

Конструктор отчетов

Мастер "конструктор отчетов" позволяет собирать произвольные отчеты в пользовательском режиме, без программирования.

Ниже подробное описание каждого шага.

Выбор отчета

На данном этапе выбирается отчет для редактирования (если редактируется существующий).

Если требуется создать новый надо установить соответствующий флаг.

Флаг "Скопировать конфигурацию существующего" создаст новый отчет с полной копией конфигурации выбранного отчета.

Основные данные

На данном этапе настраиваются основные параметры отчета.

Для разработки отчета рекомендуется ставить флаг "Только для суперпользователей", чтобы пользователи не видели его, пока отчет не будет готов.

Основная таблица

На данном этапе выбирается таблица, из данных которой будет собираться отчет.

Таблицы соответствуют пунктам из главной страницы админки.

Основные таблицы, которые используются для отчетов:

- **Билет** - основная таблица, если требуется детальная информация по билетам.
- **Рейс** - если требуется агрегированная информация по рейсам. В данной таблице есть все счетчики по этому рейсу (кол-во пассажиров, кол-во багажных, кол-во льготников и т.д). Если не требуется детализация по каждому билету предпочтительно использовать именно эту таблицу.
- **Движения денежных средств в кассах ККМ** - данная таблица используется, если требуется разделять дату продажи и дату возврата билетов, т.к эта таблица содержит 2 отдельные записи для продажи и возврата.

Поля для отчета

На данном этапе выбираются все поля, которые будут использоваться в отчете (для вывода, фильтрации и т.д).

Помимо выбора полей из отчета, здесь также есть возможность создавать вычисляемые поля (добавляются через "плюс" в среднем дереве).

Вычисляемые поля позволяют гибко создавать колонки в зависимости от требуемых условий.

Примеры выражений для вычисляемых полей:

Поле "Доля перевозчика билет" вычислит долю перевозчика исходя из его процента, указанном в справочнике:

```
python (100 - F('contractor__calculation_ticket')) * F('summ_ticket') / 100
```

Если требуется объединить все междугородние виды маршрутов в один:

```
python Case(When(route__type__title__iexact='Пригородный',then=Value('Пригород')), default=Value('Межгород'), output_field=CharField())
```

С вычисляемыми полями можно дальше работать как с обычными полями.

Фильтры (опционально)

На данном этапе задаются фильтры, которые отсортируют данные по нужным требованиям.

Например, если требуется, чтобы в таблице билетов были выведены только билеты явившихся пассажиров, надо выбрать соответствующее поле из дерева и задать для него значение "да".

Группировка (опционально)

Если требуется сгруппировать данные (подсчитать суммы, кол-во и т.д), то на данном этапе нужно выбрать поля:

- Поля группировки - разрез, по которым будет подсчитана агрегация.
- Поля агрегации - показатели (счетчики), по которым нужно подсчитать сумму, кол-во и т.д);

Сортировка (опционально)

На данном этапе задаются поля, по которым требуется отсортировать данные (по возрастанию или убыванию).

Важно! Если на этапе "Группировка" выбраны какие-либо поля, то для сортировки нужно выбирать именно их.

Структура отчета

На данном этапе выбираются поля, которые будут выведены в таблице для пользователя (папка колонки), а также пользовательские фильтры.

Важно! В качестве пользовательского фильтра нужно обязательно выбрать дату с типом "в диапазоне", и проставить для нее флаг "обязательный", иначе выборка будет идти по всей таблице.

Для подведения итогов по колонке нужно установить соответствующий флаг в настройке колонки.

Также в настройке "Отчет" можно задать требуемый заголовок отчета (по умолчанию берется из наименования отчета).