

CRUDを理解しよう

ここまでのレッスンは1つ目のチュートリアルの内容と重複するポイントが多かったので、もしかしたら退屈に感じた人もいるかもしれませんね。さあ、このレッスンからはいよいよ新しい概念を学習していきます！

このレッスンで学ぶのは、**CRUD**という言葉の意味です。CRUDとは、4つの単語の頭文字から成る言葉です。これらはソフトウェアで頻繁に利用される基本機能で、データを扱う上で非常に重要な概念となります。

Create



Read



Update



Delete



このチュートリアルで作成するメモアプリでは、新しいメモを作ったり（Create）、保存してあるメモのデータを読み込んで表示したり（Read）、メモの内容を編集・更新したり（Update）、不要になったメモを削除したり（Delete）します。みなさんがいつも使っているほとんどのWebサービスでもこの基本機能が使われているはずです。

これまでのレッスンでメモを画面に表示させることはもうできているので、Readの機能はもう作れていますよね!views.pyのmemo.objects.all()や、get_object_or_404(Memo, id=memo_id)という記述で必要なデータを読み込んでページに表示させています。

また、Adminページからではありますが、新規メモを作成（Create）する作業もしています。UpdateやDelete機能はこのあとのレッスンで学習することとして、まずこのレッスンでは、このCreate機能を自分で実装できるように学習していきましょう。

createメソッドを使う

メモを新しく作成（Create）するには、createメソッドを使うことで実現できます。1つ目のチュートリアルでも紹介しましたが、shell画面で実際に使っておさらいしてみましょう！

~/memo/

```
$ python manage.py shell
>>> from app.models import Memo
>>> Memo.objects.all()
<QuerySet [<Memo: 買うもののリスト>, <Memo: 明日やること>, <Memo: 出張するときの持ち物>]
>>> Memo.objects.create(title='test', text='test')
<Memo: test>
>>> Memo.objects.all()
<QuerySet [<Memo: 買うもののリスト>, <Memo: 明日やること>, <Memo: 出張するときの持ち物>, <Memo: test>]
```

うまくできましたか？これでAdminページとshell画面からは新しいメモを作成できるようになりました。ただ、これだけだと管理者ではない一般ユーザーはメモを作成できませんよね。ブラウザ上からでもメモを作成できるようにしていきましょう。

HTTPメソッドについて

ブラウザからも新しいメモを投稿できるように、HTTPメソッドというものを理解しましょう。

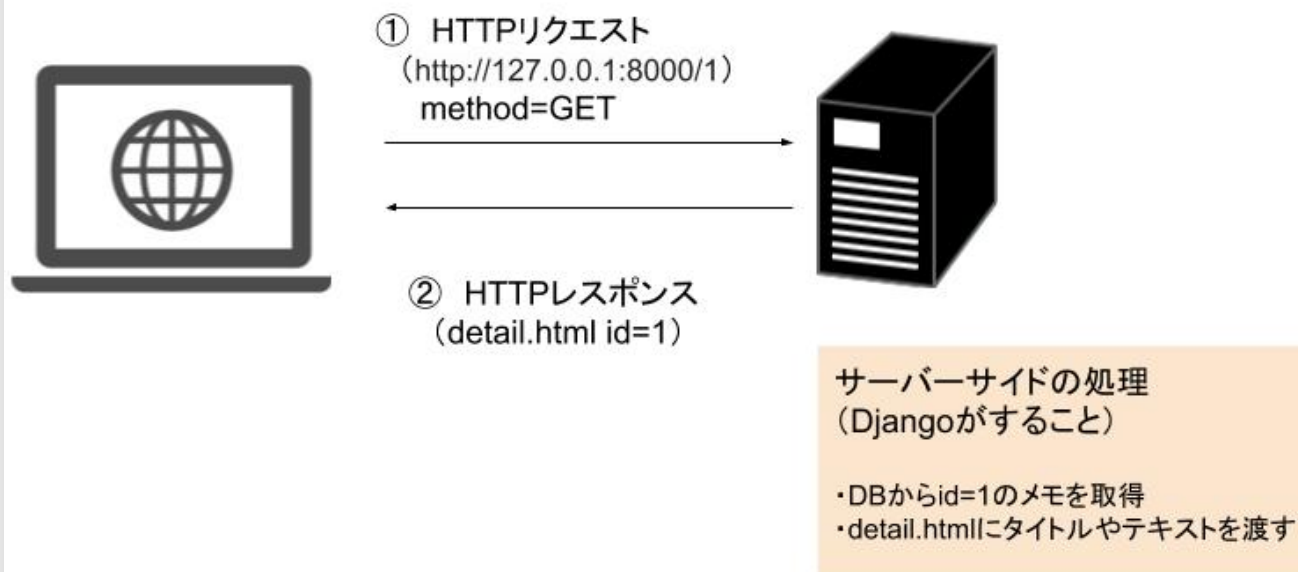
ブラウザとサーバーの通信では、HTTPというプロトコルが使われています。プロトコルとは規約といった意味があります。つまり**HTTPという決まった方式にのっとりサーバーとやり取りをしている**ということです。

