(/)

Curriculum

**Short Specializations**

Average: **97.3%**

# 0x00. Pagination

**Back-end**

By: Emmanuel Turlay, Staff Software Engineer at Cruise

Weight: 1

Project over - took place from Dec 21, 2023 6:00 AM to Jan 2, 2024 6:00 AM

☑ An auto review will be launched at the deadline

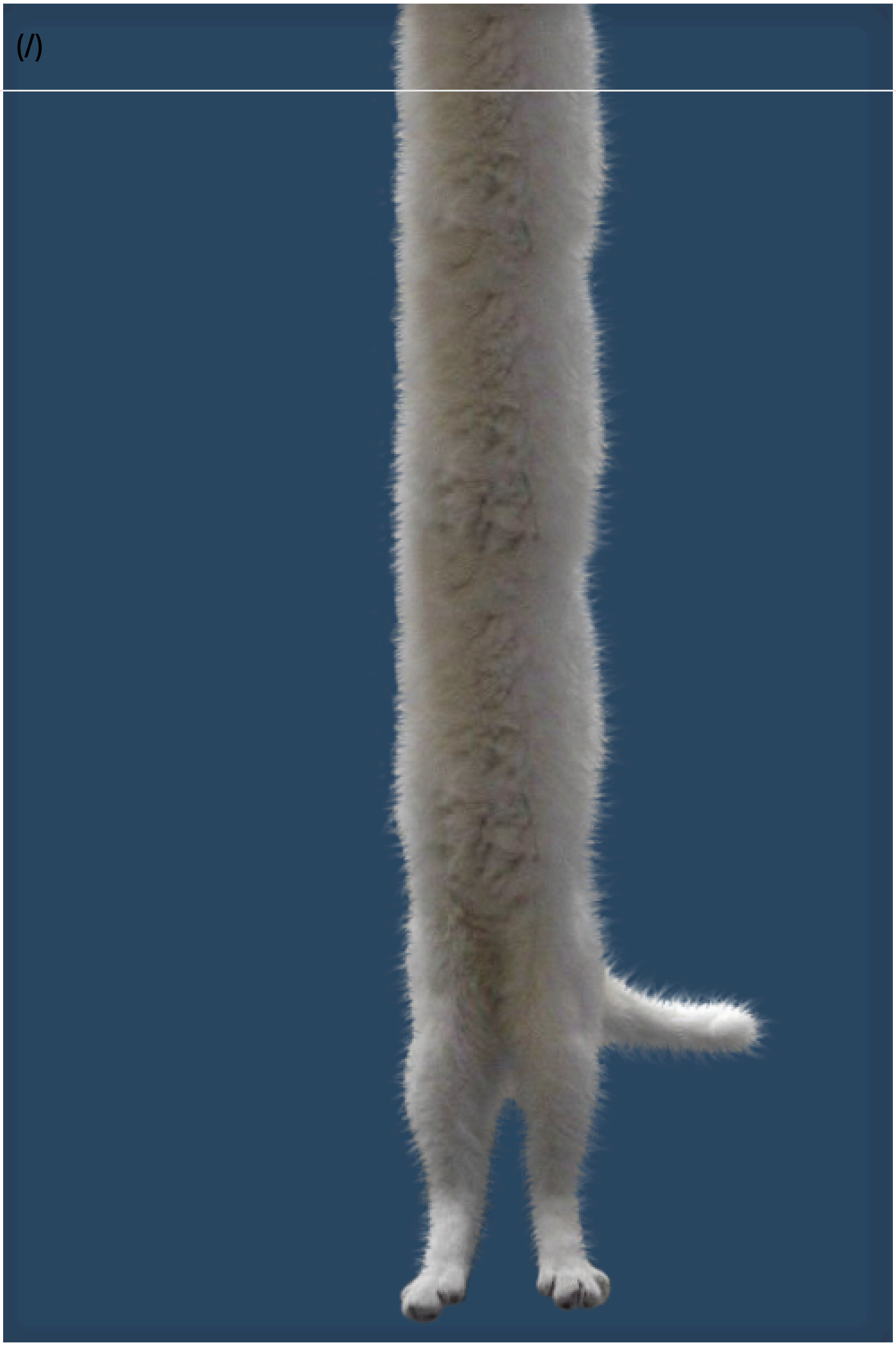
## In a nutshell…

- **Auto QA review:** 15.0/15 mandatory
- **Altogether: 100.0%**
    - Mandatory: 100.0%
    - Optional: no optional tasks

