概要

Djangoフレームワークは多言語に対応するための機能も提供しており、割と簡単に扱うことができます。基本的な使い方と流れを説明していきます。

多言語対応の準備

多言語対応する上での準備です。

settings.py

まず、LANGUAGE_CODEをjaとしておきます。これはおなじみの設定ですね。

```
LANGUAGE_CODE = 'ja'
```

特に多言語対応をしない場合ならば、このLANGUAGE_CODEに指定した言語で翻訳されていきます。しかし多言語対応、サイト上で複数の言語を扱う場合は追加の設定が必要です。

MIDDLEWAREに、django.middleware.locale.LocaleMiddlewareを読み込ませる必要があります。

```
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.locale.LocaleMiddleware',  # これ
    'django.middleware.common.CommonMiddleware',
```

これにより、Djangoはこのサイト上で複数の言語を扱うのだなと判断します。アクセスしてきたURLやセッション、使っているブラウザの言語といった情報をもとに、サイト上の文章を翻訳しようとします。

どうしてもユーザーの言語がわからない場合にだけ、LANGUAGE_CODEが使われるようになります。言い換えると、**Localeミドルウェアを使わない場合は必ずLANGUAGE_CODEの言語で翻訳される**ということになります。

ミドルウェアの場所にも注意です。SessionMidlewareとCommonMiddlewareの間に書きます。キャッシュ関連ミドルウェアを使う場合は、CommonMiddlewareの前にLocaleMiddlewareが来るようにしてください(つまりキャッシュの後に読み込ませる)。

また、次の記述も追加しておきましょう。

```
LOCALE_PATHS = (
    os.path.join(BASE_DIR, 'locale'),
)
```

localeディレクトリの作成

プロジェクト直下と、アプリケーションディレクトリ内にlocaleというディレクトリを作成しておきましょう。

今回はappというアプリケーションです。次の画像のようになります。 2つのlocaleディレクトリがあることを確認しましょう。  プロジェクト直下とアプリケーション直下にlocaleディレクトリを置く

Windowsの場合

翻訳ファイルを作成するためのツールが必要です。こちらにアクセスし、ダウンロードしてください。

多言語化したい文章を指定する

それでは、多言語化したい文章をマークしていきましょう。

テンプレート内の文章

テンプレート内で文章を多言語化する簡単な方法は、テンプレートタグtransを使う方法です。

```
 {% load i18n %}
 <h1>{% trans "Hello" %}</h1>
```

{% load i18n %}とするのを忘れないようにしましょう。{% trans %}をはじめとする、翻訳関連のテンプレートタグ・フィルタを読み込むための記述です。これがどういう動作をするかは、後々わかるようになります。

Pythonファイル内の文章

フォームのlabelやモデルのverbose_nameなど、そういった部分も多言語化できますし、テンプレートへ渡す文字列を多言語化したいかもしれません。

例えば、モデルの例です。

```
 from django.db import models
 from django.utils.translation import gettext_lazy as _


 class Post(models.Model):
     title = models.CharField(_('post title'), max_length=255)
```

テンプレートではtransを使いましたが、Pythonファイル内では
django.utils.translation.gettext_lazy又はgettextをよく使います。lazyとついているのは遅延評

価用...reverse関数とreverse_lazy関数の関係と似たようなものです。上の例ならば、titleフィールドの表示名を実際に表示する場面で翻訳をしてくれます。

モデルやフォームのフィールド内にあるlabel, verbose_name,help_text...モジュールのimport時にすぐに読み込まれるような場所、クラス属性内に関してはlazyを使うほうが無難です。

as _ として、**gettext_lazy関数をアンダースコアで使える**ようにしていますがこれは慣習的なものです。

翻訳ファイルの作成

以下のコマンドを実行しましょう。これは日本語の翻訳ファイルを作成するコマンドです。

```
django-admin makemessages -l ja
```

localeディレクトリの中に、**ja**というディレクトリができたはずです。おそらく、2通りの状態になります。

アプリケーション直下のlocaleディレクトリだけ中身がある
アプリケーション直下と、プロジェクト直下のlocaleディレクトリ両方に中身がある

Djangoが翻訳ファイルをどこに配置するかというと、**まずアプリケーション内のlocaleディレクトリに配置しようとします**。app/models.pyに翻訳テキストがありましたが、これはappアプリケーション内のlocaleディレクトリに翻訳ファイルが置かれます。

