ВНЕШНИЕ СПРАВОЧНИКИ — КОНФИГУРАЦИЯ

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

```yaml
# название справочника
# наш пример — справочник сотрудников
name: employees
# описание справочника (21.7+)
# необязательно, используется только для показа в админке
meta:
  title: Сотрудники
  description: Справочник сотрудников компании
  source: Отдел кадров
# 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*.