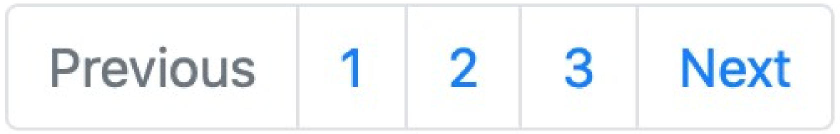
Pagination

Using pagination we can split data to several pages, with Previous/Next links.



Django provides a few classes that help you manage paginated data:-

- Paginator Class
- Page Class

Paginator Class

class Paginator(object_list, per_page, orphans=0, allow_empty_first_page=True)

Where,

object_list – It takes tuple, list, QuerySet or other sliceable object with a count() or __len__() method. It is required.

per_page – The maximum number of items to include on a page, not including orphans. It is required.

orphans – Use this when you don't want to have a last page with very few items. If the last page would normally have a number of items less than or equal to orphans, then those items will be added to the previous page (which becomes the last page) instead of leaving the items on a page by themselves. orphans defaults to zero, which means pages are never combined and the last page may have one item. It is optional.

allow_empty_first_page - Whether or not the first page is allowed to be empty. If False and object_list is empty, then an EmptyPage error will be raised. It is optional.

Pagination Class Attributes

- count - The total number of objects, across all pages.
- num_pages - The total number of pages.
- page_range - A 1-based range iterator of page numbers, e.g. yielding [1, 2, 3, 4].

Pagination Class Methods

- get_page(number) – This method returns a Page object with the given 1-based index, while also handling out of range and invalid page numbers.
  If the page isn’t a number, it returns the first page.
  If the page number is negative or greater than the number of pages, it returns the last page.
  Raises an EmptyPage exception only if you specify Paginator(..., allow_empty_first_page=False) and the object_list is empty.

- page(number) – This method returns a Page object with the given 1-based index.
  Raises InvalidPage if the given page number doesn’t exist.

Page Class

class Page(object_list, number, paginator)

A page acts like a sequence of Page.object_list when using len() or iterating it directly.

Page Class Attributes

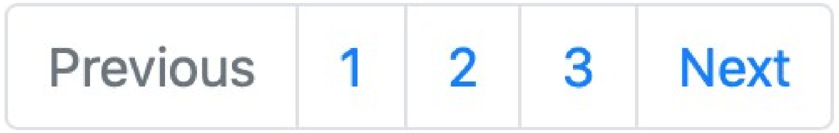
- object_list - The list of objects on this page.
- number - The 1-based page number for this page.
- paginator - The associated Paginator object.

Page Class Methods

- has_next() - It returns True if there’s a next page.
- has_previous() - It returns True if there’s a previous page.
- has_other_pages() - It returns True if there’s a next or previous page.
- next_page_number() - It returns the next page number. Raises InvalidPage if next page doesn’t exist.
- previous_page_number() - It returns the previous page number. Raises InvalidPage if previous page doesn’t exist.
- start_index() - It returns the 1-based index of the first object on the page, relative to all of the objects in the paginator’s list. For example, when paginating a list of 5 objects with 2 objects per page, the second page’s start_index() would return 3.
- end_index() - It returns the 1-based index of the last object on the page, relative to all of the objects in the paginator’s list. For example, when paginating a list of 5 objects with 2 objects per page, the second page’s end_index() would return 4.

Using Pagination

- Pagination with Function Based View
- Pagination with Class Based View


Previous
1
2
3
Next