1. Design a simple Java Api which takes input list of dates in yyyy-MM-dd format and return below response:

Suppose current date is **20-July-2020**

|  |  |  |
| --- | --- | --- |
| Input | Output | Remarks |
| 20-July-2020 | {Today,20-July-2020} | NA |
| 21-July-2020 | {Tomorrow,21-July-2020} | NA |
| 22-July-2020 | {Wednesday, 22-July-2020} | Needs to print day of week |
| Any date including 23-25 July | {This week, 20 Jul-25Jul} | Needs to print week range with Mon-Sat to be considered as week start and end day |
| Any date 27 Jul-31 Jul | {Next Week, 27-31 Jul) | Needs to print week range with Mon-Sat to be considered as week start and end day |
| Any day between 01-Aug to 31-Aug-2020 | {Next Month, August} | Needs to print only month name |
| Any day between 01-Sep to 31-12-2020 | {This Year, 2020} | Need to print year |
| 19-July-2020 | {Yesterday, 19-July-2020} | NA |
| 18-July-2020 | {Saturday, 18-July-2020} | Needs to print day of week |
| Any date between 13 Jul-17 Jul | {Last Week, 13 Jul-18 Jul} | Need to print last week range |
| Any date between 01-July to 16 July | {This month, July} | Need to print current month |
| Any date in June | {Last month, June} | Need to print previous month name |
| Any date from 01-Jan-2020 to 31-May | {This year, 2020} | Need to print current year |
| Any date outside range of current year | Not A Valid choice | NA |

1. Design a api that can be used to refresh inmemory cache of any table level entity and another api to refresh its value as and when required.

Ex: Suppose there are 5 configurational tables in database, and we need to cache the values of complete table so as to use the same values from cache. Also write another api which can refresh value in db as well as cache when values are changed, so as to avoid server startup.

1. Below is the relation between tables:

Promotion Headers: Header1 , Header2 and so on

Promotion Banner: Banner1, Banner2 and so on

Promotion Banner Type: Large, Small, Xtra Large

Promotion Type: Fashion, Sports, Medicines etc

Promotion Banner Image : Image Details

Promotion Page: Homepage, Pdp,Clp etc

1 Promotion Header can be associated wth any number of pages and can be associated with any number of types.

A single promotion can have any number of banners and one banner can have only one image and has only one type.

Design a database and apis for storing retrieval and deletion of above promotion. Your api must support pagination and should not take more than 1sec.

1. Design some base classes which can be used across system as a common response structure and supports some status:

{

"status": "OK",

"code": 2000,

"message": "Request Successful",

"data": {

"pojo": {

}

}

}

This needs to be the structure of base response, status represent String representation of **HttpStatus,**  code is **user-define code,** message is **any custom message for any use case** and data is json representation o pojo.

Sample code is shared ::

|  |  |
| --- | --- |
| **Response Type** | **Code** |
| SUCCESS\_CODE | 2000 |
| PARAMS\_MISSING\_CODE | 4000 |
| TOKEN\_EXPIRE\_CODE | 4001 |
| TOKEN\_REQUIRED | 4002 |
| TOKEN\_INVALID | 4003 |
| STORE\_ALREADY\_EXIST | 4005 |
| STORE\_NOT\_FOUND | 4006 |
| ALREDAY\_EXIST\_CODE | 4030 |
| NOT\_EXIST\_CODE | 4031 |
| UNAUTHORIZED | 4032 |
| BAD\_CREDENTIALS | 4033 |
| EXCEPTION\_CODE | 5000 |
| ILLEGAL\_STATUS\_CODE | -1 |

And use-case example:

1. **In case token is expired:**

{ "status": "UNAUTHORIZED", "message": "Access token has expired", "code": 4001 }

**b.In case access\_token is missing:**

           { "status": "UNAUTHORIZED", "message": "Access token is mandatory", "code": 4002 }

**c.** **In case token is invalid:**

          { "status": "UNAUTHORIZED", "message": "Access token is invalid.", "code": 4003 }

1. Design a high-level database design and java class signature for designing appointment system.

**Practo App,**  can be an example.