(/)

(/)

(/)



# Resources

**Read or watch:**

- REST API Design: Pagination (/rltoken/7Kdzi9CH1LdSfNQ4RaJUQw)
- HATEOAS (/rltoken/tfzcEbTSdMYSYxsspJH_oA)

# Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/zQ78qQVUjaPExupXQpAaHw), **without the help of Google**:

- How to paginate a dataset with simple page and page_size parameters
- How to paginate a dataset with hypermedia metadata
- How to paginate in a deletion-resilient manner

# Requirements

- All your files will be interpreted/compiled on Ubuntu 18.04 LTS using `python3` (version 3.7)
- All your files should end with a new line
- The first line of all your files should be exactly `#!/usr/bin/env python3`
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `pycodestyle` style (version 2.5.*)
- The length of your files will be tested using `wc`
- All your modules should have a documentation ( `python3 -c 'print(__import__("my_module").__doc__)'` )
- All your functions should have a documentation ( `python3 -c 'print(__import__("my_module").my_function.__doc__)'`
- A documentation is not a simple word, it’s a real sentence explaining what’s the purpose of the module, class or method (the length of it will be verified)
- All your functions and coroutines must be type-annotated.

# Setup: `Popular_Baby_Names.csv`

use this data file (/rltoken/NBLY6mdKDBR9zWvNADwjjg) for your project

# Tasks

### 0. Simple helper function

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a function named `index_range` that takes two integer arguments `page` and `page_size` .

The function should return a tuple of size two containing a start index and an end index corresponding to the range of indexes to return in a list for those particular pagination parameters.

Page numbers are 1-indexed, i.e. the first page is page 1.

```
bob@dylan:~$ cat 0-main.py
#!/usr/bin/env python3
"""
Main file
"""

index_range = __import__('0-simple_helper_function').index_range

res = index_range(1, 7)
print(type(res))
print(res)

res = index_range(page=3, page_size=15)
print(type(res))
print(res)

bob@dylan:~$ ./0-main.py
<class 'tuple'>
(0, 7)
<class 'tuple'>
(30, 45)
bob@dylan:~$
```

**Repo:**

- GitHub repository: `alx-backend`
- Directory: `0x00-pagination`
- File: `0-simple_helper_function.py`

☑ Done! Help Check your code Get a sandbox QA Review

## 1. Simple pagination

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Copy `index_range` from the previous task and the following class into your code

```
import csv
import math
from typing import List


class Server:
    """Server class to paginate a database of popular baby names.
    """
    DATA_FILE = "Popular_Baby_Names.csv"

    def __init__(self):
        self.__dataset = None

    def dataset(self) -> List[List]:
        """Cached dataset
        """
        if self.__dataset is None:
            with open(self.DATA_FILE) as f:
                reader = csv.reader(f)
                dataset = [row for row in reader]
            self.__dataset = dataset[1:]

        return self.__dataset

    def get_page(self, page: int = 1, page_size: int = 10) -> List[List]:
            pass
```

Implement a method named `get_page` that takes two integer arguments `page` with default value 1 and `page_size` with default value 10.

