Generic Classed Based View

ListView

class ArticleListView(ListView):

template\_name = 'blog2/article\_list.html'

queryset = Article.objects.all() # model = Article

# `object\_list` is the default context passed in template.

DetailView - `pk` passed in url

class ArticleDetailView(DetailView):

template\_name = 'blog2/article\_detail.html'

queryset = Article.objects.all() # model = Article

# `object` is the default context passed in template.

DetailView -`id` passed in url

class ArticleDetailView(DetailView):

template\_name = 'blog2/article\_detail.html'

queryset = Article.objects.all()

# overriding the primary function to replace `*pk*` with `*id*`

def get\_object(self):

# you access the URL args with `self.args` and `self.kwargs` in views

id\_ = self.kwargs.get('id')

return get\_object\_or\_404(Article, id=id\_)

CreateView

class ArticleCreateView(CreateView):

template\_name = 'blog2/article\_create.html'

form\_class = ArticleModelForm

queryset = Article.objects.all()

def form\_valid(self, form):

return super().form\_valid(form)

UpdateView

urls.py

path('<int:id>/update/', ArticleUpdateView.as\_view(),

name='article-update'),

views.py

class ArticleUpdateView(UpdateView):

template\_name = 'blog2/article\_create.html'

form\_class = ArticleModelForm

queryset = Article.objects.all()

def get\_object(self):

my\_id = self.kwargs.get('id')

return get\_object\_or\_404(Article, id=my\_id)

def form\_valid(self, form):

return super().form\_valid(form)

DeleteView - `id` passed in url

urls.py

path('<int:id>/delete/', ArticleDeleteView.as\_view(),

name='article-delete'),

views.py

class ArticleDeleteView(DeleteView):

template\_name = 'blog2/article\_delete.html'

def get\_success\_url(self):

return reverse('articles:article-list')

def get\_object(self):

my\_id = self.kwargs.get('id')

return get\_object\_or\_404(Article, id=my\_id)