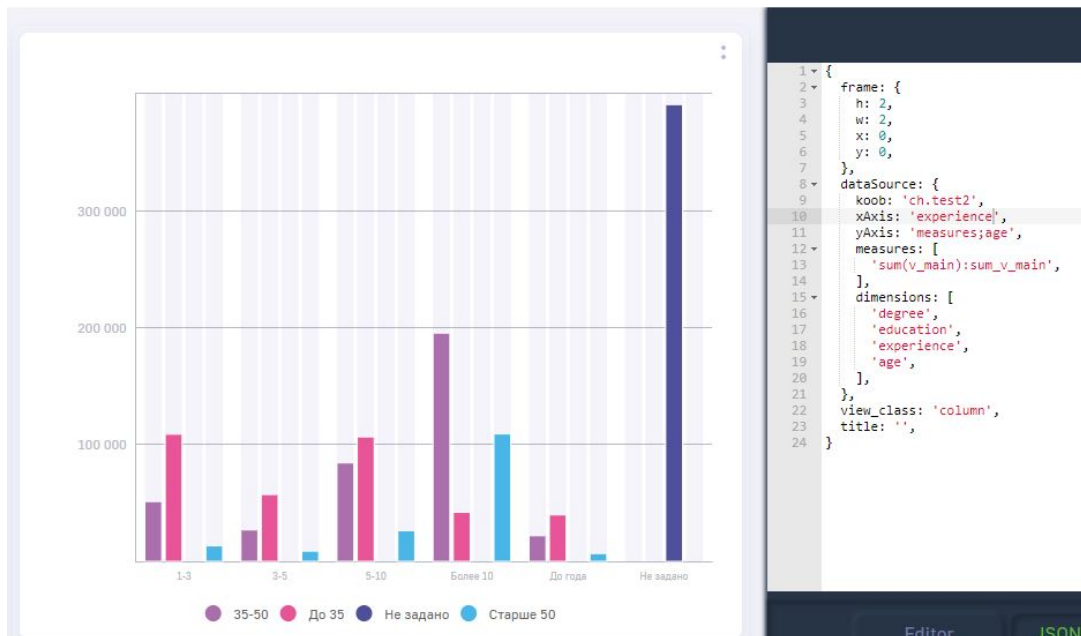


Рис. 2.12 Дэш с неотфильтрованными данными

Для того, чтобы отфильтровать значения “Не задано”, необходимо прописать в `filters` следующее:

```

1 "filters":{
2   "experience": [
3     "!=" ,
4     "Не задано"
5   ]
6 }

```

где `experience` - наименование размерности `!=` - условие фильтрации, говорящее о том, что необходимо оставлять только показатели неравные нижеперечисленным `"Не задано"` - фильтруемый показатель

Ниже приведен пример дэша после фильтрации:



Рис. 2.13 Фильтрация данных по условию “!=”

Как видим, показатель “Не задано” отфильтрован и не отображен на даше.

Аналогичным образом можно оставить только один показатель, прописав объект следующим образом:

```

1 "filters":{
2   "experience": [
3     "=",
4     "Не задано"
5   ]
6 }

```

Ниже представлен результат данной фильтрации:

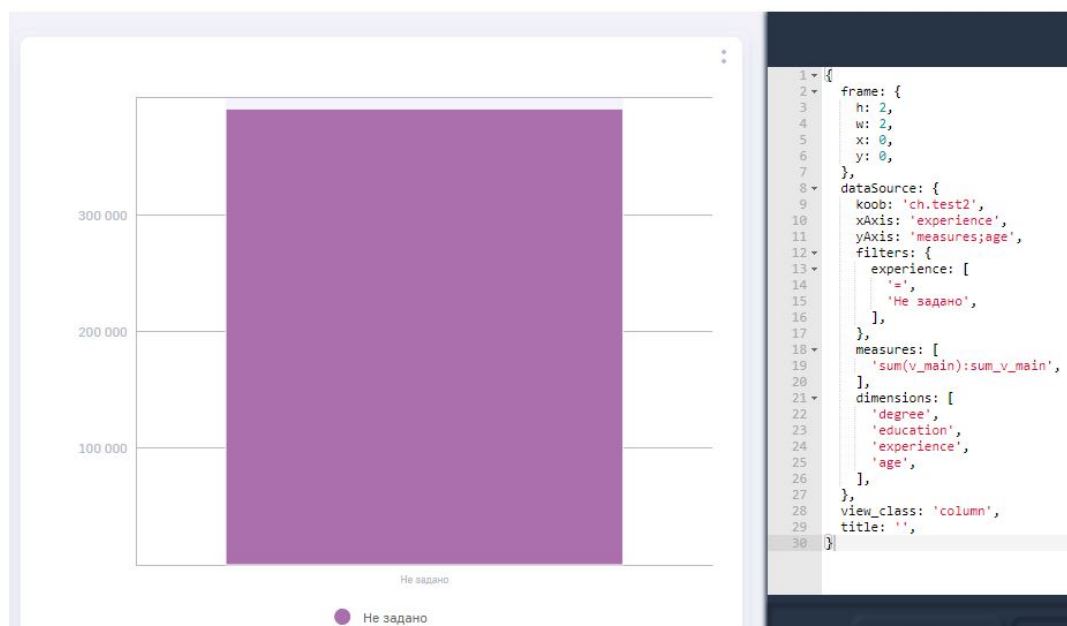


Рис. 2.14 Фильтрация данных по условию “=”

Как мы видим из рисунка выше, на дэше отображается только показатель, указанный в конфигурационном файле.

В условиях фильтрации можно использовать следующие операторы сравнения:

- =
- !=
- >
- >=
- <
- <=
- between
- like
- ilike



В случае, когда в кубе вместо пустых значений указан null, BI будет воспринимать эти данные как “подытог”. Для скрытия “подытога” необходимо отфильтровать данные по null (“!=”, “null”)

Если необходимо исключить один из показателей размерности, но чтобы размерность реагировала на действия в управляющем дэше



```
filters: { "": [ '!=', [ 'column', 'sex', ], 'Мужчины', ],
sex: true, },
```

2.4.2 Стилизация дэшей

Иногда возникает необходимость стилизации дэшей: указать конкретный цвет для показателя или размерности, поменять заголовок и т.д. Для таких случаев используется объект `style`. Рассмотрим случай, когда у нас два факта и одна размерность:

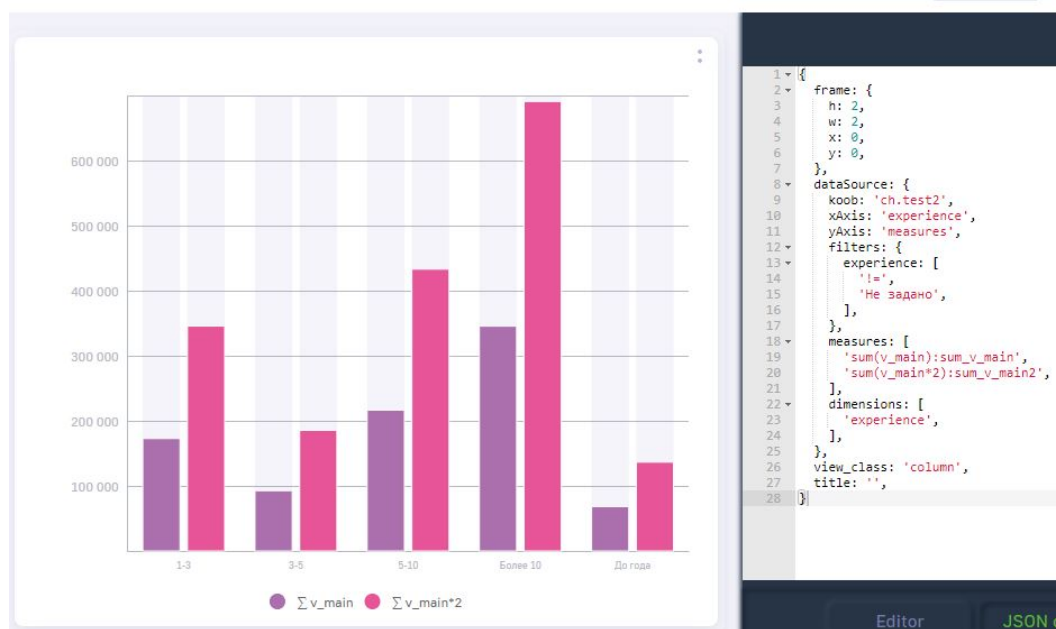


Рис. 2.15 Дэш “Столбики” для двух фактов и одной размерности

Объект `style` для стилизации дэшей составляется следующим образом:

```

1  "style":{
2    "measures":{
3      "sum_v_main":{
4        "title": "Факт1",
5        "color": "#3b3fb8",
6        "vizelType": "line",
7        "strokeDash": "ShortDash",
8      },
9      "sum_v_main2":{
10       ...
11     }
12   },
13 }
```

где `sum_v_main/sum_v_main2` - названия стилизуемых фактов `title` - поле указания заголовка для факта `color` - указание цвета факта `vizelType` - указание варианта отображения факта (возможные варианты: **line** - линии, **scatter** - точки, **bar** - столбцы) `strokeDash` - вариант отображения факта, в случае его отображения в виде “линии”. Ниже представлен список доступных значений:

1. **Solid** - сплошная линия
2. **ShortDot** - линия отображена точками, расположенными близко относительно друг друга

3. **ShortDash** - линия отображена черточками небольшой длины
4. **ShortDashDot** - линия отображена черточками небольшой длины и точками
5. **ShortDashDotDot** - линия отображена черточками небольшой длины и двумя точками
6. **Dot** - отображение точками
7. **Dash** - линия отображена черточками
8. **LongDash** - линия отображена длинными черточками(тире)
9. **DashDot** - линия отображена черточками и точками
10. **LongDashDot** - линия отображена длинными черточками(тире) и точкой
11. **LongDashDotDot** - линия отображена длинными черточками(тире) и двумя точками;

Аналогичную стилизацию можно проводить и с показателями размерностей, в случае если они отложены на оси Y. Ниже представлен пример стилизации:

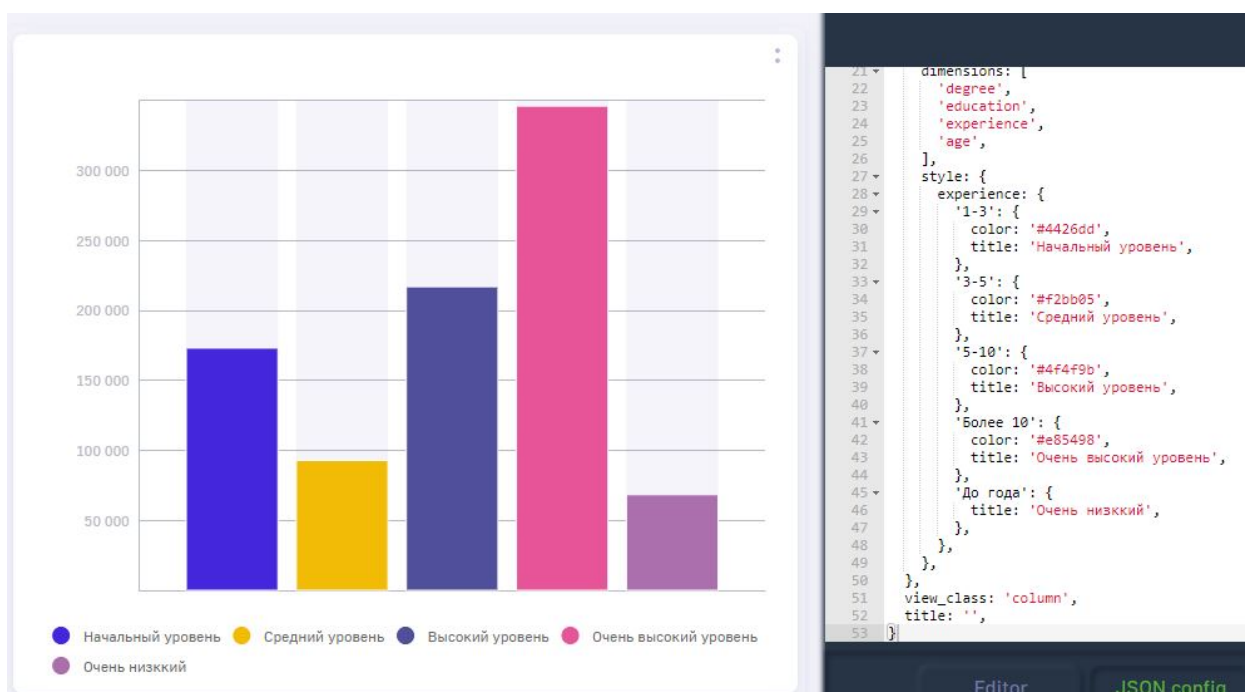


Рис. 2.16 Стилизованные показатели в дэше “Столбики”

Пример конфигурации стилей для размерности приведен ниже:

```

1  "style":{
2    "experience":{
3      "1-3":{
4        "color": "#4426dd",
5        "title": "Начальный уровень"
6      },
7      "3-5":{
8        ...
9      },
10     ...
11   }
12 }

```

Как видно в примерах выше, стилизация размерностей и фактов происходит практически аналогично. Разница в том, что размерности пишутся на верхнем уровне поля `style`, на уровне ниже прописываются стили значений выбранной размерности. Факты же на верхнем объединяются в ключи `measures`, далее происходит стилизация каждого отдельного факта.



Объект `style` записывается в объект `dataSource` для всех типов дэша, за исключением типа “Значение”, описание которого продемонстрировано ниже.

2.5 Секция `display`

Секция `display` используется для редактирования отображения данных в дэше (сортировка, ограничения количества выводимых данных и т.д.)

Рассмотрим вышеописанный пример сконфигурированного дэша “Столбики”:

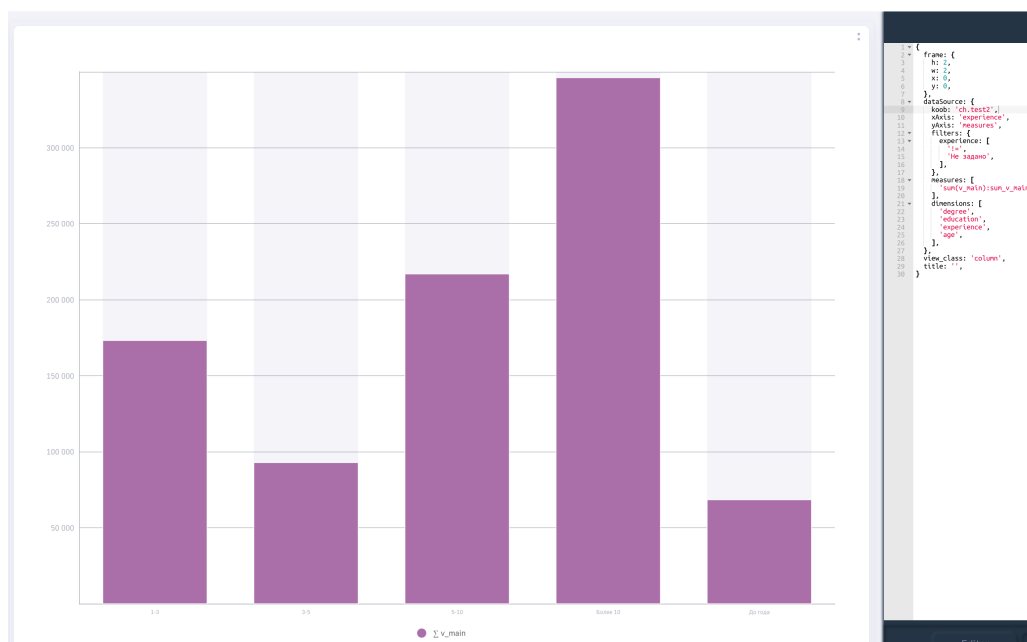


Рис. 2.17 Дэш “Столбики”

В случае необходимости сортировки столбцов по возрастанию/убыванию, в секцию `display` необходимо ввести следующие поля:

```

1 "display": {
2   "sortBy": "sum_v_main"
3 }

```

где `'sum_v_main'` - заголовок факта, относительно которого необходимо сортировать значения.

После ввода вышеописанного блока в конфигурационный файл, данные будут отсортированы по возрастанию:

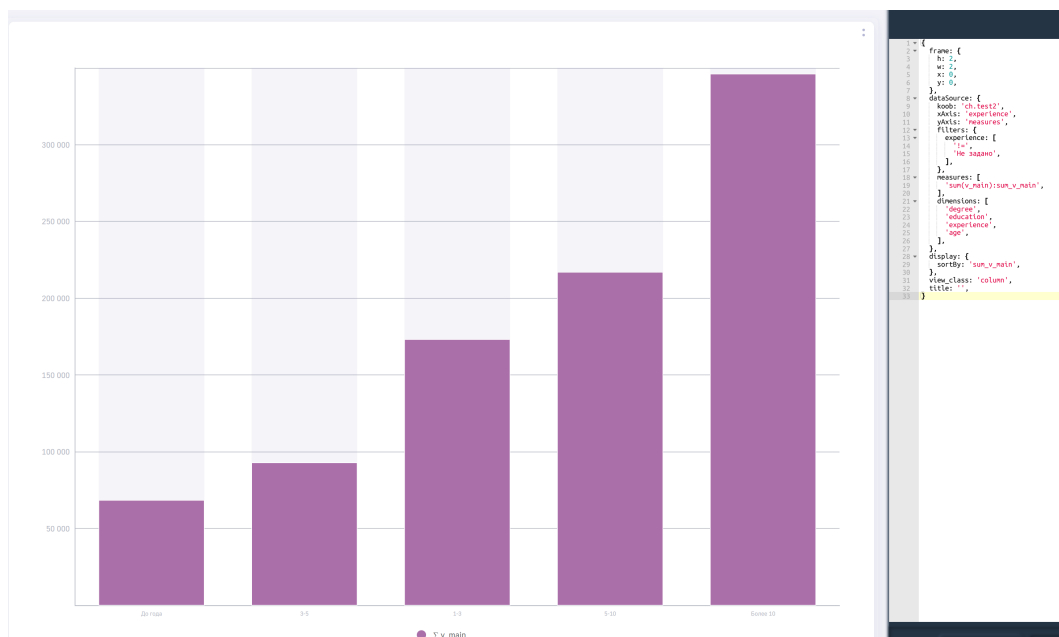


Рис. 2.18 Дэш с отсортированными столбцами по возрастанию

Для сортировки по убыванию для текущего факта необходимо дописать поле `sort` со значением `DESC`, следовательно, блок `display` будет описан следующим образом:

```

1 "display":{
2   "sortBy": "sum_v_main",
3   "sort": "DESC"
4 }

```

Ниже представлен пример отображения с вышеописанными полями.

