Environment

* Windows 10
* Python 3.8.0

Setup/ Run Instructions

* Install required modules/ packages from requirements.txt
* Command: pip install -r requirements.txt
* Run database initialisation file
* Command:
* Run test file
* Command

Database

* Household Table
* Individual Table

Database Schema

Household

* Id (Integer), Primary Key
* Individuals (List of Individuals), Not Required
* Housing Type (Enum), Required

Individual

* NRIC (String), Primary Key
* Name (String), Required
* Gender (Enum), Required
* Marital Status (Enum), Required
* Spouse (Individual), Not Required
* Occupation Type (Enum), Required
* Annual Income (Integer), Required
* Date of Birth (Date), Required

Enum/ Enumerations

* Housing Type
  + HDB
  + Condominium
  + Landed
* Gender
  + Male
  + Female
* Marital Status
  + Single
  + Married
  + Widowed
  + Divorced
  + Separated
* Occupation Type
  + Employed
  + Unemployed
  + Student

Assumptions/ Constraints

* Individual can only part of maximum one household
* Individual need not be part of household
* Household need not have individuals
* No limits to the composition of a household
* Spouses need not be in same household
* Individual has to be present in Individual table before individual can be added to a household
* Individual has to be present in Individual table before individual can be indicated as spouse of another individual

Checks

* Valid NRIC check (Checksum letter)
* Annual income >= 0
* Valid date check
* Marital status has to be married/ widowed to have spouse listed
* Spouse has to be of opposite gender
* (Age >= 16 to be married/ widowed/ divorced/ separated)
* (Age >= 16 to be employed)

Root Endpoint

http://localhost:5000

HTTP Methods

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HTTP Method | Operation | Endpoint Path | REST API Response | Description |
| Get | Read | /households  /households?:parameter=:value  /individuals  /individuals?:parameter=:value |  |  |
| Post | Create | /households  {  ‘id’: :id,  (‘individuals’: [‘:nric1’(, ‘:nric2’) …],)  ‘housingtype’: ‘:housingtype’  }  /individuals  {  ‘nric’: ‘:nric’,  ‘name’: ‘:name’,  ‘gender’: ‘:gender’,  ‘maritalstatus’: ‘:maritalstatus’,  (‘spouse’: ‘:spousenric’,)  ‘occupationtype’: ‘:occupationtype’,  ‘annualincome’: :annualincome,  ‘dateofbirth’: ‘:dateofbirth’  } |  |  |
| Put | Update (Replace) | /households  {  ‘id’: :id,  (‘individuals’: [‘:nric1’(, ‘:nric2’) …],)  ‘housingtype’: ‘:housingtype’  }  /individuals  {  ‘nric’: ‘:nric’,  ‘name’: ‘:name’,  ‘gender’: ‘:gender’,  ‘maritalstatus’: ‘:maritalstatus’,  (‘spouse’: ‘:spousenric’,)  ‘occupationtype’: ‘:occupationtype’,  ‘annualincome’: :annualincome,  ‘dateofbirth’: ‘:dateofbirth’  } |  |  |
| Patch | Update (Modify) | /households  {  ‘id’: :id,  (‘individuals’: [‘:nric1’(, ‘:nric2’) …],)  (‘housingtype’: ‘:housingtype’)  }  /individuals  {  ‘nric’: ‘:nric’,  (‘name’: ‘:name’,)  (‘gender’: ‘:gender’,)  (‘maritalstatus’: ‘:maritalstatus’,)  (‘spouse’: ‘:spousenric’,)  (‘occupationtype’: ‘:occupationtype’,)  (‘annualincome’: :annualincome,)  (‘dateofbirth’: ‘:dateofbirth’)  } |  |  |
| Delete | Delete | /households  /households?:parameter=:value  /individuals  /individuals?:parameter=:value |  |  |

Household Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value | Value Examples | Description |
| id | Valid id (int) | * 1 * 2 * 3 | Filter for household of given id |
| nric | Valid NRIC (NRIC format, NRIC checksum letter) | * s1234567d * f1234567n * t1234567j | Filter for household with individual of given NRIC |
| housingtype | Valid housing type   * hdb * condominium * landed | * hdb * condominium * landed | Filter for households with given housing type |
| agebelow | Valid age (0 <= x <= 150) | * 0 * 18 * 21 | Filter for households with individuals below given age |
| ageabove | Valid age (0 <= x <= 150) | * 50 * 65 * 70 | Search for households with individuals above given age |
| spouses | Boolean   * true * false | * true * false | Filter for households with husband and wife |
| householdincomebelow | Valid household income (>= 0) | * 100000 * 150000 * 300000 | Filter for households with household income below given household income |
| householdincomeabove | Valid household income (>= 0) | * 60000 * 100000 * 250000 | Filter for households with household income above given household income |
| grant | Valid grant   * studentencouragement * familytogetherness * elder * babysunshine * yologst | * studentencouragement * familytogetherness * elder * babysunshine * yologst | Filter for households that are eligible for given grant |

Individual Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Value | Value Examples | Description |
| nric | Valid NRIC (NRIC format, NRIC checksum letter) | * s1234567d * f1234567n * t1234567j | Filter for individual of given nric |
| name | String | * abc def * ghi jklmn | Filter for individuals whose name comprises given name |
| gender | Valid gender   * male * female | * male * female | Filter for individuals of given gender |
| maritalstatus | Valid marital status   * single * married * widowed * divorced * separated | * single * married * widowed * divorced * separated | Filter for individuals of given marital status |
| spouse | Valid NRIC (NRIC format, NRIC checksum letter) | * s1234567d * f1234567n * t1234567j | Filter for individuals whose spouse has given NRIC |
| occupationtype | Valid occupation type   * employed * unemployed * student | * employed * unemployed * student | Filter for individuals of given occupationtype |
| annualincomebelow | Valid annual income (>=0) | * 50000 * 60000 * 70000 | Filter for individuals with annual income below given annual income |
| annualincomeabove | Valid annual income (>=0) | * 30000 * 45000 * 50000 | Filter for individuals with annual income above given annual income |
| dateofbirthbefore | Valid date (yyyy-mm-dd, present date or before) | * 1/1/2020 * 11/12/2021 | Filter for individuals whose date of birth falls before given date |
| dateofbirthafter | Valid date (yyyy-mm-dd, present date or before) | * 1/1/1990 * 11/12/2000 | Filter for individuals whose date of birth falls after given date |

HTTP Status Codes