ユーザーはブラウザ上で特定のURLにアクセスすることによって、サーバーに対してHTTPリクエストを送っています。リクエストを受け取ったサーバーは、その内容に基づいて情報を返します（レスポンス）。

例えば、ユーザーがhttp://127.0.0.1:8000/1にアクセスした場合、サーバーに対して「detail.htmlを返してくれ」というリクエストが送られます。リクエストを受け取ったサーバーは、URLからid=1のメモを返せばいいんだと判断します。そしてDBからid=1のメモを探しだし、その情報をdetail.htmlに載せてブラウザに返します。このサーバー上の処理を定義しているのがDjangoというわけです。

この例の場合、情報を読み取っているなのでCRUDの中のReadに該当する機能となります。

Read(読み取り)



HTTPリクエストの中には様々な情報が含まれていますが、その中にはmethodという項目があります。methodは、そのリクエストがどんな種類のものであるかを表します。

上の図にある**GETメソッド**は、**サーバーからデータを返して欲しいとき**に使われます。ブラウザにURLを直接入力したときや、aタグで囲まれたリンクをクリックしたときなどは、サーバーに対してmethod=GETのリクエストが送られます。

一方、**サーバーに対して何かしらのデータを送りたいときはPOSTというmethod**を使います。POSTメソッドはデータを送ることによってDBに新しい情報を追加することができます。

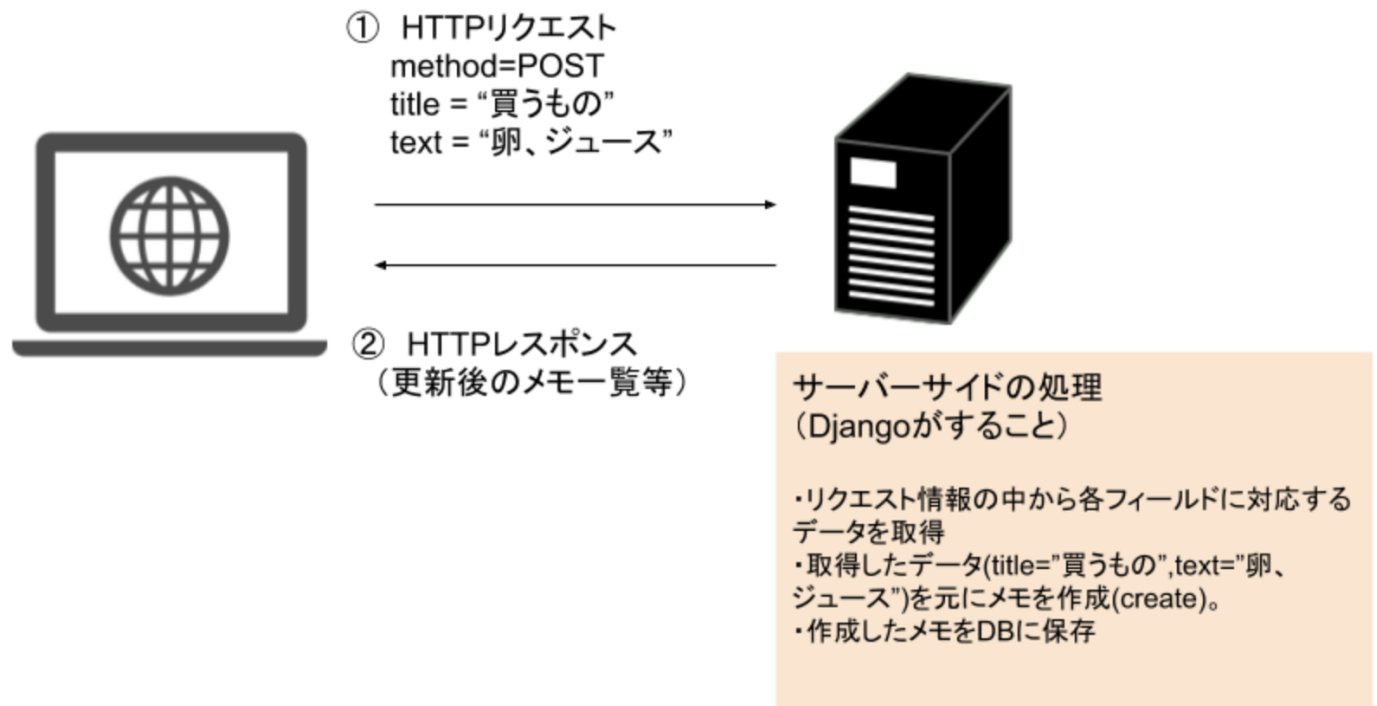
HTTPリクエストには、GETとPOST以外にも種類がありますが、ほとんどの場合このどちらかを利用することになります。