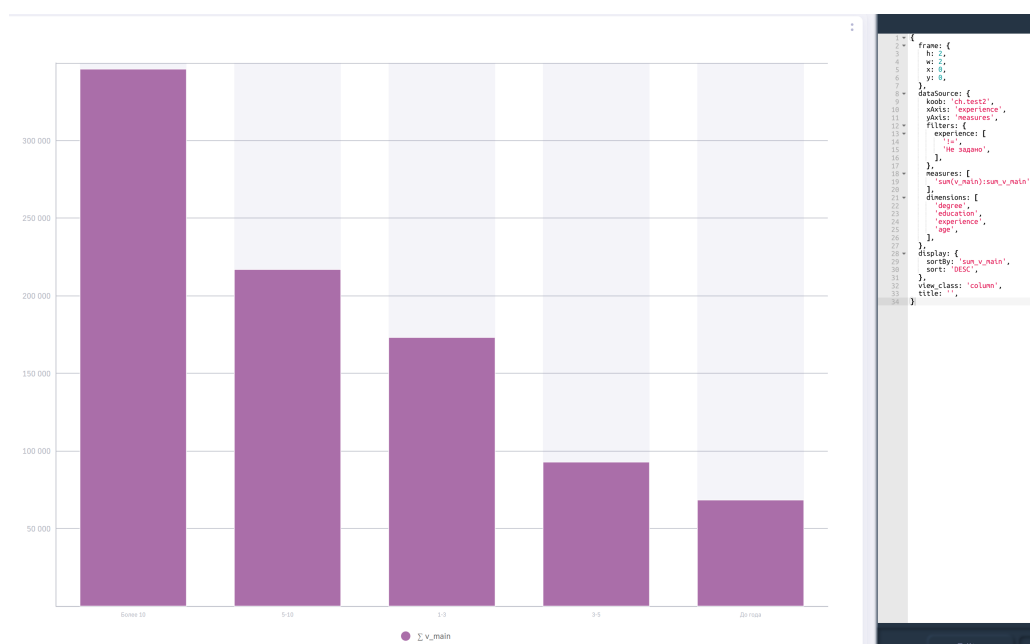


Рис. 2.19 Дэш с отсортированными столбцами по убыванию

Для случаев, когда размерность, имеющая тип “строка”, имеет показатели с большим коли-

чеством символов, в случае ограниченного количества места на дэшборде, названия показателей будут налезать друг на друга:

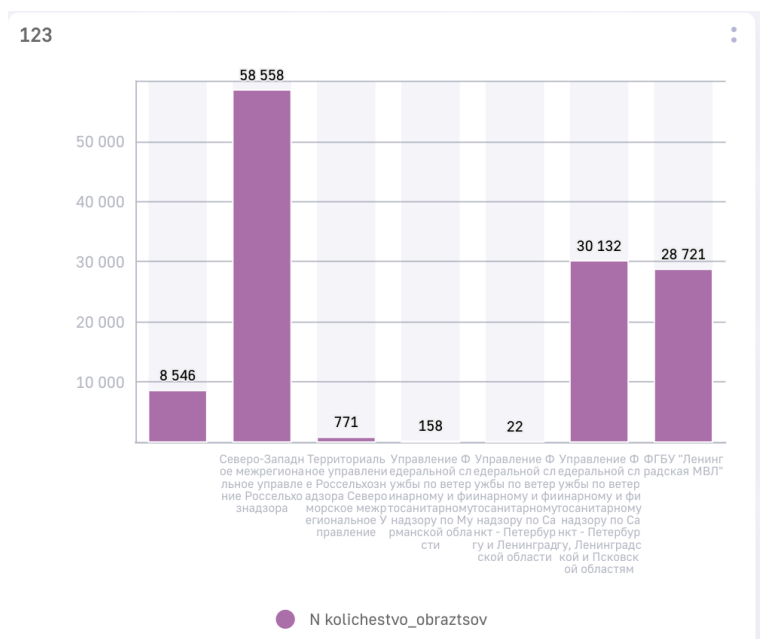


Рис. 2.20 Наложение подписей по оси X

Для таких случаев существуют поля `xAxisLabelLimit` - для ограничения вывода подписей по осям X и Y соответственно. Ниже представлены примеры использования данных полей:

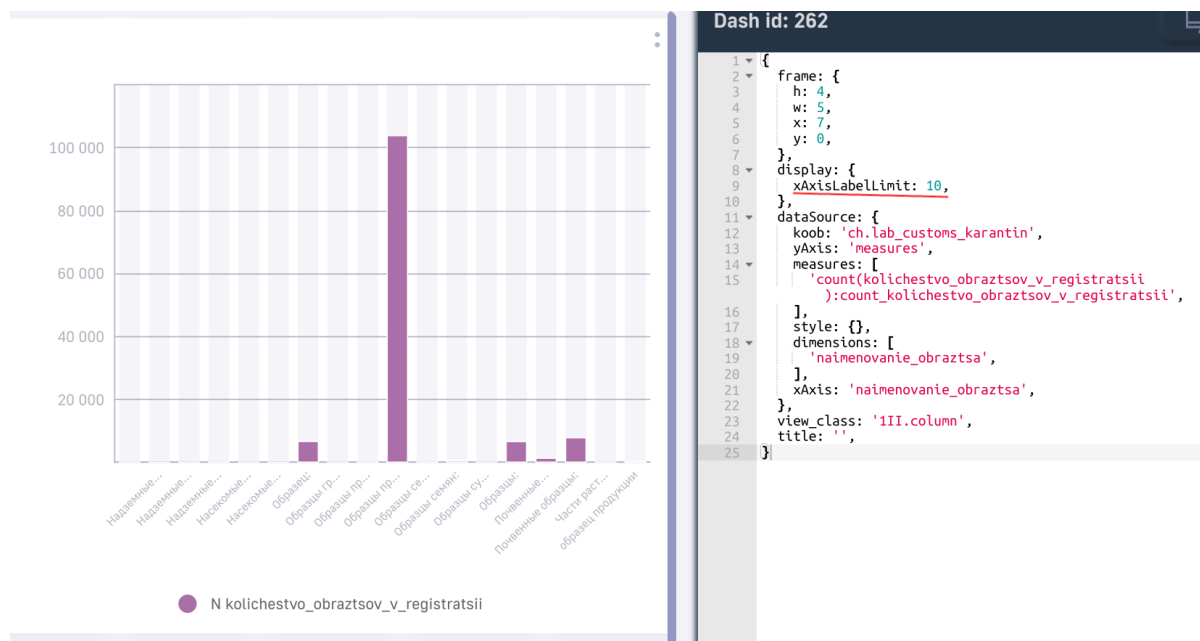


Рис. 2.21 Использование `xAxisLabelLimit` для дэша “Вертикальные столбики”

Данные поля ограничивают отображение текста на оси тем количеством символов, которое указано в конфигурационном файле. Для отображения полного наименования в всплывающей подсказке (тултипе) необходимо указать в массиве `options` опцию `TooltipXAxisTitle`.

Помимо этого, существует возможность менять угол отображения наименования показателей на оси, используя поле `rotateXLabel` со значением угла поворота. Ниже представлен пример дэша с использованием данного функционала:



Рис. 2.22 Использование `rotateXLabel` для дэша “Линии”

Все вышеописанные действия (включая объект `style` внутри блока `dataSource`) применимы для двумерных типов дэшей (столбики, линии, точки, штабели, и т.д.)

Далее будут представлены опции для нольмерных типов дэшей: значение, термометр, спидометры и т.д.

2.5.1 Стилизация типа дэша “Значение”

Для стилизации типа дэша значение (`label`, `text`, `value`) необходимо использовать секцию `display` и использовать поля, созданные специально для рассматриваемого дэша. Ниже приведен простой пример конфигураций дэша “Значение”:

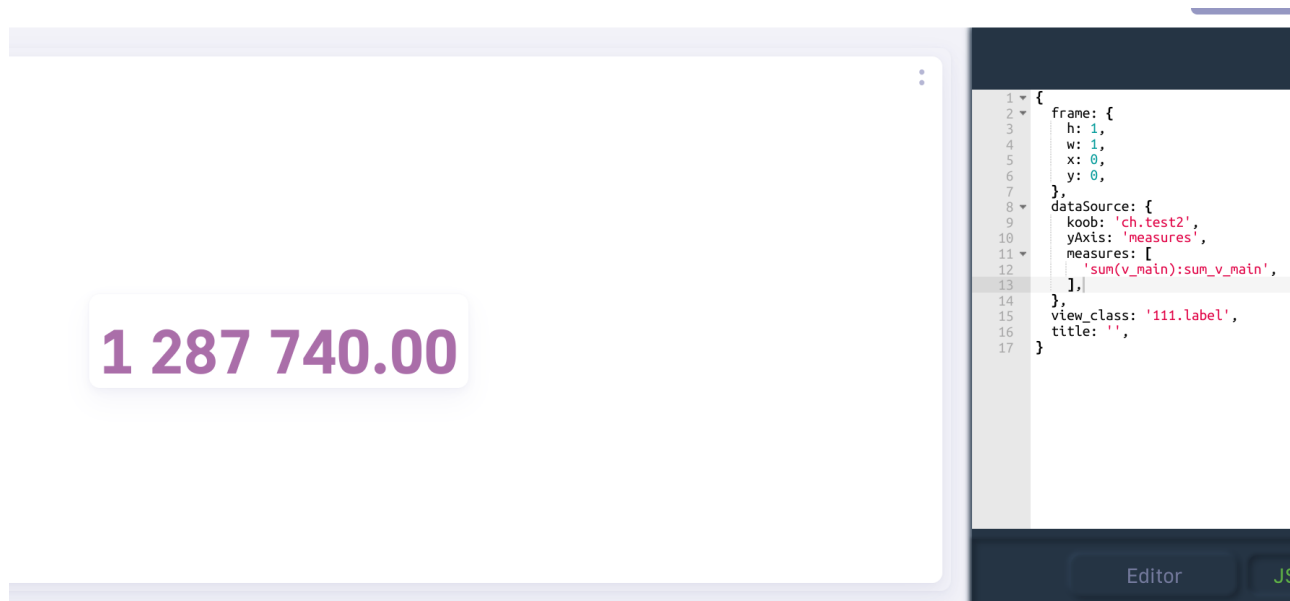


Рис. 2.23 Дэш “Значение”

На скриншоте в конфигурационном файле представлены все поля, описанные выше.

Данное значение существует возможность подписать сверху, для этого используется поле `title`. Для этого добавим в конфигурационный файл объект `display` с полем `title`:

```

1 "display": {
2   "title": "Заголовок значения",
3 }

```

Ниже представлено отображения дэша после сохранения изменений в конфигурационном дэше:

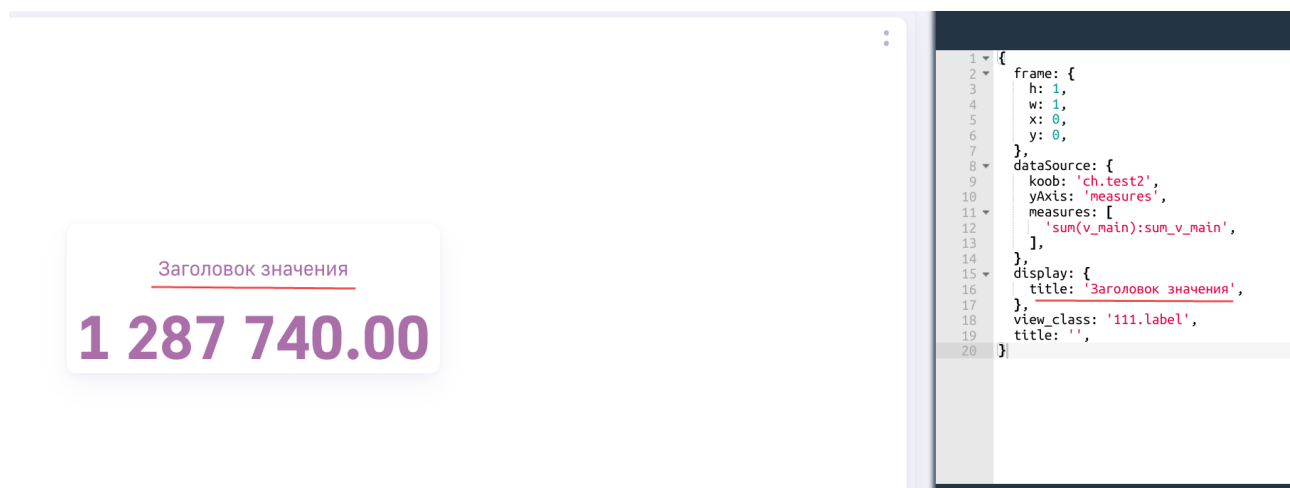


Рис. 2.24 Указание заголовка значения

Для изменения цвета значения и заголовка используется поле `color` с указанием шестнадцатеричного кода цвета. Ниже представлен пример изменения цвета после указания его в конфигурационном файле:

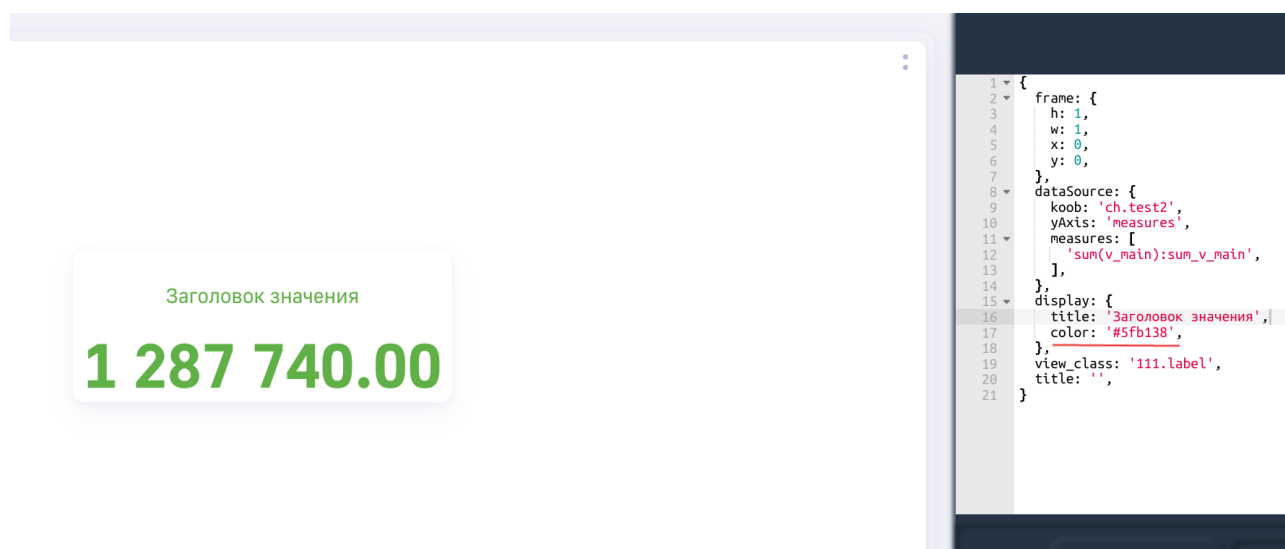


Рис. 2.25 Изменение цвета значения

Аналогичным образом, для дэша “Значение” существует возможность поменять цвет фона, используя поле `bgColor`:

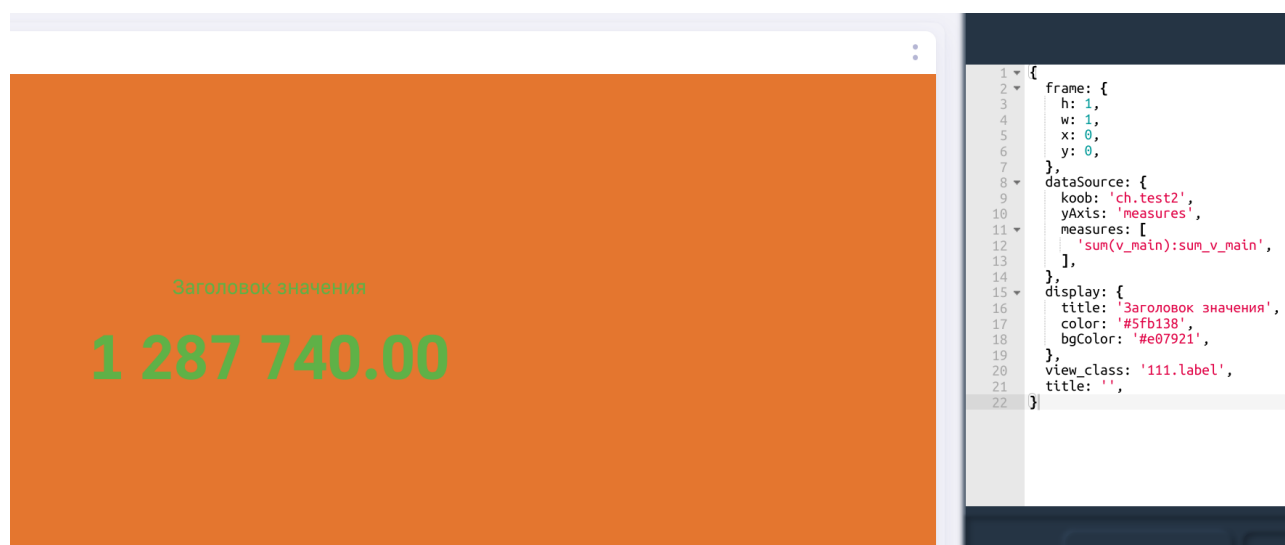


Рис. 2.26 Изменение цвета фона для дэша “Значение”

Также, для данного дэша присутствует возможность указания конкретного шрифта, используя поле `fontFamily`. Полный набор вариантов шрифтов представлен в предудщем разделе.

