ВНЕШНИЕ СПРАВОЧНИКИ

Требуется отдельная лицензия. Лицензия позволяет подключить любое количество справочников — как готовых из перечисленных ниже, так и собственных справочников заказчика.

Мы подготовили набор отраслевых справочников, которые можно подключить и использовать:

- кем выдан паспорт;
- почтовые отделения;
- налоговые инспекции;
- мировые суды;
- марки автомобилей;
- страны;
- валюты;
- ОКВЭД;
- ОКПД;
- и другие.

Если хотите подключить собственный справочник — читайте дальше, как это сделать.

Подготовить справочник

Справочник — это обычный CSV-файл:

- Плоский: даже если в исходном справочнике есть иерархия, перед подключением к подсказкам он должен быть переведен в плоский вид.
- С заголовочной строкой: она используется в настройках.
- В названии самого справочника и заголовках полей можно использовать только латинские буквы и подчёркивание.
- В кодировке UTF-8 (без BOM).

Особенности и ограничения

Если надо, чтобы подсказки возвращали вычисляемые поля (например, «ФИО», составленное из фамилии, имени, и отчества) — эти столбцы тоже должны быть в справочнике. Сами подсказки их делать не умеют.

Если нужен поиск по синонимам (хендай, хенде hyundai) или с учетом опечаток (менежер менеджер) — варианты написания нужно перечислить в отдельном столбце справочника через запятую. Подсказки не умеют автоматически искать с учетом опечаток или придумывать синонимы.

Если одни записи справочника более «весомые», чем другие (например, головные организации против филиалов), то веса должны быть предрассчитаны для всех записей и вынесены в отдельный столбец — сами подсказки этого делать не будут.

Подсказки используют префиксный поиск. Вот что это значит на примере. Допустим, есть запись справочника «Волкова Татьяна Анатольевна». Тогда:

- запрос «волк» найдет запись;
- запрос «волкова» найдет запись;
- запрос «волкова тат» найдет запись;
- запрос «волкова татьяна» найдет запись;
- запрос «волк тат» не найдет запись;
- запрос «волк татьяна» не найдет запись.

Таким образом, в тексте запроса допускается частичное совпадение только по последнему слову, по предыдущим словам совпадение должно быть полным.

По умолчанию значения полей подсказки трактуют как строки, даже если по факту это числа или даты.

Пример справочника

hid	name	surname	patronymic	fullname	email	phone	inn	salary	position	department
1001	Волкова	Татьяна	Анатольевна	Волкова Татьяна Анатольевна	volkova.tatyana@evercorp.ru	+7 926 387-34-23	5601046254 32	50000	Менеджер по закупкам	Отдел закупок

1002	Барчук	Валерий	Николаевич	Барчук Валерий Николаевич	barchuk.valeriy@evercorp.ru	+7 495 234-13-98	4908754134 90	45000	Менеджер по продажам	Отдел корпоративных продаж
1003	Жукова	Кристина	Олеговна	271307275938	ceo@evercorp.ru		2713072759 38	90000	Директор	
...										

Настроить конфигурацию

Конфигурация — это YAML-файл в кодировке UTF-8 (без BOM), который описывает, как подсказки работают со справочниками. Проще всего объяснить его на примере:

```
# название справочника
# наш пример — справочник сотрудников
name: employees
# настройки чтения CSV файла
# необязательно, значения по умолчанию описаны ниже
csv:
  quote: '"'
  delimiter: ','
  endOfLine: '\n'
  encoding: UTF-8
# какие есть поля и как их использовать
fields:
  # Исправлять латинскую раскладку, по умолчанию true
  switch_layout: false
  # какие поля участвуют в поиске по идентификаторам
  # необязательно, по умолчанию не задано
  ids:
    hid: ~
    inn: ~
  # какие поля участвуют в полнотекстовом поиске (обязательно хотя бы одно)
  # для полей можно указать параметр boost — вес для ранжирования
  #   чем больше вес, тем «весомее» считается совпадение по этому полю
  #   например, если вес по fullname = 10, а по email = 1,
  #   то совпадение по ФИО считается весомее, чем по адресу эл. почты
  #   по умолчанию вес = 1
  search:
    fullname:
      boost: 10
    email: ~
    phone: ~
  # по каким полям фильтровать
  # необязательно, по умолчанию не задано
  filter:
    department: ~
  # по какому полю ранжировать
  # необязательно, по умолчанию не задано
  # чем больше значение, тем выше ранжируется соответствующая запись справочника
  # если указано несколько полей, то значения умножаются
  # по умолчанию вес = 1
  boost:
    salary: ~
  # по какому полю сортировать
  # необязательно, по умолчанию равно настройке fields.value + ALPHA_NUMERIC
  # принцип сортировки:
```

```
#   ALPHA_NUMERIC - по алфавиту (с честной сортировкой чисел)
#   TOKEN_COUNT - по количеству слов (чем меньше слов, тем выше позиция)
sort:
  fullname:   TOKEN_COUNT
  department: ALPHA_NUMERIC
# какое поле показывать в списке (suggestion.value, обязательно)
value: fullname
# какое поле содержит полное значение одной строкой
# необязательно, по умолчанию равно настройке fields.value
unrestricted_value: fullname_with_position
# какие поля возвращать в объекте подсказки (suggestion.data)
# необязательно, по умолчанию не задано (suggestion.data == null)
data:
  name: ~
  surname: ~
  patronymic: ~
  email: ~
  phone: ~
  position: ~
  department: ~
```

