Перевод Django проекта

Для перевода в Django используется i18n Middleware

Для указания текста, который должен быть подвержен переводу, используются функции **gettext** и **gettext_lazy**, вторую функцию используется в основном при переводе статических строк

Согласно соглашению, функции **gettest/gettext_lazy** необходимо ипмортировать с псевдонимом «_» для уменьшения количества вводимых символов

Для

```python
from django.utils.translation import gettext as _


def index(request: HttpRequest):
    text = _('Hello My Friends!')
    return render(request, 'app/index.html', {'text': text})
```

указания сложной логики перевода, когда требуется единственное и множественное число, используются функции **ngettext/ngettext_lazy**

В качестве аргументов, данные функции принимают текст с единственным и множественным числом, а так же число, говорящее о количестве множественных форм. Следует учесть, что количество множественных форм может отличаться в каждом языке, поэтому следует использовать данные функции с особой осторожностью

```python
from django.utils.translation import gettext as _
from django.utils.translation import ngettext


def index(request: HttpRequest):
    text = _('Hello My Friends!')
    n = 5
    text2 = ngettext('number %(n)d', 'numbers %(n)d', n) % {'n': n}
    return render(request, 'app/index.html', {'text': text, 'text2': text2})
```

Единственным образом форматировать переводимый текст является форматирование с использованием либо «**%**» либо с использованием функции **format**

Контекстуальные маркеры

Используются, когда есть слова, которые могут использоваться в различных значения, в таком случае, используются маркеры, которые не влияют на результирующий текст, но помогают при переводе ориентироваться в том, какое значение данного слова подразумевалось в коде.

Для создания контекстуальных маркеров, используются функции **pgettext/pgettext_lazy**

Данные функции принимают строку для перевода и комментарий к нему

Переводы в шаблоне

Для реализации перевода текста или переменных в шаблоне необходимо подключить тек i18n **{% load i18n %}**

Текст, который подвержен переводу, помещается в тек **translate**

{% translate «some text» %}

Если данный текст используется многократно, его перевод можно поместить в переменную указав псевдоним

{% translate «some text» as some_text %}

Настройка

Для каждого приложения необходимо создать папку **locale** куда будут собираться файлы с переводами текста, а так же будет производится сам перевод

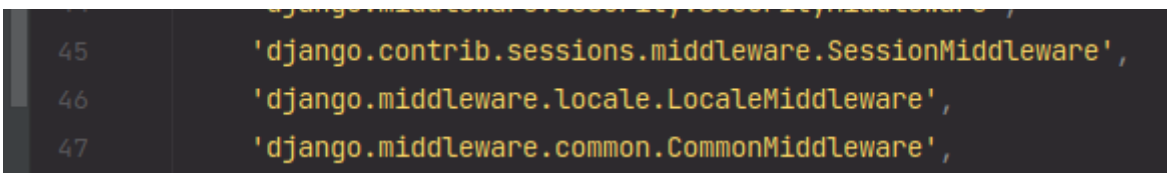
Так же возможен вариант указания переменной **LOCALE_PATH** которая является списком путей, до папок, содержащих переводы, в файле **settings.py**

Последним этапом настройки, является внесение

'django.middleware.locale.LocaleMiddleware',

В список **MIDDLEWARE** в файле **settings.py**

Подключение данного импорта обязательно должно следовать после подключение сессии, так как данный middleware использует для перевода данные из сессии



Сбор строк перевода

Для сбора файлов перевода, выполняется команда

django-admin makemessages -l ru

Данную команду необходимо вводить, находясь либо в корне проекта, либо в каком-то конкретном приложении. После ее выполнения, в папке **locale** будет создана папка с указаным языком и подпапка **LC_MESSAGES** внутри которой будет расположен файл с расширением **.po**

Для обновления файлов перевода, производится команда

django-admin makemessages -a

После перевода, необходимо их скомпилировать коммандой

django-admin compilemessages

После выполнения данной команды, рядом с файлом переводов, будет сгенерирован файл с расширением **.mo**, который по своей сути является sqlite базой данных

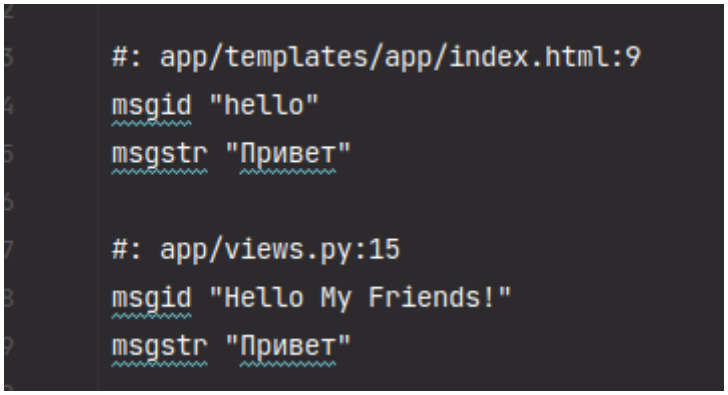
Осуществление перевода

Внутри файлов с расширением **.po** будут собраны все строки, которые были помечены какой-либо функцией отвечающей за перевод и для каждой такой строки будет сгенерировано 2 переменные

msgid — исходная строка, значение данной переменной ни в коем случае не следует изменять

msgstr — переменная, куда необходимо вписать перевод строки для данного языка

Следует отметить, что если было найдено 2 и более строки с одинаковым текстом, то они будут представлены в файле перевода как 1 строка



После выполнения всех переводов, компилируем их, и теперь текст на сайте будет отображаться исходя из языка пользователя