Ниже представлен пример “Значения” с установленным шрифтов "Times New Roman":

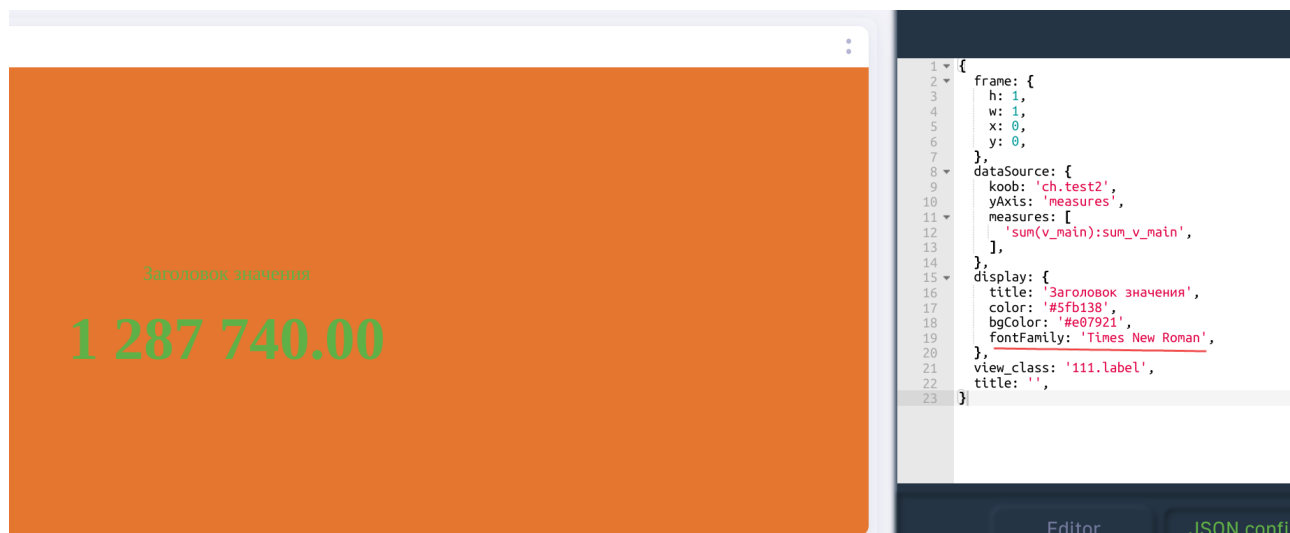
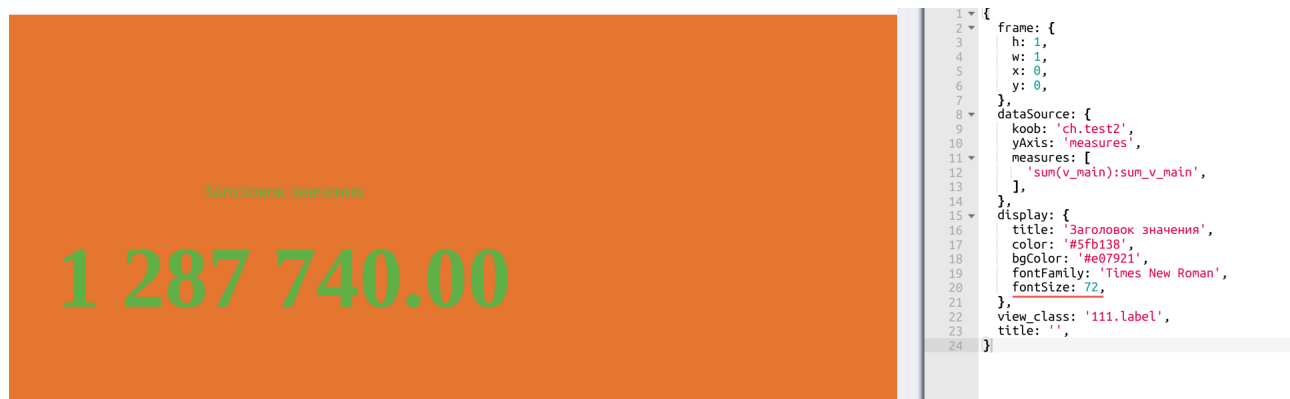
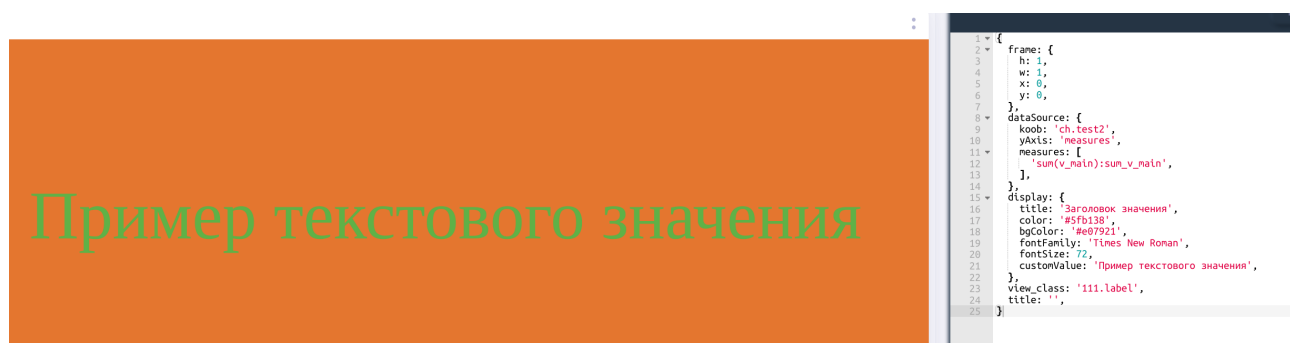


Рис. 2.27 Установка шрифта

Помимо этого, существует возможность указать размер минимальный/максимальный или конкретный размер шрифта для значения. Для этого используются поля `maxFontSize`, `minFontSize`, `fontSize` соответственно. Размер шрифта задается в относительных единицах. Пример использования представлен ниже:

Рис. 2.28 Использование поля `fontSize`

В случае, когда нам необходимо указать статичный текст в дэше, используется поле `customValue`. Ниже представлен пример использования нижеописанного поля:

Рис. 2.29 Поле `customValue` для дэша “Значение”

Ниже представлен итоговый конфигурационный файл, полученный для дэша “Значение” в ходе примеров:

```
1 {
2   "frame": {
3     "h": 1,
4     "w": 1,
5     "x": 0,
6     "y": 0,
7   },
8   "dataSource": {
9     "koob": "ch.test2",
10    "yAxis": "measures",
11    "measures": [
12      "sum(v_main):sum_v_main",
13    ],
14  },
15  "display": {
16    "title": "Заголовок значения",
17    "color": "#5fb138",
18    "bgColor": "#e07921",
19    "fontFamily": "Times New Roman",
20    "fontSize": 72,
21    "customValue": "Пример текстового значения",
22  },
23  "view_class": "111.label",
24  "title": "",
25 }
```

Так же в конфигурацию дэша Значение можно добавить размерность (dimension). В этом случае по умолчанию в дэше “Значение” сравнивается 2 значения одного факта по двум размерностям, которые пришли последними из базы, либо указанным в блоке filters.

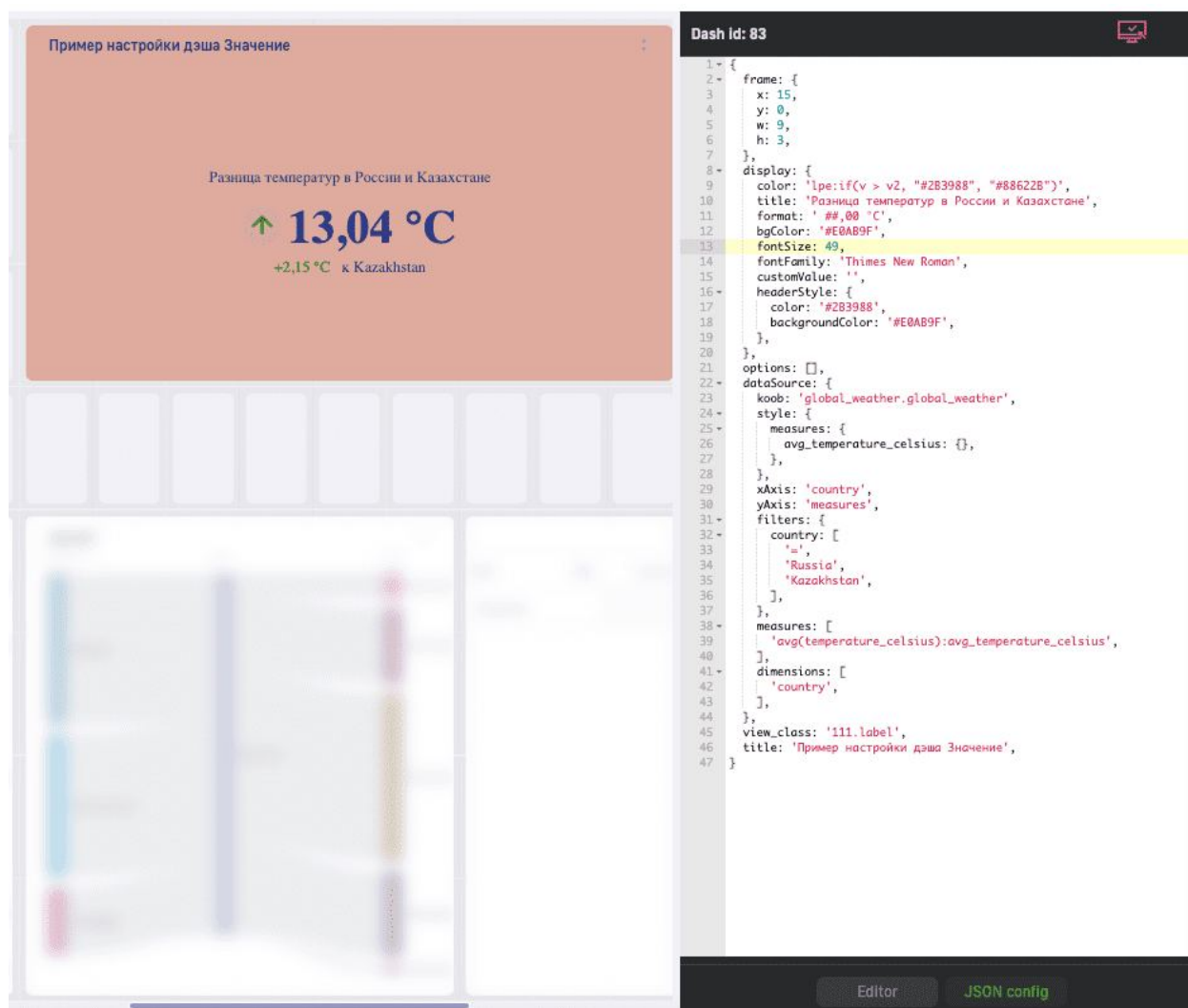


Рис. 2.30 Стилизация второго значения

С помощью дэша “Значение” можно сравнивать 2 факта с разбивкой и без по dimensions. Для этого в конструкторе требуется перенести Факты в поле “Категории” или в конфиге внутри dataSource добавить настройку xAxis: ‘measures’.

Настройка дэша Лэйбл с двумя значениями. Для этого можно воспользоваться опциями: `cmpTitle` - добавление подписи ко второму значению `forcedAlign` - принудительно выставить по центру блок со вторым значением и подписью `subtitleCfg` - объект для настройки стилей подписи ко второму значению `additionalValues` - массив для настройки дополнительных значений (в данной версии принимает только 1 объект). `v` - переменная, в которую передается первое значение `secondValue` - переменная, в которую передается второе значение

Ниже конфиг дэша Лэйбл с двумя значениями, а так же результат настройки.

```

1 {
2   frame: {
3     x: 6,
4     y: 0,
5     w: 7,

```



```

6     h: 4,
7 },
8 display: {
9     color: 'lpe:if(v < 0, "red", "green")',
10    title: 'lpe:if(v > secondValue, "a", "b")',
11    format: '##,000°C ',
12    bgColor: '#E0AB9F',
13    cmpTitle: 'lpe:if(v > secondValue, "первое больше второго", "второе больше п(←
первого")',
14    fontSize: 50,
15    _cmpTitle: 'подпись ко второму значению',
16    fontFamily: 'Times New Roman',
17    forcedAlign: true,
18    headerStyle: {
19        color: '#2B3988',
20        fontSize: 30,
21        fontFamily: 'Times New Roman',
22        backgroundColor: '#E0AB9F',
23    },
24    subtitleCfg: {
25        color: 'lpe:if(v > secondValue, "red", "green")',
26        fontSize: 30,
27        fontFamily: 'Courier New',
28    },
29    _customValue: 'Альтернативная подпись, чтобы скрыть данные',
30 },
31 options: [],
32 dataSource: {
33     koob: 'global_weather.global_weather',
34     style: {},
35     xAxis: 'country',
36     yAxis: 'measures',
37     filters: {
38         country: [
39             '=',
40             'Russia',
41             'Kazakhstan',
42         ],
43     },
44     measures: [
45         'avg(temperature_celsius):avg_temperature_celsius',
46     ],
47     dimensions: [
48         'country',
49     ],
50     additionalValues: [
51         {
52             koob: 'global_weather.global_weather',
53             style: {
54                 measures: {
55                     avg_temperature_celsius: {
56                         color: 'lpe:if(v > secondValue, "red", "blue")',
57                         format: '##,0°C ',

```

```
58         fontSize: '50px',
59         fontFamily: 'Courier New',
60     },
61 },
62 },
63 xAxis: 'country',
64 yAxis: 'measures',
65 measures: [
66     'avg(temperature_celsius):avg_temperature_celsius',
67 ],
68 dimensions: [
69     'country',
70 ],
71 },
72 ],
73 },
74 view_class: '111.label',
75 title: 'Дэшлет лэйбл с двумя значениями',
76 }
```

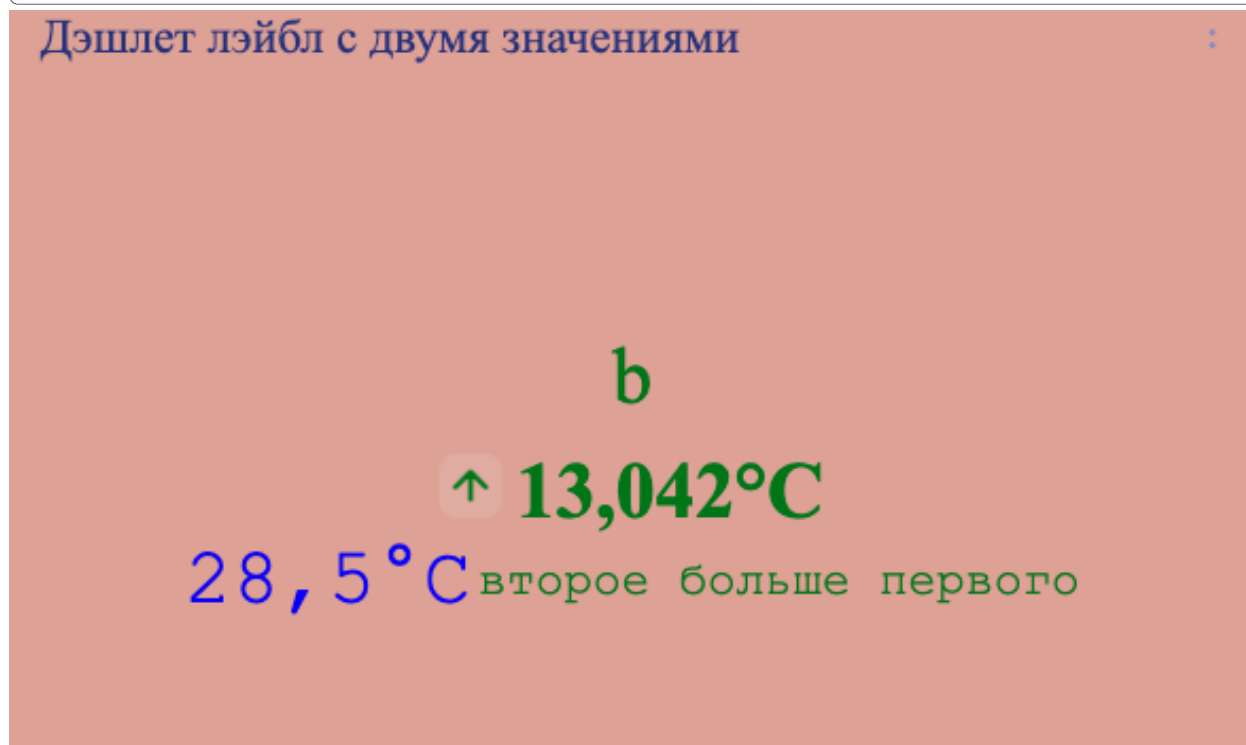


Рис. 2.31 Лэйбл с двумя значениями

2.5.2 Объект stoplights / массив range

Опции, описанные в данном подразделе, используются первоочередно в таких типах дэша как: спидометр, полуспидометр, термометр. Для примера возьмем дэш “Спидометр” и отобразим его используя вышеописанные поля:

```

1 {
2   "frame": {
3     "h": 1,
4     "w": 1,
5     "x": 0,
6     "y": 0,
7   },
8   "dataSource": {
9     "koob": "ch.test2",
10    "yAxis": "measures",
11    "measures": [
12      "sum(v_main):sum_v_main",
13    ],
14    "dimensions": [
15      "degree",
16    ],
17  },
18  "view_class": "gauge",
19  "title": "",
20 }

