Интеграция с классификатором адресов РФ (КЛАДР)

Выпуск 2.0.0

БАРС Груп

17 March 2014

Оглавление

Пакет m3-kladr предназначен для интеграции программных продуктов на базе платформы М3 с Классификатором адресов Российской Федерации (КЛАДР)

Содержание:

Глава 1

Установка

Пакет m3-kladr подключается как приложение М3.

Для функционирования, также требуется установка дополнительных библиотек.

1.1 Необходимые библиотеки

- dbfpy==2.2.5
- m3-core>=2.0
- m3-ext3>=2.0

1.2 Установка с помощью pip

Установите пакет `m3-kladr` из репозитория пакетов компании `БАРС Груп`

```
pip install m3-kladr -i https://<PyPI_сервер_БАРС_Груп>
```

В этом случае будут установлены все необходимые пакеты.

Теперь можно приступать к настройке М3-приложения.

1.3 Установка из архива

Скачайте и распакуйте архив модуля https://bitbucket.org/barsgroup/m3-kladr/downloads

1.4 Установка из Mercurial

Клонируйте исходный код модуля из репозитория

```
hg clone https://bitbucket.org/barsgroup/m3-kladr
```

1.5 Установка из каталога

Установка

```
python setup.py install
```

1.6 Настройка

Подключение пакета осуществляется в файле `settings.py` приложения. Необходимо добавить имя пакета в раздел `INSTALLED_APPS`.

```
INSTALLED_APPS = (
    kladr,
)
```

Также необходимо добавить таблицы в СУБД. Если в Вашем проекте используется модуль South запустите команду:

```
python manage.py migrate kladr
```

В противном случае запустите команду:

```
python manage.py syncdb
```

Глава 2

Заполнение базы КЛАДРа

Для использования КЛАДРа необходимо заполнить базу данными. Для этого необходимо:

1. Загрузить актуальную базу с сайта ГНИВЦ [ссылка]
2. Распаковать полученный архив.
3. Перейти в папку проекта (где расположен файл `manage.py`)
4. Запустить команду:

```
python manage.py fill_kladr <параметры>
```

2.1 Параметры команды `fill_kladr`

- `--dbf_path` – путь к папке с файлами базы КЛАДР; если параметр не указан, загрузка производится из текущей папки
- `--noclear` – перед запуском команды не очищаются существующие данные
- `--force` – команда не будет задавать вопросов
- `--update` – существующие записи будут обновлены

2.2 Примеры использования

```
# Загрузка в пустую базу
python manage.py fill_kladr --dbf-path /var/kladr/

# Обновление базы
python manage.py fill_kladr --dbf-path /var/kladr/ --noclear --update
```

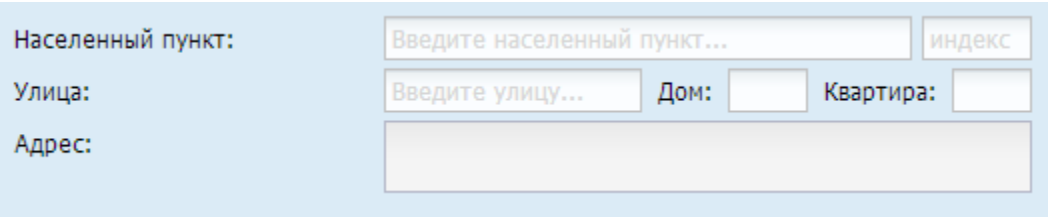
Глава 3

Использование КЛАДР в М3-приложениях

В приложениях работа с КЛАДР заключается в выборе адреса. Для добавления на форму необходимо воспользоваться классом `kladr.addrfield.ExtAddrComponent`:

```
from kladr.addrfield import ExtAddrComponent
# ***********

class SomeWindow():

    def __init__(self):

        ur_address = ExtAddrComponent()

        # *********

        self.form.extend([
            # *********
            ur_address,
        ])
```

По умолчанию компонент выглядит следующим образом:



3.1 Настройка отображения компонента `ExtAddrComponent`

С помощью атрибута `level` устанавливается количество отображаемых полей. Перечень значений атрибута `level`:

- ExtAddrComponent.PLACE - отображается только поле населенного пункта
- ExtAddrComponent.STREET - отображаются поля населенного пункта и улицы
- ExtAddrComponent.HOUSE - отображаются поля населенного пункта, улицы и номера дома

- ExtAddrComponent.FLAT - отображаются поля населенного пункта, улицы, номера дома и номера квартиры (по умолчанию)

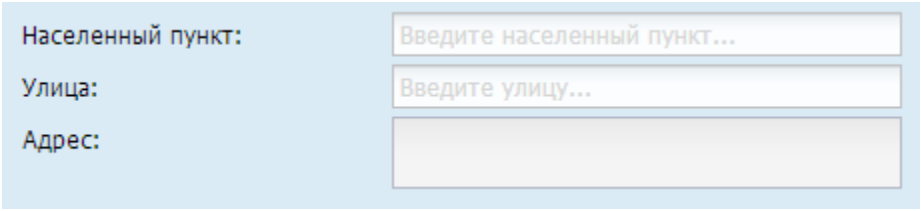


Рис. 3.1: `ExtAddrComponent` с атрибутом `level` равным `ExtAddrComponent.STREET`

С помощью атрибута `view_mode` устанавливается высота компонента. Перечень значений атрибута `view_mode`:

- ExtAddrComponent.VIEW_1 - все поля выстраиваются в одну строку + поле полного адреса
- ExtAddrComponent.VIEW_2 - все поля выстраиваются в две строки + поле полного адреса (по умолчанию). Данный параметр не используется при значении атрибута `level` == ExtAddrComponent.PLACE
- ExtAddrComponent.VIEW_3 - все поля выстраиваются в три строки + поле полного адреса. Данный параметр не используется при значенияз атрибута `level` == ExtAddrComponent.PLACE или ExtAddrComponent.STREET

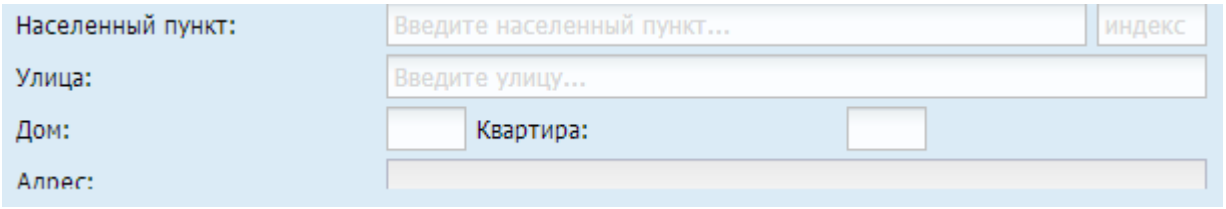


Рис. 3.2: `ExtAddrComponent` с атрибутом `view_mode` равным `ExtAddrComponent.VIEW_3`

Для сохранения адреса в моделях приложения необходимо определить в них поля соответствующие кодам полей компонента `ExtAddrComponent`.

```
class SomeModel():
    # ***********
    place = models.CharField(max_length=13, null=True, default ='')  # Населенный пункт
    street = models.CharField(max_length=17, null=True, default ='') # Улица
    house = models.CharField(max_length=10, null=True, default ='')  # Дом
    flat = models.CharField(max_length=5, null=True, default ='')  # Квартира
    zipcode = models.CharField(max_length=6, null=True, default ='') # Индекс
    addr =  models.CharField(max_length=200, null=True, default ='') # Адрес
```

При необходимости стандартные коды полей компонента и подписи полей можно переопределить, например:

```
fact_address = ExtAddrComponent()
fact_address.flat_field_name = 'fact_flat'
fact_address.addr_field_name = 'fact_addr'
fact_address.street_field_name = 'fact_street'
fact_address.place_field_name = 'fact_place'
fact_address.house_field_name = 'fact_house'
fact_address.zipcode_field_name = 'fact_zipcode'
fact_address.house_label = u'Дом/Корпус'
```

Глава 4

Дополнительно

- *genindex*
- *modindex*
- *search*