GET: 情報の読み取り、取得

POST: 情報の送信、投稿

新しくメモを作成するためには、情報をサーバーに送信する必要がありますのでPOSTメソッドを使います。イメージとして下の画像のようになります。

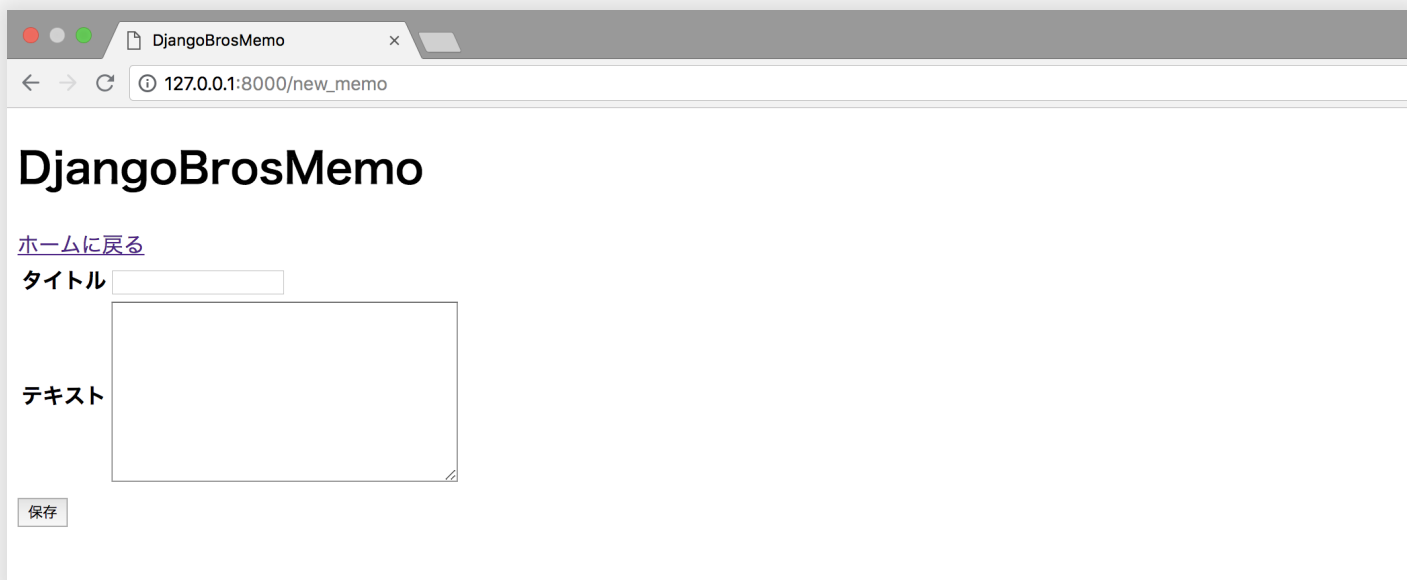
Create(生成)



入力フォームから、メモのタイトルや内容となるテキストを受け取り、それをサーバー側に送信します。サーバー側では、受け取った情報をDBに追加・保存します。この一連の流れはCRUDの中のCreateに該当します。

入力フォームのイメージ図は以下のようになりますが、この「保存」ボタンを押した時に、method=POSTのリクエストが送られるように設定します。フォームの作り方と、POSTメソ

ツドの送り方は次のレッスンで学習しましょう！



コンソールでログを確認する

Djangoでは、`python manage.py runserver` でサーバーを起動したコンソール画面から、ログを見ることができます。下の画像は、`http://127.0.0.1:8000/` にアクセスした時と、「新規メモ作成画面」にアクセスしたときの様子です。どちらもGETが表示されることが確認できますね。200という数字はステータスコードと呼ばれるもので、リクエストの受付に成功したことを表しています。

```
Django version 2.0.2, using settings 'memo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[07/Sep/2018 08:30:36] "GET / HTTP/1.1" 200 625
[07/Sep/2018 08:30:52] "GET /new_memo HTTP/1.1" 200 49
Not Found: /100
[07/Sep/2018 08:31:06] "GET /100 HTTP/1.1" 404 1711
```

前のレッスンのように、id=100のメモが存在しない状態で`http://127.0.0.1:8000/100`にアクセスすると、ログにも404が表示されることが確認できるでしょう。