```

Используя вышепредставленную конфигурацию, мы получим спидометр следующего вида:

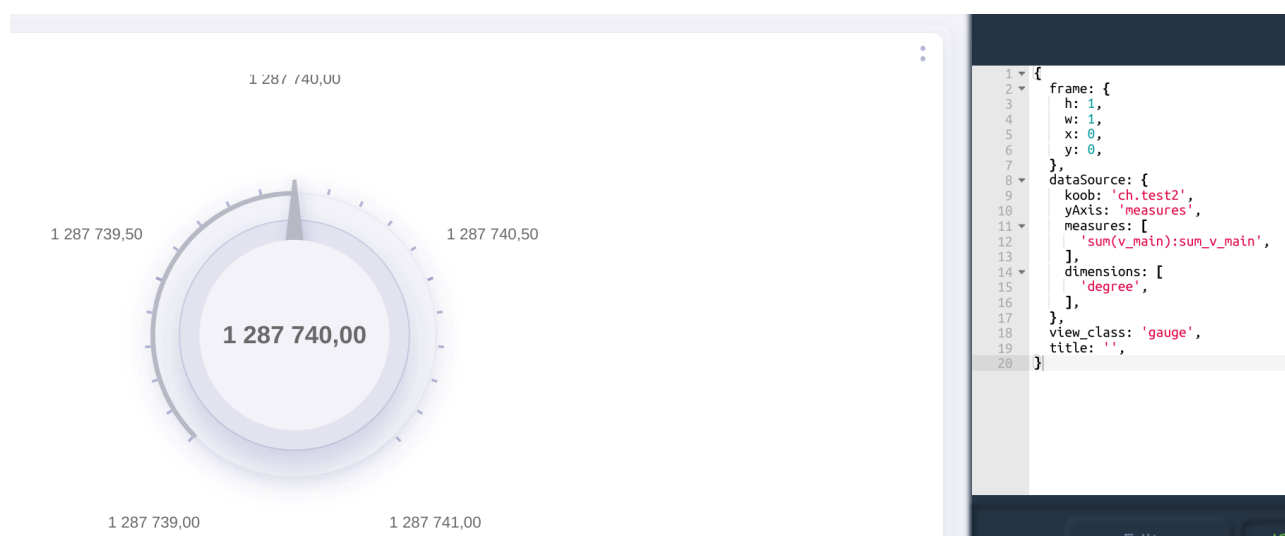
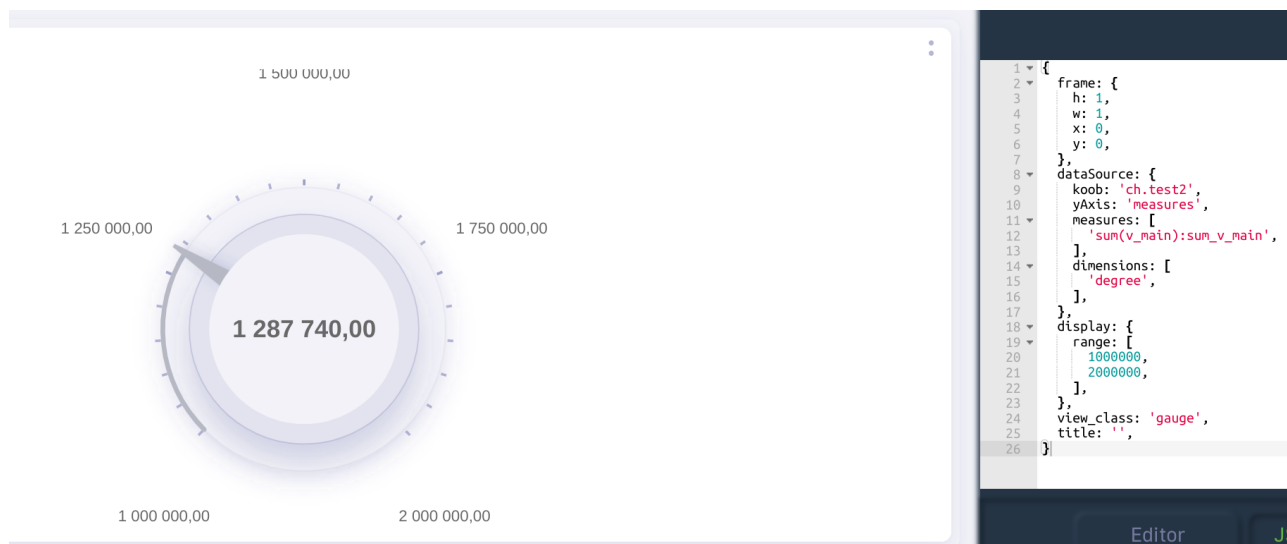


Рис. 2.32 Дэш “Спидометр”

Как можно заметить, в состоянии по-умолчанию спидометр отображает значения в диапазоне от значение - 1 до значение + 1. С помощью массива `range` в блоке `display` можно указать отображаемый диапазон значений для спидометров/термометра. Ниже представлено изображение спидометра с указанным диапазоном от миллиона до двух миллионов:

Рис. 2.33 Спидометра с указанным диапазоном в `range`

В большинстве случаев для спидометра используется для отображения того, что значение показателя находится в нужном диапазоне или нет (например, показать выполнение плана и т.д.). Для таких случаев необходимо использовать объект `stoplights` внутри которого массив объектов `lights`. Ниже приведен пример конструкции объекта `stoplights`:

```

1 "stoplight": {
2   "lights": [
3     {
4       "name": "Плохо",
5       "color": "#f05045",
6       "limit": [
7         -Infinity,
8         1500000,
9       ],
10    },
11    {
12      "name": "Средне",
13      "color": "#f2bb05",
14      "limit": [
15        1500000,
16        1750000,
17      ],
18    },
19    {
20      "name": "Хорошо",
21      "color": "#5fb138",
22      "limit": [
23        1750000,
24        Infinity,
25      ],
26    },
27    ...
28  ],
29 },

```

массив `lights` состоит из объектов, включающие в себя следующие поля: `name` - название зоны `color` - цвет зоны, указывается #HEX цвета `limit` - массив, в котором указывается два значения - диапазон, в котором будет отрисован цвет (-Infinity/Infinity используется для указания, что диапазон начинается с -бесконечности или до бесконечности)



Диапазоны, указанные в `stoplights` будут отрисованы на дэше в диапазоне, указанном в `range`

Ниже представлен пример использования `stoplights` и `range` для дэша спидометр:

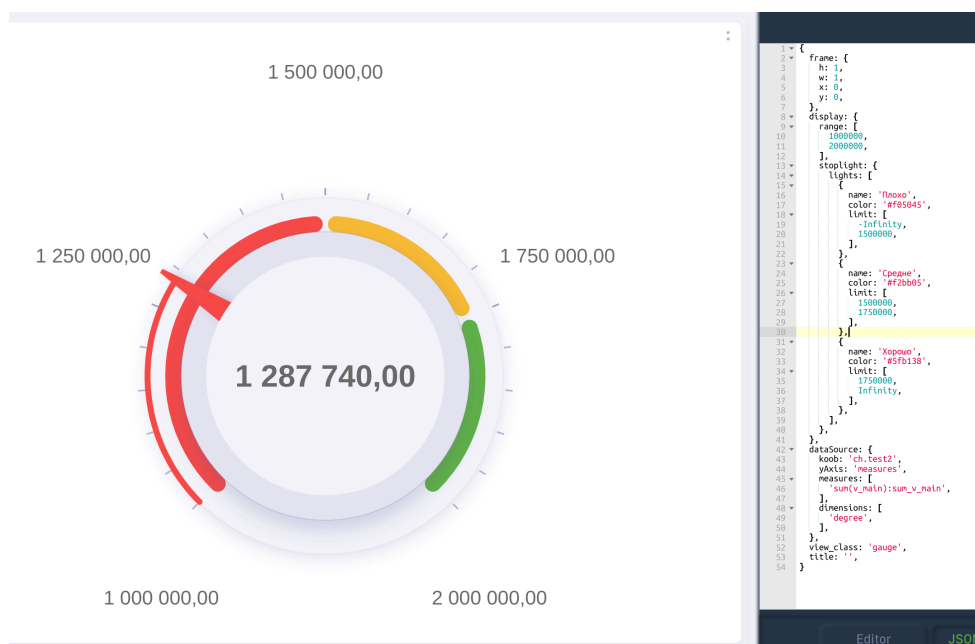


Рис. 2.34 `stoplights` и `range` для дэша спидометр

Конфигурационный файл данного дэша выглядит следующим образом:

```

1 {
2   "frame": {
3     "h": 1,
4     "w": 1,
5     "x": 0,
6     "y": 0,
7   },
8   "display": {
9     "range": [
10      1000000,
11      2000000,
12    ],
13    "stoplight": {
14      "lights": [
15        {
16          "name": "Плохо",
17          "color": "#f05045",
18          "limit": [
19            -Infinity,

```

```
20         1500000,
21     ],
22 },
23 {
24     "name": "Средне",
25     "color": "#f2bb05",
26     "limit": [
27         1500000,
28         1750000,
29     ],
30 },
31 {
32     "name": "Хорошо",
33     "color": "#5fb138",
34     "limit": [
35         1750000,
36         Infinity,
37     ],
38 },
39 ],
40 },
41 },
42 "dataSource": {
43     "koob": "ch.test2",
44     "yAxis": "measures",
45     "measures": [
46         "sum(v_main):sum_v_main",
47     ],
48     "dimensions": [
49         "degree",
50     ],
51 },
52 "view_class": "gauge",
53 "title": "",
54 }
```

Объект `stoplights` также может использоваться и для двумерных дэшей. Ниже представлен пример использования `stoplights` для дэша “Столбики горизонтальные”:

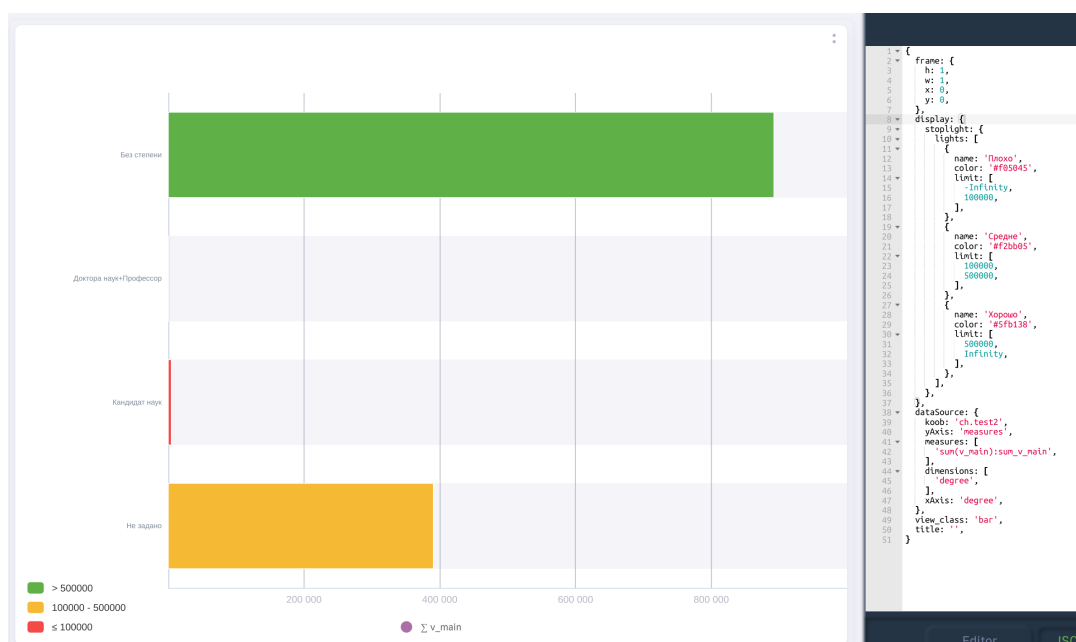


Рис. 2.35 stoplights для дэша “Столбики горизонтальные”

Конфигурационный файл примера:

```

1  {
2    "frame": {
3      "h": 1,
4      "w": 1,
5      "x": 0,
6      "y": 0,
7    },
8    "display": {
9      "stoplight": {
10       "lights": [
11         {
12           "name": "Плохо",
13           "color": "#f05045",
14           "limit": [
15             -Infinity,
16             100000,
17           ],
18         },
19         {
20           "name": "Средне",
21           "color": "#f2bb05",
22           "limit": [
23             100000,
24             500000,
25           ],
26         },
27         {
28           "name": "Хорошо",
29           "color": "#5fb138",
30           "limit": [

```

```

31         500000,
32         Infinity,
33     ],
34 },
35 ],
36 },
37 },
38 "dataSource": {
39     "koob": "ch.test2",
40     "yAxis": "measures",
41     "measures": [
42         "sum(v_main):sum_v_main",
43     ],
44     "dimensions": [
45         "degree",
46     ],
47     "xAxis": "degree",
48 },
49 "view_class": "bar",
50 "title": "",
51 }

```

Помимо указания конкретных значений в `stoplights` существует возможность градиентного их представления с использованием констант. Для этого необходимо использовать поле `stoplight`. Ниже представлен список констант для указания в `stoplight`:

1. `XL_GREEN_YELLOW_RED` - переход зеленый - желтый - красный
2. `XL_RED_YELLOW_GREEN` - переход красный - желтый - зеленый
3. `XL_GREEN_WHITE_RED` - переход зеленый - белый - красный
4. `XL_RED_WHITE_GREEN` - переход красный - белый - зеленый
5. `XL_BLUE_WHITE_RED` - переход синий - белый - красный
6. `XL_RED_WHITE_BLUE` - переход красный - белый - синий
7. `XL_WHITE_RED` - переход белый - красный
8. `XL_RED_WHITE` - переход красный - белый
9. `XL_GREEN_WHITE` - переход зеленый - белый
10. `XL_WHITE_GREEN` - переход белый - зеленый
11. `XL_GREEN_YELLOW` - переход зеленый - желтый
12. `XL_YELLOW_GREEN` - переход желтый - зеленый

Ниже представлен пример отображение горизонтальных столбцов с использованием `stoplight`:

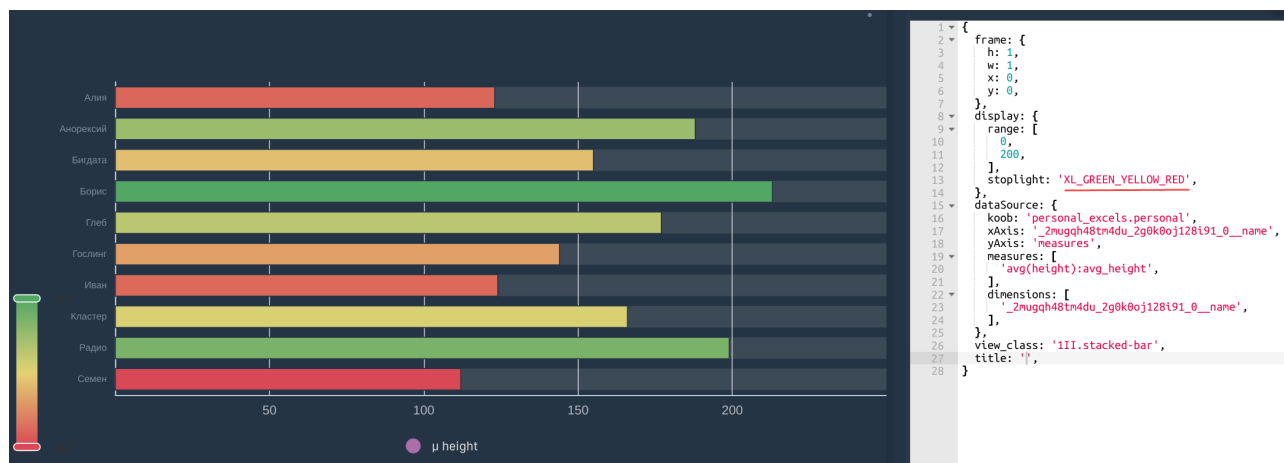


Рис. 2.36 Использование опции stoplight

2.6 Секция options

При конфигурации дэшей, существует набор опций, указывая которые можно добавлять-/удалять различные артефакты дэша. Данные опции прописываются в массиве `options` в слеующем формате:

```

1 "options": [
2   "опция_1",
3   "опция_2",
4   ...]

