

Person API

Keni – kehinde.adewusi@gmail.com

Table of Contents

1. CRUD Operations

1.1. Create a new person

1.2. Get All

1.3. Get one

1.4. Update Person Entity

1.5. Delete Person

Colophone

This Person API documentation will come in handy for integrators and front-end developers.

The service will be running at a URL such as <http://localhost:8080/> and other URLs in the documentation will be relative to this root.

TIP

The APIs have been arranged in a somewhat logical order from top to bottom.

1. CRUD Operations

This outlines the CRUD operations available in the API.

1.1. Create a new person

This API creates a new person entity.

request

HTTP

```
POST /persons/ HTTP/1.1
Content-Type: application/json
Content-Length: 100
Host: localhost:8080
```

```
{
  "age" : 25,
  "first_name" : "Harry",
  "last_name" : "Mcguire",
  "favourite_colour" : "BLUE"
}
```

response

HTTP

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 112
```

```
{
  "id" : 5,
  "age" : 25,
  "first_name" : "Harry",
  "last_name" : "Mcguire",
  "favourite_colour" : "BLUE"
}
```

1.2. Get All

Returns a list of existing person objects.

See Create to add a person entity to the existing records.

request

HTTP

```
GET /persons/ HTTP/1.1
Host: localhost:8080
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 397
```

```
{
  "person" : [ {
    "id" : 1,
    "age" : 25,
    "first_name" : "Joy",
    "last_name" : "Burgundy",
    "favourite_colour" : "RED"
  }, {
    "id" : 2,
    "age" : 29,
    "first_name" : "James",
    "last_name" : "Joyce",
    "favourite_colour" : "GREEN"
  }, {
    "id" : 3,
    "age" : 27,
    "first_name" : "Bruno",
    "last_name" : "Fernandez",
    "favourite_colour" : "GREEN"
  } ]
}
```

TIP

The Get All API provides the original version at a different endpoint, it doesn't include an ID for each person.

request

```
GET /persons/v0 HTTP/1.1
Host: localhost:8080
```

response

HTTP

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=UTF-8
Content-Length: 355
```

```
{
  "person" : [ {
    "first_name" : "Joy",
    "last_name" : "Burgundy",
    "age" : 25,
    "favourite_colour" : "RED"
  }, {
    "first_name" : "James",
    "last_name" : "Joyce",
    "age" : 29,
    "favourite_colour" : "GREEN"
  }, {
    "first_name" : "Bruno",
    "last_name" : "Fernandez",
    "age" : 27,
    "favourite_colour" : "GREEN"
  } ]
}
```

1.3. Get one

Returns a single person identified by the ID provided in the path.

request

```
GET /persons/2/ HTTP/1.1
Host: localhost:8080
```

HTTP

response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 111
```

HTTP

```
{
  "id" : 2,
  "age" : 29,
  "first_name" : "James",
  "last_name" : "Joyce",
  "favourite_colour" : "GREEN"
}
```

1.4. Update Person Entity

Updates an existing person record. Returns a 404 if the person is not found.

request

```
PUT /persons/3/ HTTP/1.1
Content-Type: application/json
Content-Length: 101
Host: localhost:8080
```

HTTP

```
{
  "age" : 27,
  "first_name" : "Bruno",
  "last_name" : "Fernandez",
  "favourite_colour" : "RED"
}
```

response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 113
```

HTTP

```
{
  "id" : 3,
  "age" : 27,
  "first_name" : "Bruno",
  "last_name" : "Fernandez",
  "favourite_colour" : "RED"
}
```

1.5. Delete Person

Deletes an existing person entity. Returns a 404 if the person is not found.

request

```
DELETE /persons/2 HTTP/1.1
Host: localhost:8080
```

HTTP

response

```
HTTP/1.1 200 OK
```

HTTP

Note: Deleting a non-existing person returns a 404.

request

DELETE /persons/15 HTTP/1.1
Host: localhost:8080

HTTP

response

HTTP/1.1 404 Not Found

HTTP

Colophone

Keni kehinde.adewusi@gmail.com

Last updated 2021-02-14 12:20:06 WAT