- You have to use this CSV file (/rltoken/NBLY6mdKDBR9zWvNADwjjg) (same as the one presented at the top of the project)
- Use `assert` to verify that both arguments are integers greater than 0.
- Use `index_range` to find the correct indexes to paginate the dataset correctly and return the appropriate page of the dataset (i.e. the correct list of rows).
- If the input arguments are out of range for the dataset, an empty list should be returned.

```
bob@dylan:~$  wc -l Popular_Baby_Names.csv
19419 Popular_Baby_Names.csv
bob@dylan:~$
bob@dylan:~$ head Popular_Baby_Names.csv
Year of Birth,Gender,Ethnicity,Child's First Name,Count,Rank
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Olivia,172,1
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Chloe,112,2
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Sophia,104,3
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Emma,99,4
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Emily,99,4
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Mia,79,5
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Charlotte,59,6
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Sarah,57,7
2016,FEMALE,ASIAN AND PACIFIC ISLANDER,Isabella,56,8
bob@dylan:~$
bob@dylan:~$  cat 1-main.py
#!/usr/bin/env python3
"""
Main file
"""

Server = __import__('1-simple_pagination').Server

server = Server()

try:
    should_err = server.get_page(-10, 2)
except AssertionError:
    print("AssertionError raised with negative values")

try:
    should_err = server.get_page(0, 0)
except AssertionError:
    print("AssertionError raised with 0")

try:
    should_err = server.get_page(2, 'Bob')
except AssertionError:
    print("AssertionError raised when page and/or page_size are not ints")


print(server.get_page(1, 3))
print(server.get_page(3, 2))
print(server.get_page(3000, 100))

bob@dylan:~$
bob@dylan:~$ ./1-main.py
AssertionError raised with negative values
AssertionError raised with 0
AssertionError raised when page and/or page_size are not ints
[['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Olivia', '172', '1'], ['2016', 'F
EMALE', 'ASIAN AND PACIFIC ISLANDER', 'Chloe', '112', '2'], ['2016', 'FEMALE', 'ASIA
N AND PACIFIC ISLANDER', 'Sophia', '104', '3']]
[['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Emily', '99', '4'], ['2016', 'FEM
ALE', 'ASIAN AND PACIFIC ISLANDER', 'Mia', '79', '5']]
[]
bob@dylan:~$
```

**Repo:**

- GitHub repository: `alx-backend`
- Directory: `0x00-pagination`
- File: `1-simple_pagination.py`

☑ Done! Help Check your code >_ Get a sandbox QA Review

## 2. Hypermedia pagination

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Replicate code from the previous task.

Implement a `get_hyper` method that takes the same arguments (and defaults) as `get_page` and returns a dictionary containing the following key-value pairs:

- `page_size` : the length of the returned dataset page
- `page` : the current page number
- `data` : the dataset page (equivalent to return from previous task)
- `next_page` : number of the next page, `None` if no next page
- `prev_page` : number of the previous page, `None` if no previous page
- `total_pages` : the total number of pages in the dataset as an integer

Make sure to reuse `get_page` in your implementation.

You can use the `math` module if necessary.

```
bob@dylan:~$ cat 2-main.py
#!/usr/bin/env python3
"""
Main file
"""

Server = __import__('2-hypermedia_pagination').Server

server = Server()

print(server.get_hyper(1, 2))
print("---")
print(server.get_hyper(2, 2))
print("---")
print(server.get_hyper(100, 3))
print("---")
print(server.get_hyper(3000, 100))

bob@dylan:~$
bob@dylan:~$ ./2-main.py
{'page_size': 2, 'page': 1, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDE
R', 'Olivia', '172', '1'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Chloe',
'112', '2']], 'next_page': 2, 'prev_page': None, 'total_pages': 9709}
---
{'page_size': 2, 'page': 2, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDE
R', 'Sophia', '104', '3'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Emma',
'99', '4']], 'next_page': 3, 'prev_page': 1, 'total_pages': 9709}
---
{'page_size': 3, 'page': 100, 'data': [['2016', 'FEMALE', 'BLACK NON HISPANIC', 'Lon
dyn', '14', '39'], ['2016', 'FEMALE', 'BLACK NON HISPANIC', 'Amirah', '14', '39'],
['2016', 'FEMALE', 'BLACK NON HISPANIC', 'McKenzie', '14', '39']], 'next_page': 101,
'prev_page': 99, 'total_pages': 6473}
---
{'page_size': 0, 'page': 3000, 'data': [], 'next_page': None, 'prev_page': 2999, 'to
tal_pages': 195}
bob@dylan:~$
```

**Repo:**

- GitHub repository: `alx-backend`
- Directory: `0x00-pagination`
- File: `2-hypermedia_pagination.py`

☑ Done! Help Check your code Get a sandbox QA Review

## 3. Deletion-resilient hypermedia pagination

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Score: 100.0% (Checks completed: 100.0%)

(/)

The goal here is that if between two queries, certain rows are removed from the dataset, the user does not miss items from dataset when changing page.