```

Ниже представлен пример использования опций для дэша “Столбики”:

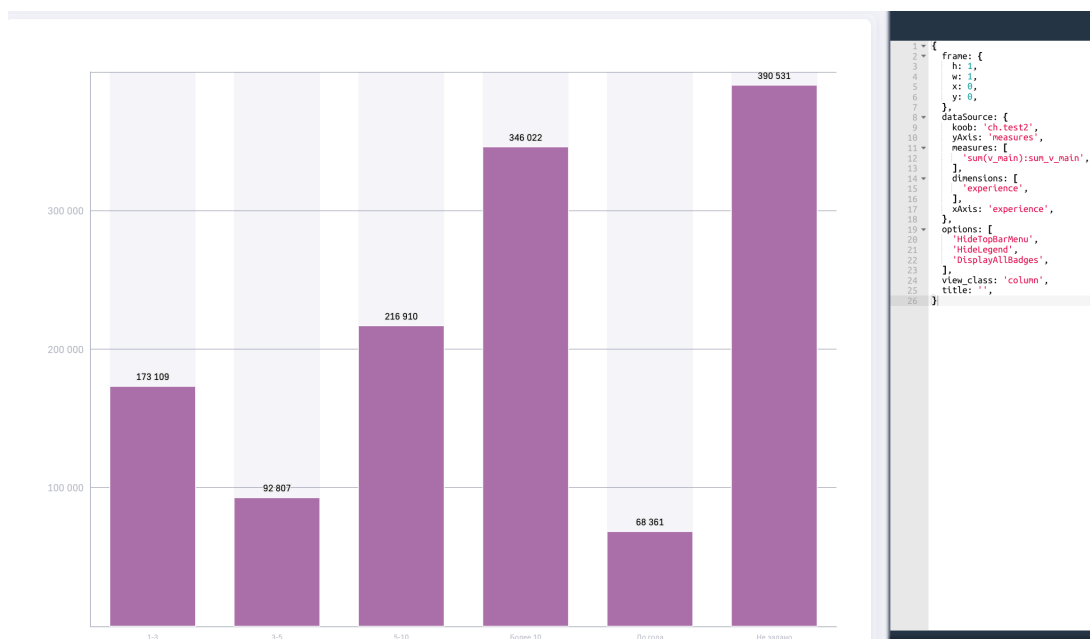


Рис. 2.37 Пример использования опций для дэша “Столбики”

Конфигурация дэша:

```

1 {
2   "frame": {
3     "h": 1,
4     "w": 1,
5     "x": 0,
6     "y": 0,
7   },
8   "dataSource": {
9     "koob": "ch.test2",
10    "yAxis": "measures",
11    "measures": [
12      "sum(v_main):sum_v_main",
13    ],
14    "dimensions": [
15      "experience",
16    ],
17    "xAxis": "experience",
18  },
19  "options": [
20    "HideTopBarMenu",
21    "HideLegend",
22    "DisplayAllBadges",
23  ],
24  "view_class": "column",
25  "title": "",
26 }

```

где `HideTopBarMenu` - скрыть меню дэша `HideLegend` - скрыть легенду дэша `DisplayAllBadges` - отобразить все значения на графике

2.7 Пример конфигурации визеля map

Для отображения данных на карте, используется тип дэша “Карта”, имеющий индивидуальные поля в конфигурационном файле. Ниже представлен пример конфигурирования “Карты”:

```

1 {
2   "frame": {
3     "h": 1,
4     "w": 1,
5     "x": 0,
6     "y": 0,
7   },
8   "children": [
9     {
10      "title": "Точки",
11      "display": {
12        "stoplight": "XL_BLUE_WHITE_RED",
13        "defaultActive": false
14      },

```

```

15     dataSource: {
16         "koob": "sourceWithCountryAndFo.goodKoobMap",
17         "style": {
18             "measures": {
19                 "v": {
20                     "title": "Показатель4",
21                 },
22                 "a1": {
23                     "title": "Показатель3",
24                 },
25                 "val2": {
26                     "title": "Показатель2",
27                 },
28             },
29         },
30         "xAxis": "name;lng;lat",
31         "yAxis": "measures",
32         "filters": {
33             lat: [
34                 ">",
35                 60,
36             ],
37         },
38         "measures": [
39             "sum(val):v",
40             "sum(val2)",
41             "concat("Бонусная-", sum(val2)):a1",
42         ],
43         "dimensions": [
44             "name",
45             "lng",
46             "lat",
47             "concat("Координаты", lng, "x", lat):aa",
48         ],
49     },
50     "view_class": "mapdots",
51 },
52 {
53     "title": "Области",
54     "display": {
55         "stoplight": "XL_RED_YELLOW_GREEN",
56     },
57     "dataSource": {
58         "koob": "testMAP.cub",
59         "style": {
60             "measures": {
61                 "v": {
62                     "title": "Показатель3",
63                 },
64                 "val2": {
65                     "title": "Показатель4",
66                 },
67             },

```

```

68     },
69     "xAxis": "name;lng;lat;region_id",
70     "yAxis": "measures",
71     "measures": [
72         "sum(val):v",
73         "sum(val2)",
74     ],
75     "dimensions": [
76         "name",
77         "lng",
78         "lat",
79         "region_id",
80     ],
81     "pathToMapApi": "glossary.russia_region_borders",
82 },
83 "view_class": "mapareas",
84 },
85 ],
86 "view_class": "map",
87 "title": "Заголовок карты",
88 }

```

Секции `frame` и `title` работают аналогично вышеописанным дэшам (отображение дэша на дэшборде и указание заголовка дэша соответственно). `view_class` необходимо указать `'map'`. Далее указывается массив `children`, в котором каждый элемент массива является отдельным слоем с возможностью скрытия его на карте. Внутри каждого элемента массива `children` используются стандартные опции конфигурации дэшей. В примере выше в массиве `children` указано два элемента: для отображения точек (`'mapdots'`) и для окраски областей карты (`'mapareas'`).

В массиве `dimensions` необходимо указать минимум три размерности:

1. `name` - Название локации
2. `lng` - долгота
3. `lat` - широта

И отложить их на оси X:

```
1 "xAxis": 'name;lng;lat',
```

Вот время как рассматриваемые факты откладываются по оси Y:

```
1 "yAxis": 'measures',
```

Стилизация слоев работает по правилам, аналогичным другим дэшам. В примере выше представлены варианты стилизации (указание заголовков фактам, `stoplight` и т.д.)

`title` указанный для объекта `children` (слоя) указывает название кнопки на карте, демонстрирующий/скрывающий слой на карте

Для `view_class: 'mapareas'` также предусмотрено поле `pathToMapApi` - указывающий путь для выделения областей и их координат (wkt). В случае, если `pathToMapApi` не указан, то по умолчанию используется таблица `'glossary.russia_region_borders'` из базы данных LuxmsBI с картой Российской Федерации, поделенной на регионы и федеральные округа.

В случае необходимости создания собственной таблицы для областей на карте, в таблицы должны присутствовать следующие столбцы: 1. `type` - в зависимости от уровня разбиения карты (доступные значения в таблице `glossary.russia_region_borders`: `region/fdistrict`) 2. `id` - уникальный номер области 3. `title` - название области 4. `wkt` - координаты области в формате wkt

В случае использования типов отображения “Точки” и “Графики” в кубе с данными должны быть прописаны также: - ширина - долгота - Название показателя на карте (например название городов, стран, областей и т.д.)

Пример полученного отображения на карте представлен выше:

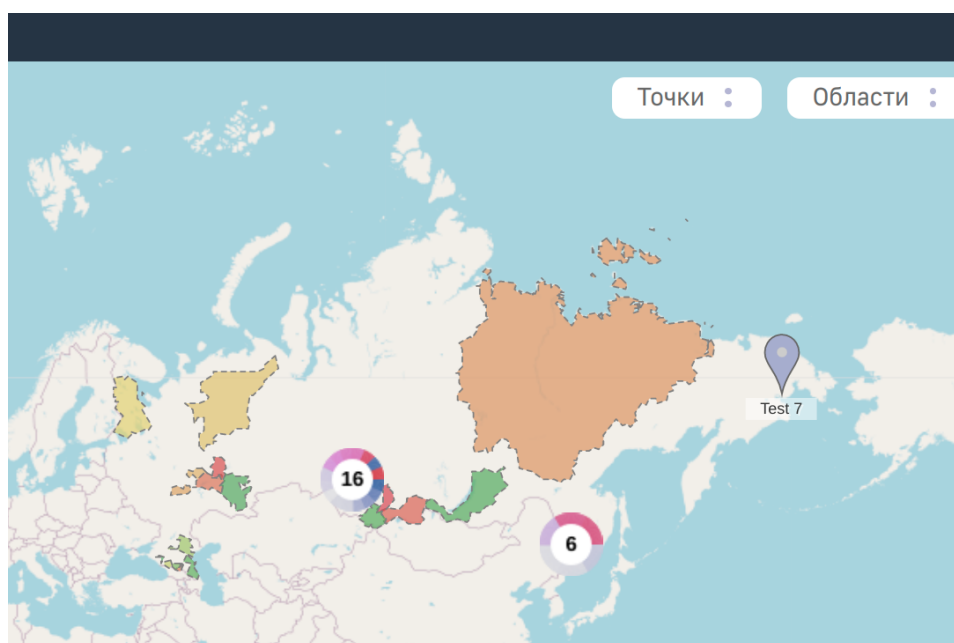


Рис. 2.38 Сконфигурированный дэш “Карта”



В случае необходимости скрытия слоя по умолчанию в `display` необходимо указать `defaultActive` со значением `false`.

Для карты присутствует опция `HoverTooltip` - при нажатии на элемент карты (точку или область) выведется информация со значениями фактов по данному элементу.



В случае указания `onClickDataPoint` для слоя, опция `HoverTooltip` будет отображать информацию при наведении на элемент

Функции из блока `display` также работают в карте, например: стилизация тултипа, указания формата выводимых данных, указания зон (`stoplights`) и т.д.

Для указания размера модального окна для слоев “Графики” можно использовать `displayDrilldown` в блоке `display`.

Пример:

```
1 displayDrilldown: {  
2   top: 'calc(50% - 150px)',  
3   left: 'calc(50% - 200px)',  
4   width: '400px',  
5   height: '300px',  
6 },
```

где, **top** - ориентация модального окна по вертикали, **left** - ориентация модального окна по горизонтали, **width** - ширина модального окна, **height** - высота модального окна

Указывать значения можно в условных единицах, пикселах, процентах, с использованием функции **calc** и т.д.

2.7.1 Доп. опции в блоке `display` для слоя “Точки”

1. **iconType** - Указание варианта отображения точек на карте. Возможные значения: **circle**, **rect**
2. **iconWidth** - Указание ширины иконки, указывающей точки. Задается числом. Число преобразуется в количество пикселей.
3. **iconHeight** - Указание высоты иконки, указывающей точки. Задается числом. Число преобразуется в количество пикселей.
4. **hideTitle** - скрытие подписи точек / графиков на карте
5. **zoomCheck** - отображение заголовка при приближении зума к точке. Задается числом
6. **iconUseFilters** - если указано значение **false**, то иконки точек будут отображены без размывтия. По-умолчанию **true**
7. **disableCluster** - если указано значение **true**, то точки на карте не кластеризируются. По-умолчанию **false**

Ниже представлен пример использования вышеописанных опций и вариант отображения карты с их учетом:

```
1 {  
2   frame: {  
3     h: 7,  
4     w: 11,  
5     x: 0,  
6     y: 0,  
7   },  
8   children: [  
9     {  
10      id: '12:0',  
11      title: 'Тестовый слой',  
12      display: {  
13        iconType: 'circle',  
14        hideTitle: true,
```

```

15     iconWidth: 40,
16     iconHeight: 40,
17     headerStyle: {},
18     disableCluster: true,
19     iconUseFilters: false,
20   },
21   dataSource: {
22     koob: 'mapKoob.mapKoob',
23     style: {},
24     xAxis: 'lat;lng;name',
25     yAxis: 'measures',
26     measures: [
27       'sum(val):sum_val',
28     ],
29     dimensions: [
30       'lat',
31       'lng',
32       'name',
33     ],
34   },
35   view_class: 'mapdots',
36 },
37 ],
38 dataSource: {},
39 view_class: 'map',
40 title: '',
41 }

```

Пример отображения карты с вышепредставленной конфигурацией:

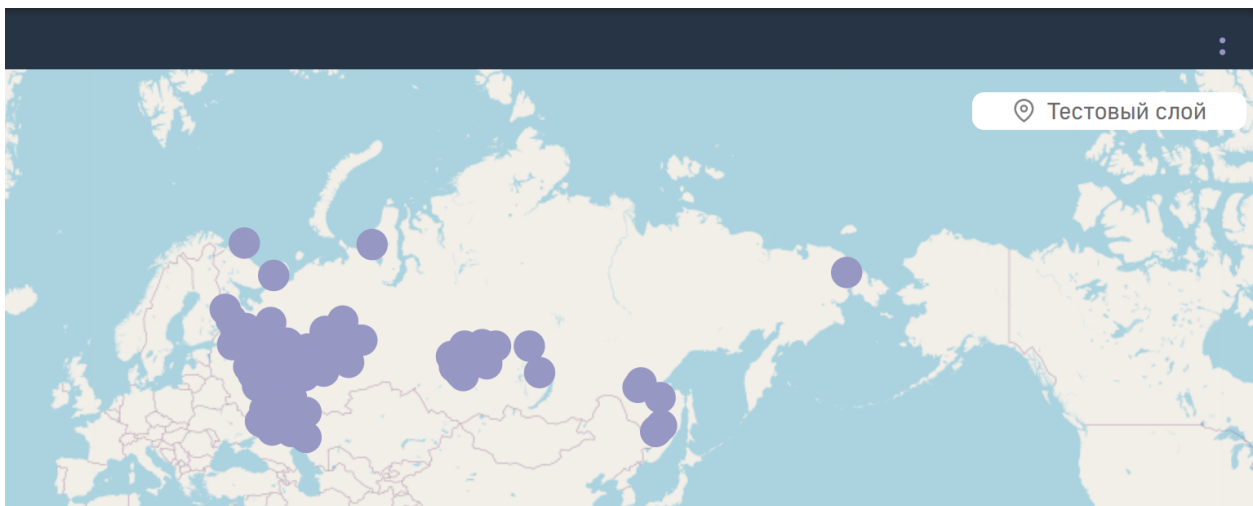


Рис. 2.39 Слой “Точки” после редактирования блока display

2.8 Дэш axes-selector и работа с его конфигурационным файлом

В системе LuxmsBI существует возможность отображения одного дэша с различным набором данных. Для этого можно использовать `axes-selector`.

Рассмотрим функционал данного дэша:

Для добавления возможности переключения необходимо в поле `view_class` указать `axes-selector/` + название визеля для отображения (в нашем случае `axes-selector/bar`). После указания `axes-selector` дэш визуально не помелся. Для добавления выпадающего списка на дэше необходимо в конфигурационном файле указать массивы `xAxes/yAxes`, в которых перечисляется список для отображения переключения данных, откладываемых по оси X и Y соответственно. Для переключения куба в дэше необходимо добавить массив `koobes` в блок `dataSource` с перечислением кубов для выбора. При добавлении в массив `options` опции `MeasuresSelector` добавится дополнительный выпадающий список для выбора отображения фактов (Выбор фактов осуществляется чек-боксами). После добавления массивов `xAxes/yAxes`, `koobes` и сохранения изменений, дэш выглядит следующим образом:

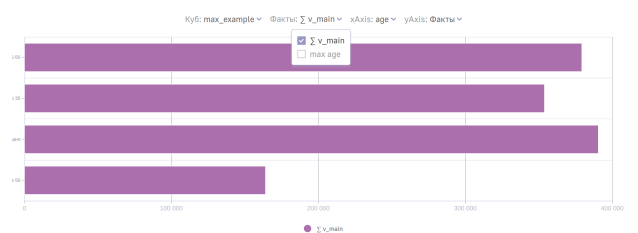


Рис. 2.40 Дэш “Столбики” с возможностью смены вариантов оси для отображения

Конфигурационный файл данного дэша представлен ниже:

```

1 {
2   frame: {
3     h: 4,
4     w: 10,
5     x: 0,
6     y: 0,
7   },
8   options: [
9     'MeasuresSelector',
10  ],
11  dataSource: {
12    koob: 'ch.max_example',
13    style: {},
14    xAxes: [
15      'age',
16      'measures;sex',
17      'sex',
18    ],
19    xAxis: 'age;sex;date',
20    yAxes: [
21      'age',
22      'measures',
23      'date',
24    ],
25    yAxis: 'measures',
26    koobes: [
27      'ch.max_example',

```



```

28     'ch_luxms.max_example',
29 ],
30 measures: [
31     'sum(v_main):sum_v_main',
32     'max(age):max_age',
33 ],
34 dimensions: [
35     'age',
36     'sex',
37     'date',
38 ],
39 },
40 view_class: 'axes-selector/bar',
41 title: '',
42 }

```

2.9 Конфигурация визуеля “Вкладки” (tabs)

Для данного визуеля конфигурация работает аналогично визуелю “Доска”. Конфигурации дочерних элементов прописываются в массиве `children`:

```

1 {
2   frame: {
3     h: 6,
4     w: 6,
5     x: 4,
6     y: 6,
7   },
8   options: [
9     '!TabsPanel',
10    '!TopBar',
11  ],
12  children: [
13    {
14      id: '-1000000:0',
15      title: 'Новая вкладка 1',
16      options: [
17        '',
18      ],
19      dataSource: {},
20      view_class: '1II.bar',
21    },
22    {
23      id: '-1000000:1',
24      title: 'Новая вкладка 2',
25      options: [],
26      dataSource: {},
27      view_class: '1I1.pie',
28    },
29  ],
30  dataSource: {

```

```

31   },
32   view_class: '11I.tabs',
33   title: '',
34 }

```

В данном визеле существует возможность “прослушивания” действий, прописанных в другом дэше в функции `navigate` в `onClickDataPoint`. Например, нам необходимо, чтобы два визеля “Вкладки” переключались одновременно при переключении вкладок на одном. Для этого в первом визеле необходимо использовать `onChangeTab` и прописать для него LPE-выражение с использованием функции `navigate`:

```

1  onChangeTab: 'lpe:navigate("tab", tab)',

```

Данный функционал позволяет дописывать к URL-странице дэшборда открытую в данный момент вкладку:

```

1  http://test.ru/#/ds/ds_1661/dashboards?dboard=105&=&tab=0

```

С помощью данной функции можно для “прослушивания” как переход на определенный дэшборд или дэш.

