

# API Document:

**BASE URL** : <http://localhost/contacts/v1/api/>

**Example** : <http://localhost/contacts/v1/api/person.create>

**API KEY**: To get the API Key run the **data.php** file

Definition	Description	Restful API URL	Request
E1	Create Person	/person.create	<p>1.HTTP Method: POST</p> <p>2. Parameters:</p> <p>Header: Content-Type : application/json</p> <p>Body:</p> <pre>{   "user":{     "first_name":"Annamalai",     "last_name":"Soman"   },   "emails":[     "annamalai19@gmail.com",     "annamalai19+test@gmail.com"   ],   "phone":{     "home":"1010",     "office":"13434"   },   "address":[     {       "address":"Jurong West",       "city":"Singapore",       "state":"Singapore",       "country":"SG"     },     {       "address":"Holland Village",       "city":"Singapore",       "state":"Singapore",       "country":"SG"     },     {       "address":"Menara Jaya",       "city":"Petaling Jaya",       "state":"Kuala Lumpur",       "country":"MY"     }   ],   "groups":[     "1",     "2"   ] }</pre>

E2	Search Person by First Name, Last Name or Email	/person.list	<b>1.HTTP Method:</b> GET <b>2. Parameters:</b> <b>Header: Content-Type :</b> application/json <b>q:</b> searchText, <b>first_name:</b> searchText <b>last_name:</b> searchText
E3	Update Person Record	/person.update	<b>1.HTTP Method:</b> PUT <b>2. Parameters:</b> <b>Header: Content-Type :</b> application/json <b>Body:</b> {"16":{"user":{"first_name":"Annamalai","last_name":"Soman"}}}
E4	Create Group	/group.create	<b>1.HTTP Method:</b> POST <b>2. Parameters:</b> <b>Header: Content-Type :</b> application/json <b>Body:</b> {"group":{"name":"Friends"}}
E5	Add Person to Group	/group.join	<b>1.HTTP Method:</b> PUT <b>2. Parameters:</b> <b>Header: Content-Type :</b> application/json <b>Body:</b> {"groups":{"join":[{"user_id":1,"group_id":1},{"user_id":1,"group_id":2},{"user_id":1,"group_id":5}], "remove":[]}}
E6		/group.update	To do
E7		/person.delete	To do

Any code that is a multiple of 100 or 10 is a catch-all for the codes below it, and may be split off into more specific codes in the future. The error codes that are most likely to change are 200, 320, and 330.

Code	Error	Description
100	Missing Argument	Missing the Required Argument
200	Person not created	A required argument was not supplied or Data not inserted to the database
201	Person not updated	A required argument was not supplied or Data not updates to the database
202	Group not created!	A required argument was not supplied or Data not inserted to the database
404	Invalid Entity Point	Missing the url action names

## HOW TO RUN THE REST API CALLS : Example shows for Creating Person records

The screenshot displays the Postman interface for a REST API call. The call is a POST request to the URL `http://localhost/contacts/v1/api/person.create`. The method is POST, and the data format is JSON (application/json). The request body contains a JSON object representing a person record. The response is a JSON object indicating the person was created successfully.

**1. URL**: `http://localhost/contacts/v1/api/person.create`

**2. METHOD**: POST

**DATA FORMAT**: JSON (application/json)

**4. REQUEST DATA IN BODY**

```
{
  "user": {
    "first_name": "Annamalai",
    "last_name": "Soman",
    "email": [
      "annamalai19@gmail.com",
      "annamalai19-test@gmail.com"
    ],
    "phone": {
      "home": "1010",
      "office": "13434"
    },
    "address": [
      {
        "address": "Jurong West",
        "city": "Singapore",
        "state": "Singapore",
        "country": "SG"
      },
      {
        "address": "Holland Village",
        "city": "Singapore",
        "state": "Singapore",
        "country": "SG"
      },
      {
        "address": "Petaling Jaya",
        "city": "Kuala Lumpur",
        "state": "MY",
        "country": "MY"
      }
    ],
    "groups": [
      "1",
      "2"
    ]
  }
}
```

**RESPONSE**

```
{
  "body": "To do",
  "status": true,
  "message": "Person created successfully!"
}
```