Start `3-hypermedia_del_pagination.py` with this code:

```
#!/usr/bin/env python3
"""
Deletion-resilient hypermedia pagination
"""

import csv
import math
from typing import List


class Server:
    """Server class to paginate a database of popular baby names.
    """
    DATA_FILE = "Popular_Baby_Names.csv"

    def __init__(self):
        self.__dataset = None
        self.__indexed_dataset = None

    def dataset(self) -> List[List]:
        """Cached dataset
        """
        if self.__dataset is None:
            with open(self.DATA_FILE) as f:
                reader = csv.reader(f)
                dataset = [row for row in reader]
            self.__dataset = dataset[1:]

        return self.__dataset

    def indexed_dataset(self) -> Dict[int, List]:
        """Dataset indexed by sorting position, starting at 0
        """
        if self.__indexed_dataset is None:
            dataset = self.dataset()
            truncated_dataset = dataset[:1000]
            self.__indexed_dataset = {
                i: dataset[i] for i in range(len(dataset))
            }
        return self.__indexed_dataset

    def get_hyper_index(self, index: int = None, page_size: int = 10) -> Dict:
            pass
```

(/)

Implement a `get_hyper_index` method with two integer arguments: `index` with a `None` default value and `page_size` with default value of 10.

- The method should return a dictionary with the following key-value pairs:
    - `index` : the current start index of the return page. That is the index of the first item in the current page. For example if requesting page 3 with `page_size` 20, and no data was removed from the dataset, the current index should be 60.
    - `next_index` : the next index to query with. That should be the index of the first item after the last item on the current page.
    - `page_size` : the current page size
    - `data` : the actual page of the dataset

**Requirements/Behavior**:

- Use `assert` to verify that `index` is in a valid range.
- If the user queries index 0, `page_size` 10, they will get rows indexed 0 to 9 included.
- If they request the next index (10) with `page_size` 10, but rows 3, 6 and 7 were deleted, the user should still receive rows indexed 10 to 19 included.

```
bob@dylan:~$ cat 3-main.py
#!/usr/bin/env python3
"""
Main file
"""

Server = __import__('3-hypermedia_del_pagination').Server

server = Server()

server.indexed_dataset()

try:
    server.get_hyper_index(300000, 100)
except AssertionError:
    print("AssertionError raised when out of range")


index = 3
page_size = 2

print("Nb items: {}".format(len(server._Server__indexed_dataset)))

# 1- request first index
res = server.get_hyper_index(index, page_size)
print(res)

# 2- request next index
print(server.get_hyper_index(res.get('next_index'), page_size))

# 3- remove the first index
del server._Server__indexed_dataset[res.get('index')]
print("Nb items: {}".format(len(server._Server__indexed_dataset)))

# 4- request again the initial index -> the first data retreives is not the same as the first request
print(server.get_hyper_index(index, page_size))

# 5- request again initial next index -> same data page as the request 2-
print(server.get_hyper_index(res.get('next_index'), page_size))

bob@dylan:~$
bob@dylan:~$ ./3-main.py
AssertionError raised when out of range
Nb items: 19418
{'index': 3, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Emma', '99', '4'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Emily', '99', '4']], 'page_size': 2, 'next_index': 5}
{'index': 5, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Mia', '79', '5'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Charlotte', '59', '6']], 'page_size': 2, 'next_index': 7}
Nb items: 19417
```

```
{'index': 3, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Emily', '9
9', '4'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Mia', '79', '5']], 'page
_size': 2, 'next_index': 6}
{'index': 5, 'data': [['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Mia', '79',
'5'], ['2016', 'FEMALE', 'ASIAN AND PACIFIC ISLANDER', 'Charlotte', '59', '6']], 'pa
ge_size': 2, 'next_index': 7}
bob@dylan:~$
```

**Repo:**

- GitHub repository: `alx-backend`
- Directory: `0x00-pagination`
- File: `3-hypermedia_del_pagination.py`

☑ Done! Help Check your code Get a sandbox QA Review