Во втором визеле “Вкладки” необходимо использовать `activeTab` для “прослушивания” URL:

```

1  activeTab: 'lpe:url("tab") || 0',

```

Прописанное LPE-выражение позволяет визелю отображать вкладку, прописанную в URL, в противном случае, будет отображена первая вкладка.

Полные конфигурационные JSON-файлы для двух визелей “Вкладки” представлены ниже.

Визель для записи открытой вкладки в URL:

```

1  {
2    frame: {
3      h: 6,
4      w: 4,
5      x: 0,
6      y: 6,
7    },
8    options: [
9      '!TopBarMenu',
10     'TabsPanel',
11   ],
12   children: [
13     {
14       id: '981:0',
15       title: 'Новая вкладка 1',
16       options: [],
17       dataSource: {},
18       view_class: '1I1.pie',
19     },
20     {

```

```

21     id: '981:1',
22     title: 'Новая вкладка 2',
23     options: [],
24     dataSource: {},
25     view_class: '1II.line',
26   },
27   {
28     id: '981:2',
29     title: 'Новая вкладка 3',
30     options: [
31       'ActiveTab',
32     ],
33     dataSource: {
34       koob: '',
35     },
36     view_class: '1II.bar',
37   },
38 ],
39 dataSource: {
40   koob: 'oracle.orders_full',
41   style: {},
42   yAxis: 'measures',
43   measures: [
44     'avg(unit_cost):avg_unit_cost',
45   ],
46 },
47 onChangeTab: 'lpe:navigate("tab", tab)',
48 view_class: '1II.tabs',
49 title: '',
50 }

```

Визель для “прослушивания” какая вкладка должна быть открыта с учетом значения в URL:

```

1  {
2    frame: {
3      h: 6,
4      w: 6,
5      x: 4,
6      y: 6,
7    },
8    options: [
9      '!TabsPanel',
10     '!TopBar',
11   ],
12   children: [
13     {
14       id: '-1000000:0',
15       title: 'Новая вкладка 1',
16       options: [
17         '',
18       ],
19       dataSource: {},
20       view_class: '1II.bar',

```

```

21     },
22     {
23       id: '-1000000:1',
24       title: 'Новая вкладка 2',
25       options: [],
26       dataSource: {},
27       view_class: '1I1.pie',
28     },
29   ],
30   activeTab: 'lpe:url("tab") || 0',
31   dataSource: {
32     koob: 'Цитатник.sisis',
33     style: {},
34     yAxis: 'measures',
35     measures: [
36       'count(c):count_c',
37     ],
38   },
39   view_class: '11I.tabs',
40   title: '',
41 }

```

Для дэша “Вкладки” существует возможность скрытия одной из вкладок по определенному условию, прописанному в LPE-выражении по ключу **hidden**:

```

1 {
2   frame: {
3     h: 6,
4     w: 6,
5     x: 4,
6     y: 6,
7   },
8   children: [
9     {
10      id: '-1000000:0',
11      title: 'Новая вкладка 1',
12      hidden: 'lpe:if(100>10,true, false)',
13      options: [
14        '',
15      ],
16      dataSource: {},
17      view_class: '1III.bar',
18    },
19    {
20      id: '-1000000:1',
21      title: 'Новая вкладка 2',
22      options: [],
23      dataSource: {},
24      view_class: '1I1.pie',
25    },
26  ],
27  view_class: '11I.tabs',
28  title: '',

```

29

}

В данном условии можно также использовать функцию `getFilters` описанную выше, в секции `title`.

2.10 Конфигурация управляющего дэша

Для управляющего дэша большинство полей конфигурируется аналогичным образом, как и остальных типов визуалов (`frame`, `dataSource`, `view_class`, `title`) за исключением отсутствия для него массива `measures`.

По-умолчанию в управляющем дэше строковые размерности отображены как `checkbox`'ы,

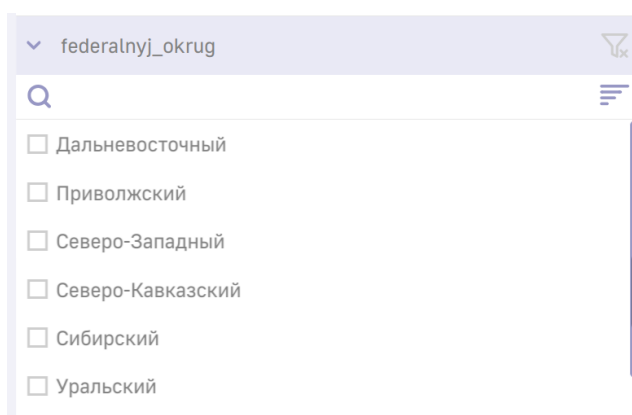


Рис. 2.41 Размерность с типом “строка” в управляющем дэше

размерности типа “период” отображены календарем с выбором диапазона дат для отображения,

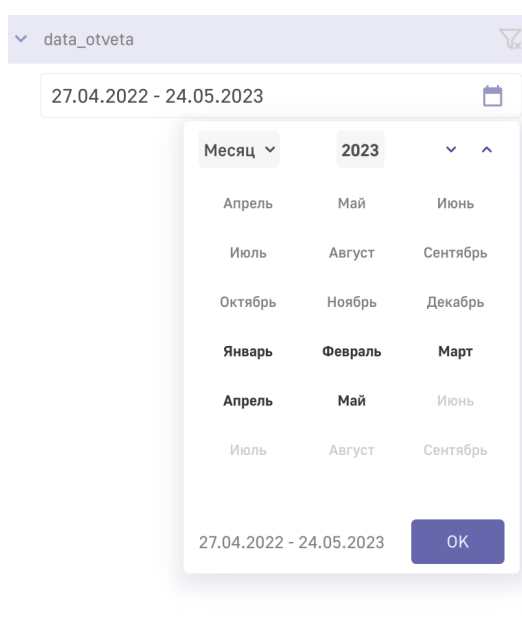


Рис. 2.42 Размерность с типом “период” в управляющем дэше

для численных размерностей используется ползунок, для выбора численного диапазона для отображения.

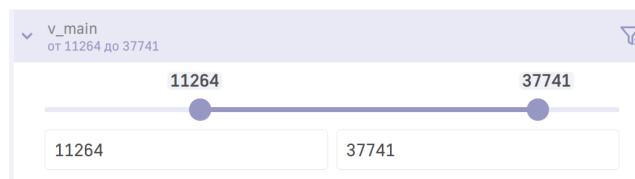


Рис. 2.43 Размерность с типом “число” в управляющем дэше

В случае необходимости отображения типа “число” или “период” в формате строки (чек-боксов) необходимо в блоке **style** для данного показателя указать **type: 'STRING'**. Это может быть полезно в случае использования справочников



Внимание! С версии 9.3.2 luxmsbi-web айди справочников автоматически отображаются как числовые значения. Для отображения заголовков справочника необходимо указать **type: 'STRING'**

Ниже пример:

```
1 style: {
2   sex_id: {
3     type: 'STRING',
4   },
5 }
```

В случае необходимости указания одного показателя размерности, в поле **style** для необходимой размерности указать **type: 'radio'** для представления данной размерности в управляющем дэше в стиле **radiobutton** (возможен выбор только одного показателя)



Для указания конкретной работы радиобаттонов необходимо в конфигурации куба указать **defaultValue** с конкретным показателем

Также существует возможность стилизации шапки и тела управляющего дэша. Для указания цвета фона управляющего дэша необходимо указать ключ **bgColor** со значением HEX цвета. Для стилизации шапки управляющего дэша в объекте **display** необходимо указать поле **headerStyle**, в котором можно прописать следующие поля:

- **color** - указание цвета заголовка
- **backgroundColor** - указание цвета фона шапки
- **textAlign** - ориентация заголовка в шапке (допустимые значения: left, right, center)

Также для размерности можно указать опцию **PanelOpened** - опция, позволяющая раскрыть список показателей для фильтрации по умолчанию.

Для размерностей типа “строка” поисковое поле отображается в случае, если для отображения всех показателей необходим скролл. В случае необходимости отображения поискового поля в любом варианте, необходимо добавить опцию **Search** и **!Search** для постоянного скрытия поля.

Для реализации каскадных фильтров в конфигурации управляющего дэша необходимо прописать массив `hierarchy` с указанием иерархии.

Для размерностей типа “период” также существуют дополнительные опции, прописываемые в блоке `style`:

- `granularity` - массив с перечислением доступных типов периода: день(`day`), неделя(`week`), месяц(`month`), квартал(`quarter`), год(`year`).
- `single` - выбор одного периода, вместо диапазона (по умолчанию - `false`)
- `defaultGranularity` - включенный по умолчанию тип периода

В LuxmsBI существует 2 варианта отображения календаря: простой и сложный. Сложный раскрывается растягиванием за левый нижний угол календаря. В случае необходимости отображения только одного из вариантов, необходимо использовать опцию `calendarType` принимающую 2 варианта значения:

- `simple` - для отображения только простого календаря;
- `complex` - для отображения только сложного календаря.

Ниже представлен вариант отображения сложного календаря:

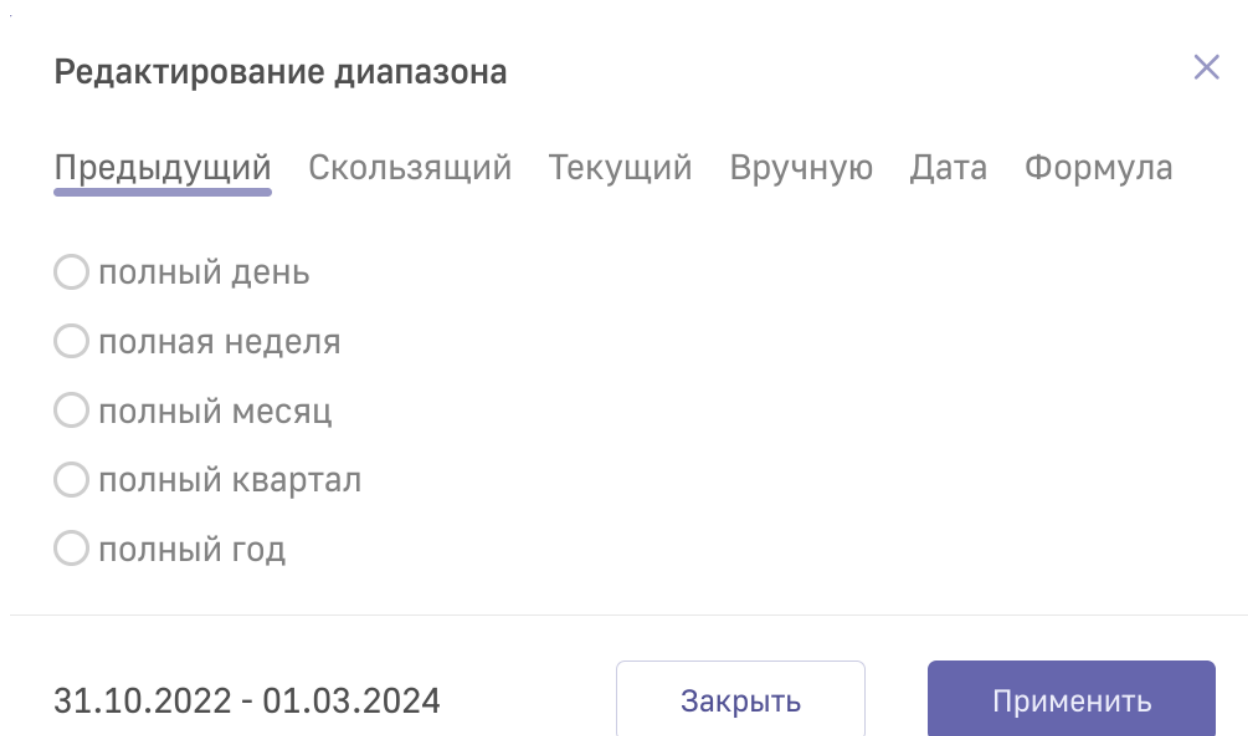


Рис. 2.44 Сложный календарь

Подробности работы со сложным календарем описаны в [Руководстве пользователя LuxmsBI](#)

Пример JSON-конфигурации управляющего дэша со сложным календарем:

```
1 {  
2   "frame": {
```

```

3     "h": 5,
4     "w": 4,
5     "x": 7,
6     "y": 0,
7 },
8 "dataSource": {
9     "koob": 'luxmsbi.public_maxexample',
10    "style": {
11        "dt": {
12            "calendarType": "complex"
13        }
14    },
15    "xAxis": 'dt',
16    "yAxis": 'measures',
17    "dimensions": [
18        'dt'
19    ],
20 },
21 "view_class": 'VizelKoobControl',
22 "title": '',
23 }

```

Ниже представлен пример реализации управляющего дэша в простом календарем и остальными вышеописанными опциями:

```

1 {
2     "frame": {
3         "h": 5,
4         "w": 4,
5         "x": 7,
6         "y": 0,
7     },
8     "display": {
9         "headerStyle": {
10             "color": 'blue',
11             "textAlign": 'center',
12             "backgroundColor": 'red',
13         },
14     },
15     "bgColor": 'blue',
16     "dataSource": {
17         "koob": 'luxmsbi.public_maxexample',
18         "style": {
19             "sex": {
20                 "type": 'radio',
21                 "options": [
22                     "PanelOpened",
23                     "Search"
24                 ]
25             },
26             "dt": {
27                 "calendarType": "simple",
28                 "granularity": [

```



```
29     'month',
30   ],
31   "single": true,
32   "defaultGranularity": 'month',
33 }
34 },
35 "hierarchy": [
36   'region=>gorod=>address',
37 ],
38 "xAxis": 'sex;dt',
39 "yAxis": 'measures',
40 "dimensions": [
41   'sex',
42   'dt'
43 ],
44 },
45 "view_class": 'VizelKoobControl',
46 "title": '',
47 }
```

Управляющий дэш может работать в двух режимах: фильтрация в режиме реального времени и фильтрация после подтверждения действия (кнопка применения фильтров и отмены). Первый режим работы включен по-умолчанию. Чтобы подключить второй вариант работы фильтра необходимо в блоке `options` указать опцию `ApplyFilterButtons`. После указания опции управляющий дэш будет выглядеть следующим образом:

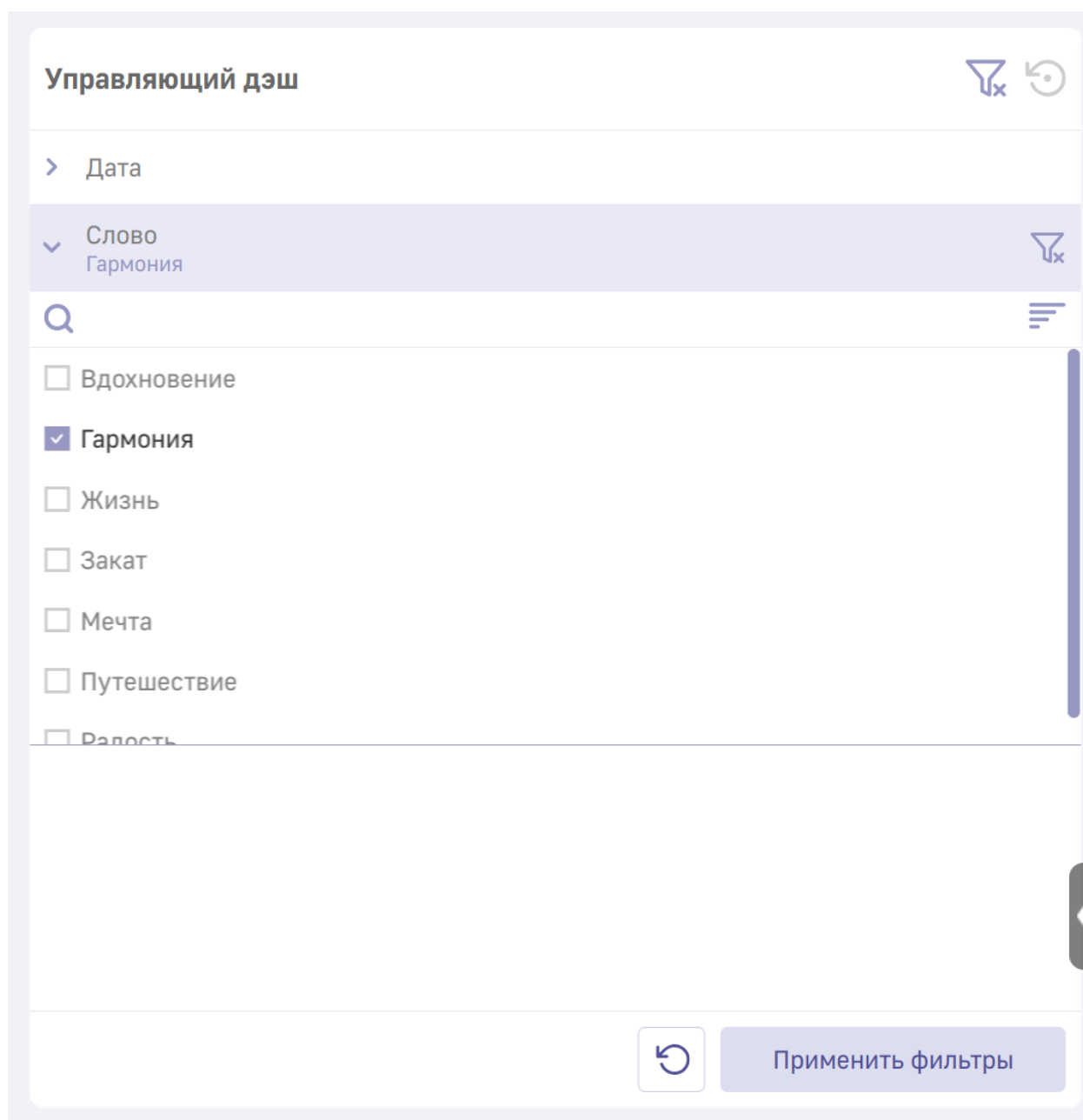


Рис. 2.45 Управляющий дэш в режиме с подтверждением фильтрации

Кнопка “Применить фильтры” становится активной после выбора показателей для фильтрации. Данные будут отфильтрованы только после нажатия на данную кнопку. Для отмены действия нажмите на кнопку “Назад”

Для закрепления управляющего дэша слева можно указать `fixed: true` в блоке `display` :

```
1 display: {  
2   fixed: true,  
3 }
```

2.11 Конфигурация дэша what-if

Если необходимо отображать список переменных для анализа на нескольких дэшах, необходимо использовать массив `vars`.

```
1 vars:[  
2   'a1',  
3   'a2',  
4   ....]
```

Указанные в данном массиве переменные будут отображены в дэше.

