

# Django ミドルウェア

[Django]

- ・ <http://docs.djangoproject.jp/ja/latest/topics/http/middleware.html>
- ・ ミドルウェア (Middleware) とは、Django のリクエスト / レスポンス処理をフック するためのフレームワークです。ミドルウェアは軽量かつ低水準な「プラグイン」システムで、Django の入出力を操作します。

## ミドルウェアの有効化

- ・ ミドルウェアコンポーネントを有効化するには、Django 設定ファイルの MIDDLEWARE\_CLASSES リストにコンポーネントを追加します。

## セッションの使い方

- ・ <http://docs.djangoproject.jp/ja/latest/topics/http/sessions.html>

## セッションを有効にする

- ・ セッション機能はミドルウェア として実装されています。
- ・ セッション機能をオンオフするには MIDDLEWARE\_CLASSES 設定を編集します。
- ・ MIDDLEWARE\_CLASSES を編集して、  
'django.contrib.sessions.middleware.SessionMiddleware' を入れます。
- ・ INSTALLED\_APPS 設定に 'django.contrib.sessions' を入れ、manage.py syncdb を実行して、セッションデータを保存するためのデータベーステーブルを作成します。

## セッションエンジンの設定

- ・ デフォルトでは、Django はセッションをデータベースに  
(django.contrib.sessions.models.Session モデルを使って ) 保存します。

## ファイルベースのセッション

- ・ SESSION\_ENGINE 設定を "django.contrib.sessions.backends.file" にします。
- ・ 必要に応じて SESSION\_FILE\_PATH も設定してください ( デフォルト値は tempfile.gettempdir() の戻り値で、たいていは /tmp です )。

## キャッシュベースのセッション

- ・ SESSION\_ENGINE に "django.contrib.sessions.backends.cache" を設定してください。

## ビュー中でセッションを扱う

- ・ SessionMiddleware を有効にすると、各々の HttpRequest オブジェクト (Django ビュー関数の最初の引数 ) は辞書ライクオブジェクトの属性 session を持つようになります。この属性は読み書き可能です。

セッションオブジェクトは標準辞書オブジェクトのメソッドを実装しています

例

```
fav_color = request.session['fav_color']
request.session['fav_color'] = 'blue'
del request.session['fav_color']
```

key がない場合には KeyError を送出します。

```
fav_color' in request.session
fav_color = request.session.get('fav_color', 'red')
```

## ログインの例

```
def login(request):
    m = Member.objects.get(username__exact=request.POST['username'])
    if m.password == request.POST['password']:
        request.session['member_id'] = m.id
        return HttpResponse("You're logged in.")
    else:
        return HttpResponse("Your username and password didn't match.")
```

## ログアウトの例

```
def logout(request):
    try:
        del request.session['member_id']
    except KeyError:
        pass
    return HttpResponse("You're logged out.")
```