Apr 29

How to configure our own template file:

By using template_name variable we have to specify our own template file.

How to configure our own context object:

We have to use: context_object_name variable.

- **views.py**

```
class BookListView(ListView):
    model = Book
    template_name = 'testapp/books.html'
    context_object_name = 'books'
```

- **books.html**

```
<body>
    <div class="container">
      <h1>All Books Information from customized template file</h1>
      {% for book in books %}
      <ul>
        <li>Title:<strong>{{book.title}}</strong></li>
        <li>Author:<strong>{{book.author}}</strong></li>
        <li>Pages:<strong>{{book.pages}}</strong></li>
        <li>Price:<strong>{{book.price}}</strong></li>
      </ul>
      <hr>
      {% endfor %}
    </div>
</body>
```

DetailView:

To get the details of a particular record.
Default template_file = book_detail.html
Default context_object_name = book or object

- **views.py**

```
class BookListView2(ListView):
    model = Book
    template_name = 'testapp/books.html'
    context_object_name = 'books'
```

- **books.html**

```
<body>
  <div class="container">
   <h1>All Books Information</h1>
   {% for book in books %}
   <li><a href="/{{book.id}}">{{book.title}}</a></li>
   {% endfor %}
  </div>
</body>
```

- **urls.py**

```
path('list2/', views.BookListView2.as_view()),
```

- **views.py**

```
class BookDetailView(DetailView):
  model = Book
```

- **urls.py**

```
path('<int:pk>/', views.BookDetailView.as_view()),
```

- **book_detail.html**

```
<body>
  <div class="container">
   <h1>{{book.title}} Information</h1>
   <ul>
    <li>Title:<strong>{{book.title}}</strong></li>
    <li>Author:<strong>{{book.author}}</strong></li>
    <li>Pages:<strong>{{book.pages}}</strong></li>
    <li>Price:<strong>{{book.price}}</strong></li>
   </ul>
  </div>
</body>
```

CreateView:

To insert data into our database table(model).

- **views.py**

```
class BookCreateView(CreateView):
  model = Book
```

- **urls.py**

path('/', views.BookCreateView.as_view()),

If we send request:

http://127.0.0.1:8000/create/

ImproperlyConfigured at /create/
Using ModelFormMixin (base class of BookCreateView) without the 'fields' attribute is prohibited.

- **views.py**

fields = ('title','author','pages','price')

Now send the request:

http://127.0.0.1:8000/create/

TemplateDoesNotExist at /create/
testapp/book_form.html

Note:

The default template is display form for create operation is:book_form.html

We have to create this template file

- **book_form.html**

```
<body>
  <div class="container" align='center'>
    <h1>Book Insert/Create Form</h1>
    <form method="post">
      {{form.as_p}}
      {% csrf_token %}
      <input type="submit" name="" class="btn btn-success" value="Insert New Book">
    </form>
  </div>
</body>
```

<u>If we fill the form and submit</u>

-->The record will be inserted into database, but we will get an error.

-->After inserting to which page, control has to go, we did not define anywhere.
This is the reason for error.

ImproperlyConfigured at /create/
No URL to redirect to. Either provide a url or define a get_absolute_url method on the Model.

- **models.py**

```
from django.urls import reverse
class Book(models.Model):
        ----
        def get_absolute_url(self):
    return reverse('detail',kwargs={'pk':self.pk})
```

- **urls.py**

```
path('<int:pk>/', views.BookDetailView.as_view(),name='detail'),
```

UpdateView:
To update an existing record.

- **views.py**

```
class BookUpdateView(UpdateView):
  model = Book
  fields = ('pages','price')
```

- **urls.py**

```
path('update/<int:pk>', views.BookUpdateView.as_view()),
```

The default template is: book_form.html

Add update button in book_details.html page

```
<a class="btn btn-warning" href="/update/{{book.id}}">Update This Book Information</a>
```

DeleteView:

- **views.py**

```
from django.urls import reverse_lazy
class BookDeleteView(DeleteView):
  model = Book
```

```
success_url = reverse_lazy('listbooks')
```

success_url represents the target page which should be displayed after delete operation.
reverse_lazy() function will wait until deleteing the record.

- **urls.py**

```
path('delete/<int:pk>', views.BookDeleteView.as_view()),
```

- **book_confirm_delete.html**

```
<body>
  <div class="container" align='center'>
    <h1>Do you want to really delete book:{{book.title}}???</h1>
    <form method="post">
      {{form.as_p}}
      {% csrf_token %}
      <input type="submit" name="" class="btn btn-danger" value="Delete Book">
      <a class="btn btn-success" href="/list2">Cancel(Don't Delete)</a>
    </form>
  </div>
</body>
```

```
success_url = reverse_lazy('listbooks')
path('list2/', views.BookListView2.as_view(),name='listbooks')
```

Add delete button in book_detail.html

```
<a class="btn btn-danger" href="/delete/{{book.id}}">Delete This Book Information</a>
```

Add Insert button in books.html

```
<a class="btn btn-primary" href="/create">Insert New Book</a>
```