3 Использование LPE-выражений для стилизации дэшей

Существуют случаи, когда нам необходимо в конфигурационном файле использовать не конкретное значение, а функции, выполняющие условие / действие, в зависимости от приходящих значений или действий пользователя. Для этого необходимо использовать LPE-выражения. Ниже представлен пример-псевдокод использования LPE-выражений:

```
1 поле_конфигурационного_файла: "lpe:название_функции(аргумент1, аргумент2, ...)"
```

В случае, когда необходимо использовать несколько функций одновременно, существует возможность указания нескольких функций внутри одного LPE-выражения:

```
1 поле_конфигурационного_файла: "lpe:название_функции1(аргумент1, аргумент2, ...);↵
название_функции2(аргумент1, аргумент2, ...)..."
```

Более конкретные примеры использования данных выражений представлено ниже.

В случае, когда нам необходимо стилизовать дэши в зависимости от приходящих значений, существует возможность использования условных LPE-выражений. Для использования условных выражений используются следующие функции:

1. **when** - используется для неограниченного количества условий в выражении. Пример:

```
1 "lpe:when(lpe(public_mv_doc_base1_amount_discount > 0), '#00b9ac', '#93CAFE')"
```

В данном примере представлен случай, когда LPE-выражение представлено внутри LPE-выражения

2. **if** - используется для одного условного выражения. Необходимые для указания аргументы:
2.1 условие, 2.2 значение параметра, если условие истинно 2.3 значение параметра, если условие ложно

Ниже представлен пример использования условных выражений с оператором **if**:

```
1 fontSize: "lpe:if(value<10, '120%', '80%')"
```

В данном примере, в случае, если значение будет меньше 10, то оно отобразится на 20% больше стандартного размера, в противном случае, на 20% меньше.

Данные условные выражения можно использовать для различных полей стилизации. Ниже представлен файл конфигурации дэша “Данные” с использованием условных LPE-выражений:

```

1 {
2   frame: {
3     h: 4,
4     w: 7,
5     x: 0,
6     y: 0,
7   },
8   dataSource: {
9     koob: "luxmsbi.public_mv_doc_base1",
10    style: {
11      measures: {
12        ttt: {
13          color: "lpe:when(lpe(public_mv_doc_base1_amount_discount > 0),↵
14          '#ff0000', '#00ff00')",
15          title: "Индктр",
16        },
17        public_mv_doc_base1_azs_name: {
18          backgroundColor: "lpe:when(lpe(public_mv_doc_base1_amount_discount > ↵
19          0), '#00b9ac', '#93CAFE')",
20        },
21        public_mv_doc_base1_add_info_quantity: {
22          fontSize: "lpe:if(value<10, '120%', '80%')",
23          minWidth: 300,
24          fontStyle: "lpe:if(value<10, 'italic', '')",
25          fontWeight: "lpe:if(value>40, 'bold', 'normal')",
26          textDecoration: "lpe:if(value>40, 'underline', '80%')",
27          backgroundColor: "lpe:if(value = 2, '#ffaabb', 'transparent')",
28        },
29      },
30    },
31    yAxis: "measures;",
32    sortBy: "public_mv_doc_base1_by_day_s,public_mv_doc_base1_trans_time",
33    filters: {},
34    measures: [
35      "concat('Бонусная-', public_mv_doc_base1_card_type):a1",
36      "if(public_mv_doc_base1_rc='P', 'Платиновая', if(public_mv_doc_base1_rc=↵
37      'J', 'Золотая', if(public_mv_doc_base1_rc='R', 'Серебряная', 'Отсутствует')):↵
38      a10",
39      "if(public_mv_doc_base1_amount_discount > 0, '↑', '↓'):ttt",
40      "public_mv_doc_base1_add_info_quantity",
41      "public_mv_doc_base1_amount_discount",
42    ],
43    dimensions: [
44      "public_mv_doc_base1_goods_code_type",
45    ],
46    view_class: "koob-table-simple",
47  }
48 }

```

Ниже на изображении представлен дэш с вышеописанным конфигурационным файлом:

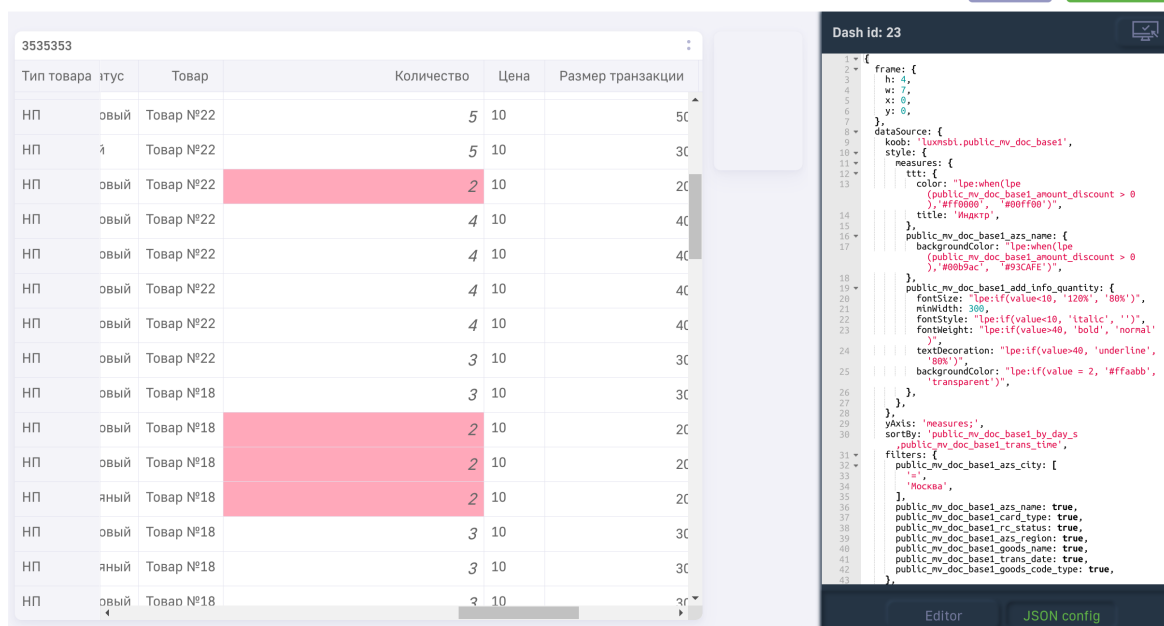


Рис. 3.1 Дэш “Данные” со стилизацией столбцов с использованием LPE-выражений



В данный момент, стилизация с использованием LPE-выражений работает для дэшей `tableP`, `koob-table-simple` и `label`

Для двумерных дэшей (отложенные в системе координат) существует возможность выводить вместо значений факта - значение, полученное по формуле, написанной в LPE-выражении. Для этого можно использовать поле `badgeTitle`, пример использования которого представлен ниже:

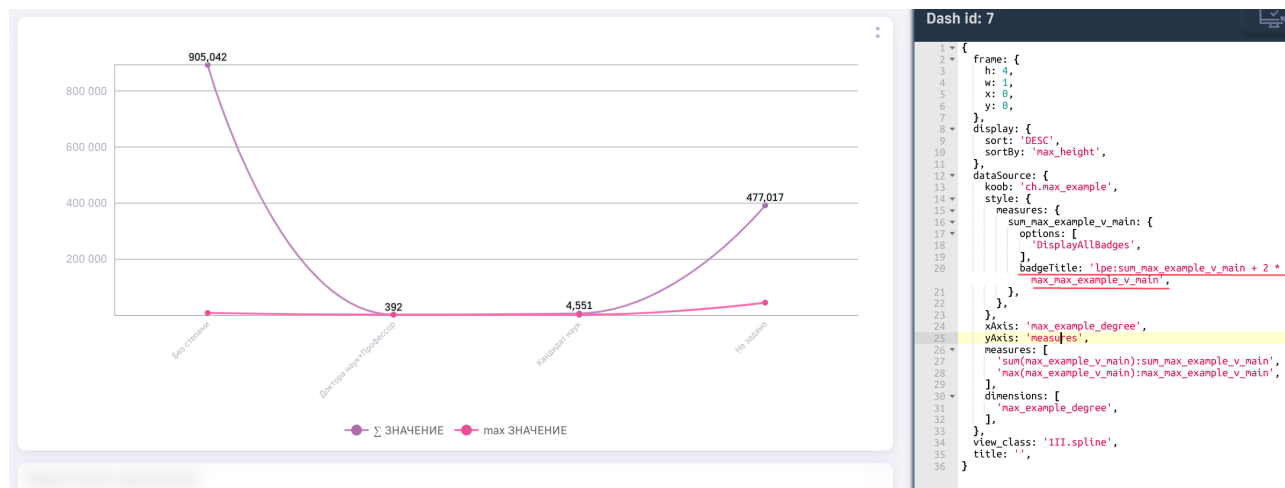


Рис. 3.2 Использование LPE-выражения в поле `badgeTitle`

На примере выше, в выводимых значениях предомostrировано значение, рассчитанное по формуле: $1 \text{ факта} + 2 * 2 \text{ факт}$



Внимание! Поле `badgeTitle` будет работать только при указании в массиве `options` значения `DisplayAllBadges`

3.0.1 Функции, доступные внутри выражений if и when

Внутри выражений вы можете использовать:

- математические операнды `+`, `-`, `*`, `/`, `(`, `)`;
- операторы сравнения `<`, `>`, `<=`, `>=`, `=`, `!=`;
- Значения `true`, `false`, `#t(true)`, `#f(false)`, `null`, `NIL(null)`;
- Логические операторы `and`, `or`, `&&(and)`, `||(or)`, `!`;
- Функцию `cond()`;

Пример:

```
1 color: 'lpe:when(cond(1>null), "#ff0000", "#00ff00")'
```

- специальную переменную `value` чтобы изменить параметры только некоторых значений.

Пример:

```
1 color: 'lpe:if(value>989, "#00f0f0", "#0000ff")'
```

Результат такой настройки

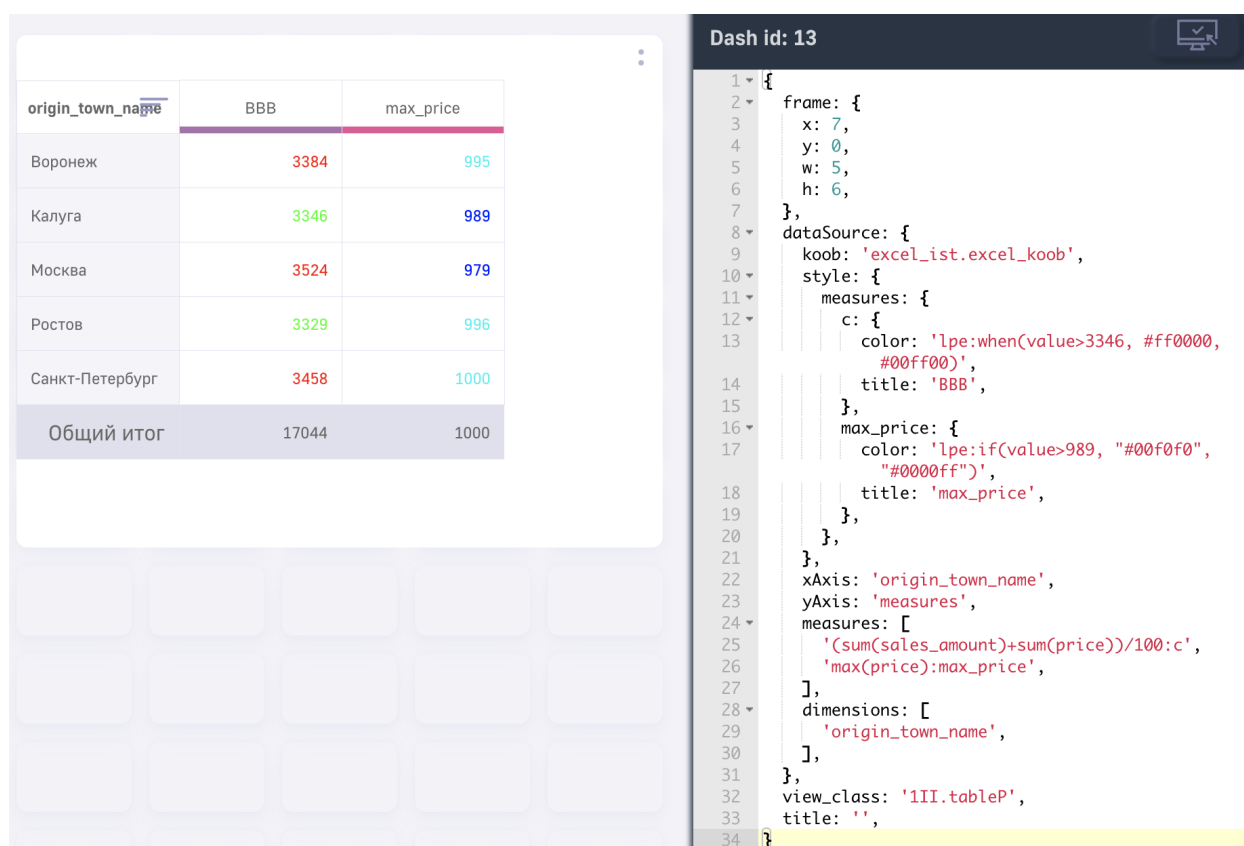


Рис. 3.3 Использование `value` в LPE в цвете дэша

4 Руководство по конфигурации дэшборда

В данном разделе представлено описание ключей и значений стандартной JSON-конфигурации для дэшборда. Для того, чтобы отредактировать конфигурацию дэшборда необходимо перейти в режим разработчика (edt-entities). Для перехода в данный режим откройте необходимый для редактирования атлас, в URL-пути до вашего дэшборда вместо **dashboards** укажите ключевое слово **edt-entities**. Ниже пример:

Было:

```
1 http://localhost:3000/#/ds/ds_1/dashboards?dboard=1
```

Стало:

```
1 http://localhost:3000/#/ds/ds_1/edt-entites?dboard=1
```

После изменения адреса откроется режим разработчика. В данном руководстве описана работа одной из доступных вкладок в данном режиме - **dashboards**. Нажмите на соответствующую кнопку:

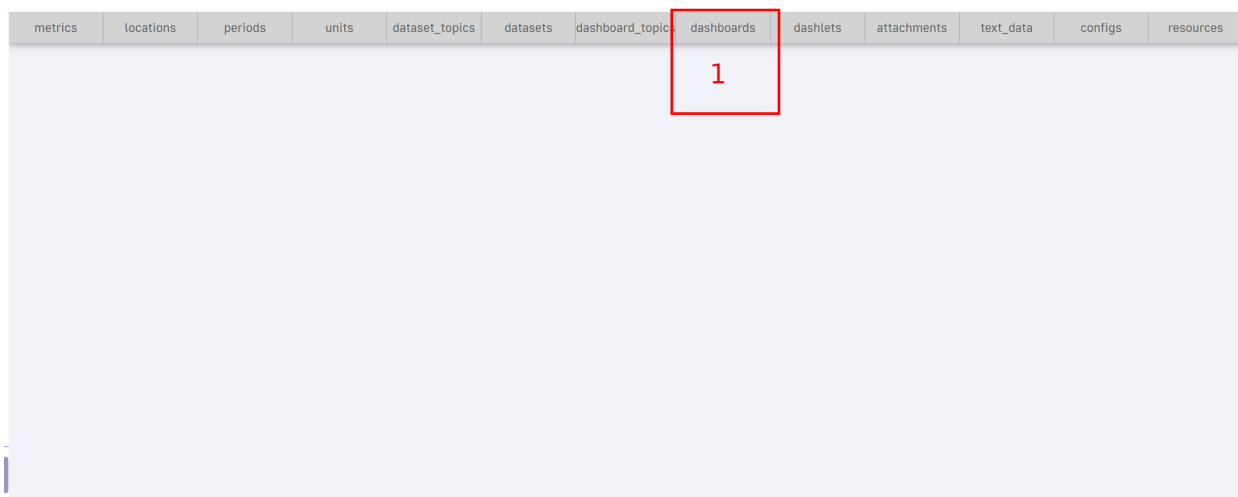


Рис. 4.1 Вкладка **dashboards** в режиме разработчика

В левом меню выбрать необходимый для редактирования атлас:

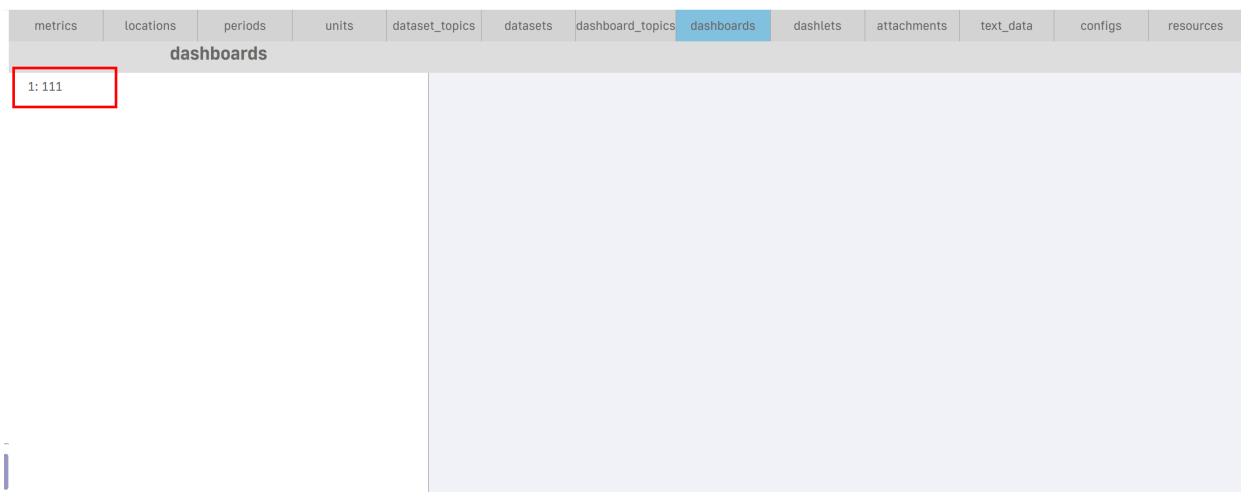


Рис. 4.2 Список дэшбордов для редактирования

После выбора дэшборда, в основном блоке будет представлена конфигурация дэшборда:

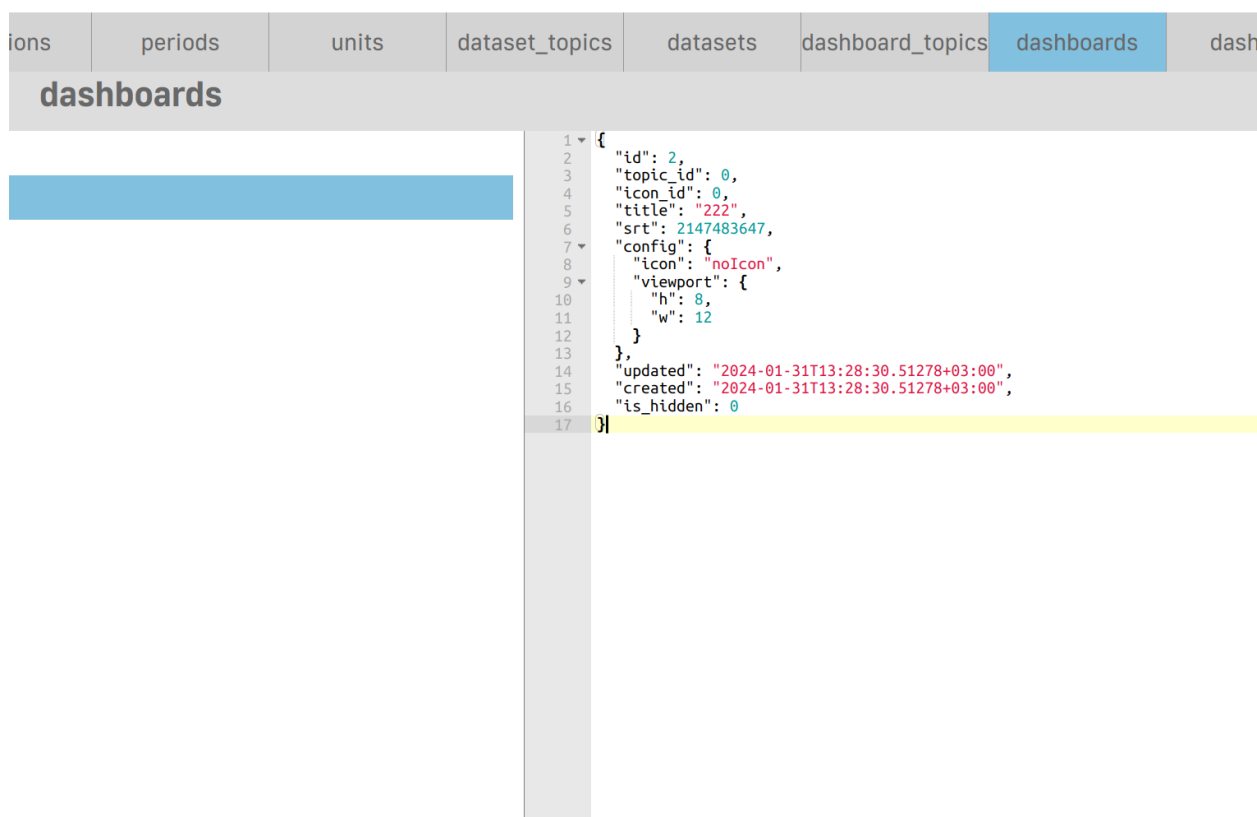


Рис. 4.3 Конфигурация дэшборда

В представленном JSON-файле мы рекомендуем редактировать только объект **config** для корректной работы дэшборда и атласа в целом.

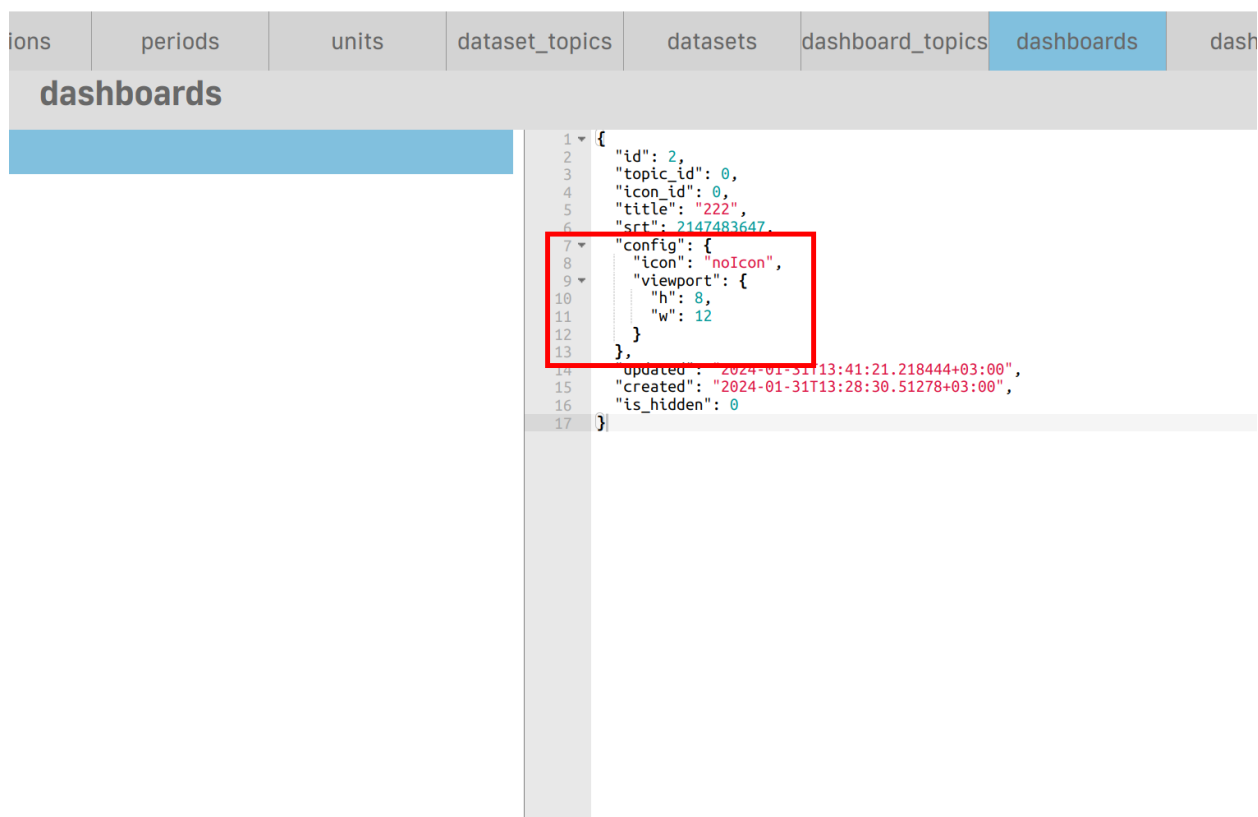


Рис. 4.4 Объект config для редактирования дэшборда

В данном объекте по-умолчанию представлены следующие поля:

1. **icon** - указание иконки для дэшборда, выбирается из доступного списка. (Рекомендуется выполнять данную манипуляцию через интерфейс режима редактирования дэшборда) по-умолчанию значение "noIcon";
2. **viewport** - объект, с указанием количества ячеек для отображения дэшборда по вертикали (**h**) и горизонтали (**w**). Задается числами.

Помимо вышеописанных пунктов, существует возможность указать размер ячейки сетки дэшборда в пикселах, для этого используются поля **cellWidth** и **cellHeight**. Ниже представлен пример указания данных значений:

```

1 "config": {
2   "icon": "noIcon",
3   "viewport": {
4     "h": 8,
5     "w": 12
6   },
7   "cellWidth": 150,
8   "cellHeight": 150
9 },

```

После редактирования конфигурации необходимо нажать кнопку сохранения изменений:

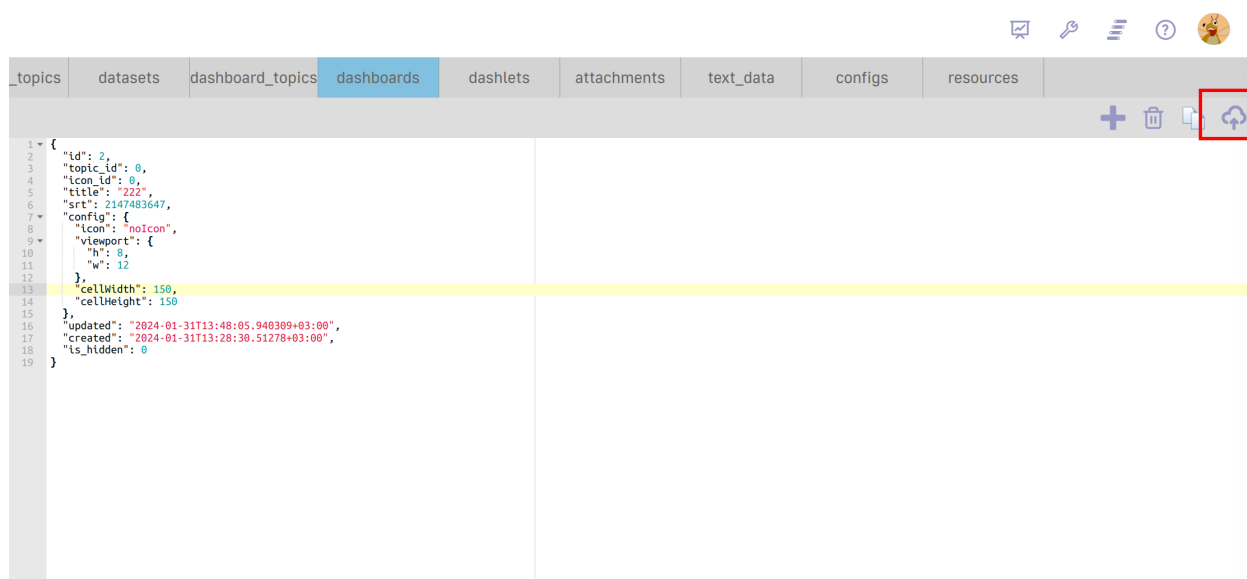


Рис. 4.5 Сохранения изменений в конфигурации

После сохранения изменений ячейка в сетке будет размером 150x150:

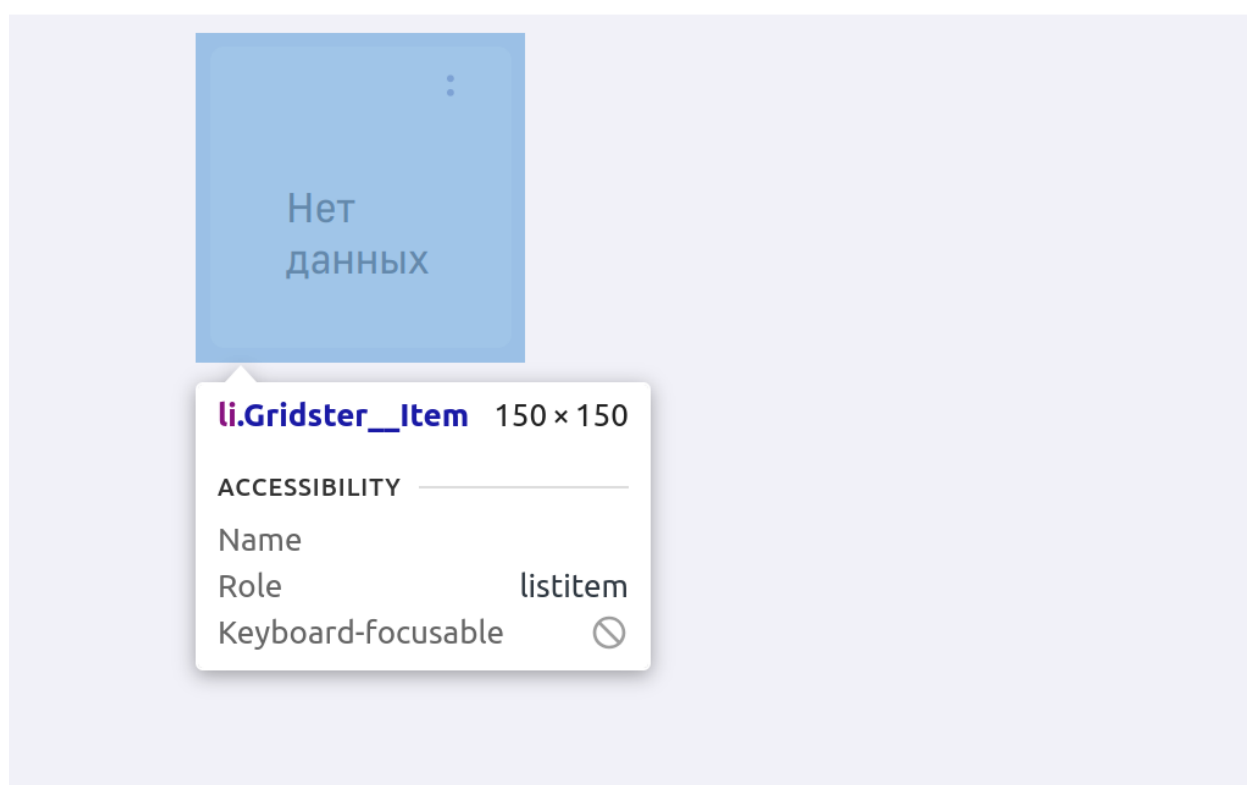


Рис. 4.6 Размер ячейки в соответствии с указанием значений в конфигурации

5 Таблица изменений

Дата изменения	Пункт	Описание изменения
02.04.2024	Типы дэшей в Luxms BI	Добавлены статистические дэши: гистограмма и ящик с усами
29.03.2024	Поле bgImage	Добавлено описание
22.03.2024	Опции для визеля koob-table-simple (Плоская таблица)	Удалено описание устаревшей опции overflow
18.03.2024	Опции для визеля koob-table-simple (Плоская таблица)	Добавлено описание опции groupColumns
11.03.2024	Секция onClickDataPoint	Добавлено описание функции setKoobFilter
05.02.2024	Расширенные настройки виджетов	Вставлена другая картинка в раздел о forecast
29.02.2024	Конфигурация управляющего дэша	Добавлено описание calendarType
28.02.2024	Конфигурация управляющего дэша	Добавлено описание type: 'STRING'
22.02.2024	Расширенные настройки виджетов	Добавлены опции для управляющего дэша
20.02.2024	Поле description	Добавлено описание работы данного поля
20.02.2024	1.2.15 Опции для визеля tableP (Таблица с группировкой)	Переписана логика работы указания ширины столбцов
20.02.2024	1.2.13 Опции для визеля koob-table-simple (Плоская таблица)	Переписана логика работы указания ширины столбцов
16.02.2024	1.2.13 Опции для визеля koob-table-simple (Плоская таблица)	Подробное описание логики работы указания ширины столбцов в плоской таблице
16.02.2024	1.2.8 Секция options	Добавлено описание опции !Expanded
14.02.2024	Расширенные настройки виджетов	Актуализирована информация по filters

Дата изменения	Пункт	Описание изменения
13.02.2024	Расширенные настройки виджетов	Добавлена информация о стилизации заголовка
13.02.2024	Примеры конфигурации дэшей в LuxmsBI	Добавлено описание опции ApplyFilterButtons
09.02.2014	Примеры конфигурации дэшей в LuxmsBI	Добавлена информация о дэше Лэйбл
06.02.2024	1.2.3 Секция forecast	Добавлено описание блока forecast
31.01.2024	4 Руководство по конфигурации дэшборда	Добавлен раздел
30.01.2024	1.2.7 Секция options	Добавлено описание Fixed
16.01.2024	1.2.12 Опции для визеля koob-table-simple (Плоская таблица)	Добавлено описание Pin , удалено описание isPin . Добавлено примечание для опции wordBreak
16.01.2024	2.10 Конфигурация управляющего дэша	Добавлено описание опции Search и обновлены изображения
08.12.2023	1.2.5 Секция echart	Изменена логика стилизации осей
08.12.2023	2.10 Конфигурация управляющего дэша	Добавлена опция granularity для размерности типа “период”
08.12.2023	1.2.12 Опции для визеля koob-table-simple/tableP	Отредактировано описание опции formula