なので、appアプリケーション内のlocaleディレクトリ内、django.poには次の記述が必ずあるはずです。

```
#: .¥app¥models.py:6
msgid "post title"
msgstr ""
```

特定のアプリケーションに属さないファイルはsettings.pyに定義したLOCALE_PATHSに翻訳ファイルが置かれます。特定のアプリケーションに属さないファイルというのは例えば、settings.pyやプロジェクト直下に置いたtemplates内の翻訳テキストが該当します。

なので、プロジェクト直下にtemplatesを置いている方であれば、プロジェクト直下のlocale内のdjango.poに次の記述があるはずです。

```
#: .¥templates¥app¥top.html:7
msgid "Hello"
msgstr ""
```

templatesをアプリケーション内に置いている方ならば、プロジェクト直下のlocaleは空で、アプリケーション内のlocaleにそれがあります。

```
#: .¥app¥templates¥app¥top.html:7
msgid "Hello"
msgstr ""
```

アプリケーション内にlocaleディレクトリがない場合は、LOCALE_PATHSに翻訳ファイルが置かれます。なので、プロジェクト全ての翻訳を一か所にまとめたい場合は、アプリケーション内にlocaleディレクトリを置かないほうが良いかもしれません。また、LOCALE_PATHSも空の場合はエラーになります。

翻訳する

---

django.poというファイルがあるので開きましょう。

```
#: .¥app¥models.py:6
msgid "post title"
msgstr ""

#: .¥app¥templates¥app¥top.html:5
msgid "Hello"
msgstr ""
```

msgidは、{% trans Hello %}のHello部分であったり、_('post title')のpost title部分です。ここに入れたい日本語の文章をmsgstrに書いていきます。

```
#: .¥app¥models.py:6
msgid "post title"
msgstr "記事タイトル"

#: .¥app¥templates¥app¥top.html:5
msgid "Hello"
msgstr "こんにちは"
```

翻訳する文章を作成したら、コンパイルします。

```
django-admin compilemessages
```

これにより、**django.mo**というファイルができます。これでひとまず終了です。

今後は、新しい言語を追加したり翻訳したい文章が増えればmakemessagesを行い、修正があればdjango.poを編集し、そしてcompilemessagesを行っていきます。

UnicodeEncodeErrorが出る場合

ロケールの問題が殆どです。CentOS7ならば、以下のようにロケールを変更できます。

```
localectl set-locale LANG=ja_JP.utf-8
source /etc/locale.conf
```

翻訳の上書き

翻訳ファイルの探索順序は、次のようになっています。

settings.pyのLOCALE_PATHSが指している場所
この記事で言うところの、プロジェクト直下のlocale
各アプリケーション内にあるlocaleディレクトリ(INSTALLED_APPSの上から順に)
authやadminなどの組み込みのDjangoアプリケーション内にもlocaleがあります。
django/conf/locale
Django全体で使われる基本翻訳。例えば曜日の翻訳文章などはここで。

もしかしたら、Django標準の翻訳テキストを上書きしたい場合もあるかもしれませんし、サードパーティ製ライブラリの翻訳を上書きしたい、といったケースもあるかもしれません。

Django管理サイトのタイトルは"Django administration"なのですが、これを上書きするならばLOCALE_PATHS内で次のように定義を上書きすればOKです。

```
msgid "Django administration"
msgstr "管理サイトへようこそ!"
```

もしくは、アプリケーション内のlocaleで上書きすることも可能です。各アプリケーション内のlocaleディレクトリですが、INSTALLED_APPSの上から順に探索され、見つかったら探索が終わります。各種テンプレートやstaticファイルの探索順序と同じです。INSTALLED_APPS**で自作アプリが上書きしたいアプリケーション(管理サイトタイトルならば、**admin**)より上にある状態**にしておくことが必要です。

言語切り替えプルダウン

各ユーザーが、表示したい言語をそれぞれ切り替えれるようにしてみます。

urls.pyに以下を追加し...