Начиная с версии 19.11 поддерживается поиск ближайшего объекта в справочнике по географическим координатами. Чтобы он заработал, в справочнике должны быть поля с широтой и долготой. Их необходимо указать в конфигурации:

```
fields:
    ...
    geolocate:
        latitude: {название поля с широтой}
        longitude: {название поля с долготой}
    ...
```

Начиная с версии 19.11 можно явно указать тип полей справочника. Поддерживаются типы: int, float, bool, str.

```
fields:
  ...
  data:
    postal_code: str
    is_closed: bool
    address_qc: int
    geo_lat: float
    ...
```

По умолчанию для *boost* и *geolocate* полей устанавливается тип *float*, для всех остальных полей – *string*. Для boost-полей тип может быть только *float* или *int*, а для geolocate-полей тип может быть только *float*.

Подключить конфигурацию и справочник

Конфигурация:

1. Сохранить конфигурацию в файл `{name}.yaml`, где `{name}` — название справочника, как указано в самом первом параметре конфигурации.
   Для нашего примера это `employees.yaml`
2. Создать на сервере каталог `/SGT_ROOT/configuration/outward/` и скопировать в него файл конфигурации.
   Для нашего примера итоговый путь к файлу будет `/data/configuration/outward/employees.yaml`

Справочник:

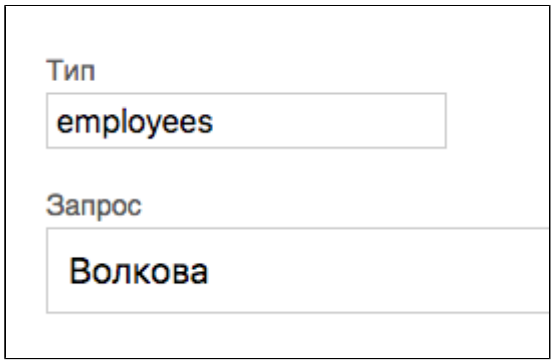
1. Сохранить справочник в файл `{name}.csv`, где `{name}` — название справочника, как указано в конфигурации.
   Для нашего примера это `employees.csv`
2. Создать на сервере каталог `/SGT_ROOT/dictionaries/{name}/` и скопировать в него файл справочника.
   Для нашего примера итоговый путь к файлу будет `/data/dictionaries/employees/employees.csv`

Чтобы подсказки увидели, справочник, перезапустите их либо используйте API загрузки.

Проверить на демо-странице

Работу подсказок можно проверить на демо-странице по адресу `http://СЕРВЕР:ПОРТ/suggestions/outward`

В поле «Тип» укажите название справочника, после этого подсказки заработают:



Использовать через API

Поиск по справочнику

Аналогично «родным» справочникам:

```
POST /suggestions/api/4_1/rs/suggest/employees HTTP/1.1
Host: suggestions.evercorp.ru
Content-Type: application/json

{
  "query": "Василий"
}
```

Фильтрация

Поддерживается стандартный параметр count и фильтрация (если настроена фильтрация, параметр filter в конфигурации справочника):

```
POST /suggestions/api/4_1/rs/suggest/employees HTTP/1.1
Host: suggestions.evercorp.ru
Content-Type: application/json

{
  "query": "Василий",
  "filters": [{"department": "Маркетинг"}, {"department": "ИТ"}]
  "count": 5
}
```

Поиск по идентификатору

Работает, если настроен параметр ids в конфигурации справочника:

```
POST /suggestions/api/4_1/rs/findById/employees HTTP/1.1
Host: suggestions.evercorp.ru
Content-Type: application/json

{
  "query": "1024"
}
```

Поиск ближайшего объекта

Работает в версиях 19.11+, если настроен параметр geolocate в конфигурации справочника. Работает аналогично геолокации для адресов – возвращает ближайшие объекты по заданным координатам:

```
POST /suggestions/api/4_1/rs/geolocate/postal_unit HTTP/1.1
Host: suggestions.evercorp.ru
Content-Type: application/json

{
  "lat": 55.763938,
  "lon": 37.637281,
  "radius_meters": 10000,
  "count": 15
}
```

Геолокация работает не только через POST, но и через GET:

```
```

```
GET /suggestions/api/4_1/rs/geolocate/postal_unit?lat=55.763938&lon=37.637281 HTTP/1.1
Host: suggestions.evercorp.ru
```

Параметры запроса:

Параметр	Обязательный?	Описание
`lat`	да	Географическая широта
lon	да	Географическая долгота
count	нет	Количество возвращаемых подсказок (по умолчанию — 10, максимум — 20).
radius_meters	нет	Радиус поиска в метрах (по умолчанию – 100, максимум – 10000)

Использовать через jQuery-плагин

В плагине всё как обычно, только в поле `type` указывается название справочника:

```
$("#manager").suggestions({
  type: "employees",
  ...
});
```

Если конфигурация справочника допускает, можно использовать фильтрацию:

```
$("#manager").suggestions({
  type: "employees",
  params: {
    filters: [
      { "department": "Отдел закупок"}
    ]
  }
});
```