**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>

**If we fill the form and submit**

**-->**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>