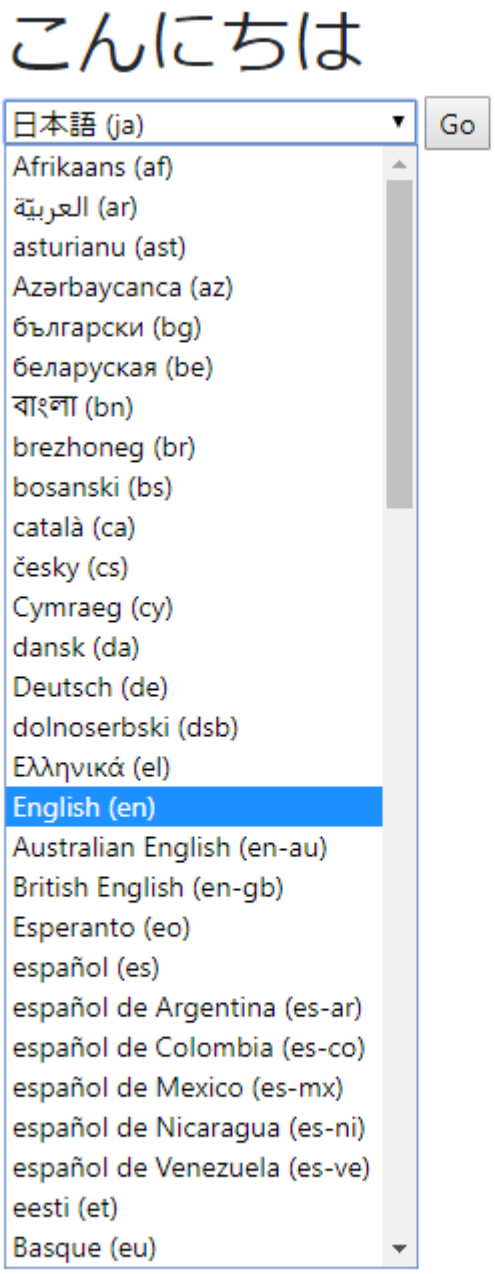
```
path('i18n/', include('django.conf.urls.i18n')),
```

テンプレート内に以下の記述をしておきます。

```
{% load i18n %}

<form action="{% url 'set_language' %}" method="post">{% csrf_token %}
    <input name="next" type="hidden" value="{{ redirect_to }}">
    <select name="language">
        {% get_current_language as LANGUAGE_CODE %}
        {% get_available_languages as LANGUAGES %}
        {% get_language_info_list for LANGUAGES as languages %}
        {% for language in languages %}
            <option value="{{ language.code }}"{% if language.code == LANGUAGE_CODE %} se
                {{ language.name_local }} ({{ language.code }})
            </option>
        {% endfor %}
    </select>
    <input type="submit" value="Go">
</form>
```

ヘッダーとかに置いておくと良さそうですね。



プルダウンの言語が多すぎる場合は、絞ることもできます。settings.pyに追記しましょう。

```
from django.utils.translation import ugettext_lazy as _
LANGUAGES = [
    ('en', _('English')),
    ('ja', _('Japanese')),
]
```

URLで言語を判断する

プロジェクトのurls.pyにて、以下のように定義します。

```
from django.conf.urls.i18n import i18n_patterns
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),

]

urlpatterns += i18n_patterns(
    path('', include('app.urls')),
)
```

いつもならば、path('', include('app.urls')),という記述はpath('admin/', admin.site.urls),の下あたりにありますね。しかし、今回は別の場所...i18n_patterns内に移しました。これにより、appの各URLに関しては、URLの頭に/ja/hogehoge とか/en/hogehoge とかでアクセスできるようになります