

## FAMILY

API	Request	Sub-path	Description	Exception
/families	GET		<ul style="list-style-type: none"> <li>It is used to get a Collection of families.</li> <li><u>Streetname</u> and <u>houenumber</u> can be used as request parameters.</li> </ul> <p>.../families .../families?streetname=Thorsgade&amp;houenumber=2</p>	<p>May return 500 Internal Server Error (if something fails)</p> <p>May return 404 Not Found (if provided streetname and houenumber do not match any family)</p>
	PUT		<p>It is used to create a new Family. Provide the Family as json in the request body.</p> <p>.../families</p>	<p>May return 400 Bad Request( if the data is invalid)</p> <p>May return 500 Internal Server Error (if something fails)</p>
	DELETE		<p>It is used to delete a Family. The deleted family is returned. Provide two query parameters: streetname, houenumber</p> <p>.../families?streetname=Thorsgade&amp;houenumber=2</p>	<p>May return 500 Internal Server Error (if something fails)</p> <p>May return 404 Not Found (if provided streetname and houenumber does not match any family)</p>

The Family is sent back and forth as json. A Family has a collection of Adults, a collection of Children, and a collection of Pets.

The format of a Family is seen in the example below.

```
{
  "streetName": "Francis IIStrreet",
  "houseNumber": 1,
  "adults": [
    {
      "jobTitle": "Animal Nutritionist",
      "id": 13,
      "firstName": "Rina",
      "lastName": "May",
      "hairColor": "Brown",
      "eyeColor": "Brown",
      "age": 33,
      "weight": 56.2,
      "height": 148,
      "sex": "F"
    },
    {
      "jobTitle": "Environmental Engineer",
      "id": 14,
      "firstName": "Raina",
      "lastName": "May",
      "hairColor": "Black",
      "eyeColor": "Hazel",
      "age": 47,
      "weight": 129.6,
      "height": 152,
      "sex": "F"
    }
  ],
  "children": [
    {
      "childInterests": [
        {
          "childId": 14,
          "interestId": "Hockey"
        }
      ],
      "pets": [],
      "id": 14,
      "firstName": "Dawson",
      "lastName": "May",
      "hairColor": "Black",
      "eyeColor": "Brown",
      "age": 3,
      "weight": 23,
      "height": 100,
      "sex": "M"
    },
    {
      "childInterests": [
        {
          "childId": 15,
          "interestId": "Kite Flying"
        },
        {
          "childId": 15,
          "interestId": "Board Games"
        },
        {
          "childId": 15,
          "interestId": "Roller Blading"
        },
        {
          "childId": 15,
          "interestId": "Drawing"
        }
      ],
      "pets": [],
      "id": 15,
      "firstName": "Skyler",
      "lastName": "May",
      "hairColor": "Brown",
      "eyeColor": "Brown",
      "age": 9,
      "weight": 42.670002,
      "height": 113,
      "sex": "F"
    }
  ],
  "pets": []
}
```

## ADULT

API	Request	Sub-path	Description	Exception
/adults	GET		<ul style="list-style-type: none"><li>It is used to get a Collection of adults.</li><li>Firstname, lastname and id can be used as request parameters.</li></ul> .../adults .../adults?id=2 .../adults?firstname=kailey .../adults?lastname=dalton	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id,firstname or lastname do not match any adult)
	PUT		It is used to create a new Adult. Provide the adult as json in the request body.  .../adults	May return 400 Bad Request( if the data is invalid) May return 500 Internal Server Error (if something fails)
	DELETE	/id:int}	It is used to delete an adult. The deleted adult is returned. Provide query parameter id.  .../adults/5 .../adults?id=5	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id,firstname or lastname do not match any adult)

The Adult is sent back and forth as json. The json format is seen in the example below:

```
{"jobTitle":"Painter","id":40,"firstName":"Mikael","lastName":"Mejia","hairColor":"Black","eyeColor":"Blue","age":49,"weight":76.2,"height":168,"sex":"M"}
```

## CHILD

API	Request	Sub-path	Description	Exception
/children	GET		<ul style="list-style-type: none"><li>It is used to get a Collection of children.</li><li>Firstname, lastname and id can be used as request parameters.</li></ul> .../children .../children?firstname=peter .../children?lastname=smith .../children?id=3	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id,firstname or lastname do not match any child)
	PUT		It is used to create a new Child. Provide the child as json in the request body.  .../children	May return 400 Bad Request( if the data is invalid) May return 500 Internal Server Error (if something fails)
	DELETE	/id:int}	It is used to delete a child. The deleted child is returned. Provide query parameter id.  .../children/5 .../children?id=5	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id,firstname or lastname do not match any adult)

The Child is sent back and forth as json. A child has a collection of ChildInterests, and a collection of Pets.

The format of a Child is seen in the example below.

```
{ "childInterests": [ { "childId": 2, "interestId": "Kite Flying" },  
  { "childId": 2, "interestId": "Drawing" } ], "pets": [ { "id": 0, "species": "Bunny", "name": "helli", "age": 1 }, { "id": 2, "firstName":  
  "Mathilde", "lastName": "Larsen", "hairColor": "Blond", "eyeColor": "Green", "age": 9, "weight": 140, "height": 34, "sex": "F" }
```

## PET

API	Request	Sub-path	Description	Exception
/pets	GET		<ul style="list-style-type: none"><li>It is used to get a Collection of pets.</li><li>Name and id can be used as request parameters.</li></ul> .../pets .../pets?name=bobo .../pets?id=3	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id or name do not match any pet)
	PUT		It is used to create a new Pet. Provide the pet as json in the request body.  .../pets	May return 400 Bad Request( if the data is invalid) May return 500 Internal Server Error (if something fails)
	DELETE	/id: int}	It is used to delete a pet. The deleted pet is returned. Provide query parameter id.  .../pets/5 .../pets?id=5	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided id or name do not match any pet)

The Pet is sent back and forth as json. The format of a Pet is seen in the example below:

```
{"id":1,"species":"Cat","name":"Cutie","age":1}
```

## USER

API	Request	Sub-path	Description	Exception
/users	GET		<ul style="list-style-type: none"><li>It is used to get a Collection of users.</li><li>username and usertype can be used as request parameters.</li></ul> .../users .../ users?usertype=admin .../users?username=philip	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided usertype or username do not match any user)
	PUT		It is used to create a new User. Provide the user as json in the request body.  .../users	May return 400 Bad Request( if the data is invalid) May return 500 Internal Server Error (if something fails)
	DELETE	/ {id:int}	It is used to delete an user. The deleted user is returned. Provide query parameter username.  .../users?username=philip	May return 500 Internal Server Error (if something fails)  May return 404 Not Found (if provided usertype or username do not match any user)

The User is sent back and forth as json. The format of a User is seen in the example below:

```
{"userName":"admin2","password":"123456","userType":"admin"}
```

**Natali Munk-Jakobsen**