

Github link: [https://github.com/Aldina28/db\\_task/](https://github.com/Aldina28/db_task/)

Swagger link: <http://127.0.0.1:8000/swagger/>

## TASK 2

### Models

#### 1) Control

- Represents a control entity with a unique name and description.
- Fields:
  - name: Unique identifier for the control (primary key).
  - description: Descriptive text about the control.

#### 2) ControlSetReference

- Represents a reference to a control set, with a unique reference ID and name.
- Fields:
  - reference\_id: Unique identifier for the reference.
  - name: Name of the control set reference (primary key).

Automated creation of related objects on model save.

**Trigger:** Control creation.

**Action:** Creates a ControlSetReference object.

(Creates a ControlSetReference object when a Control is created.)

#### 3) ControlSet

- Represents a set of controls with a unique slug, name, and hierarchy depth.
- Fields:
  - slug: Unique identifier generated using uuid.
  - name: Unique name for the control set.
  - hierarchy\_depth: Indicates the depth of hierarchy.

#### 4) ControlHierarchy

- Represents the hierarchical structure of controls, including references to parent and child controls.
- Fields:
  - slug: Unique identifier generated using uuid.
  - control\_set: Many-to-many relationship with ControlSetReference, allowing multiple references.
  - parents: List of parent controls.
  - children: List of child controls.

Automated creation of related objects on model save.

**Trigger:** ControlSet creation.

**Action:** Creates a ControlHierarchy object.

## Serializers

### 1) ControlModelSerializer

- A serializer for the Control model.
- Fields:

    name: CharField with a regex validator to ensure it contains only alphabetic characters.  
    description: CharField for the control's description.

### 2) ControlsetReferenceModelSerializer

- A serializer for the ControlSetReference model.
- Fields:

    reference\_id: CharField for the unique reference ID.  
    name: CharField for the name, which must exist in the Control model.

### 3) ControlsetModelSerializer

- A serializer for the ControlSet model.
- Fields:

    slug: Read-only CharField for the unique identifier (UUID).  
    name: CharField with a regex validator to ensure it contains only alphabetic characters.  
    hierarchy\_depth: IntegerField with a minimum value validator.

### 4) ControlHierarchyModelSerializer

- A serializer for the ControlHierarchy model.
- Fields:

    control\_set: Nested serializer for ControlSetReference, optional and can have multiple references.  
    parents: Custom ListField for parent controls.  
    children: Custom ListField for child controls.

- Methods:
  - validate: Ensures the hierarchical depth constraints for parents and children.
  - update: Updates the ControlHierarchy instance with validated data, handling nested control\_set relationships and updating parents and children lists.

## API Views

The API handles the creation, updating, deletion, and retrieval of Control, ControlSet, ControlSetReference, and ControlHierarchy objects. The relationships between these objects ensure that updates and deletions propagate correctly across the related models.

## 1. ControlCreateAPI

- Purpose: Creates a new Control object.
- Process:

A Control object is created using the data provided.

If the creation is successful, the response includes the details of the created Control object.

If a Control with the same name already exists, an error message is returned.

For calling this API use POST [http://127.0.0.1:8000/control\\_create/](http://127.0.0.1:8000/control_create/)

Request:

```
{  
    "name": "Forest",  
    "description": "description of the forest"  
}
```

Response:

```
{  
    "name": "Forest",  
    "description": "description of the forest"  
}
```

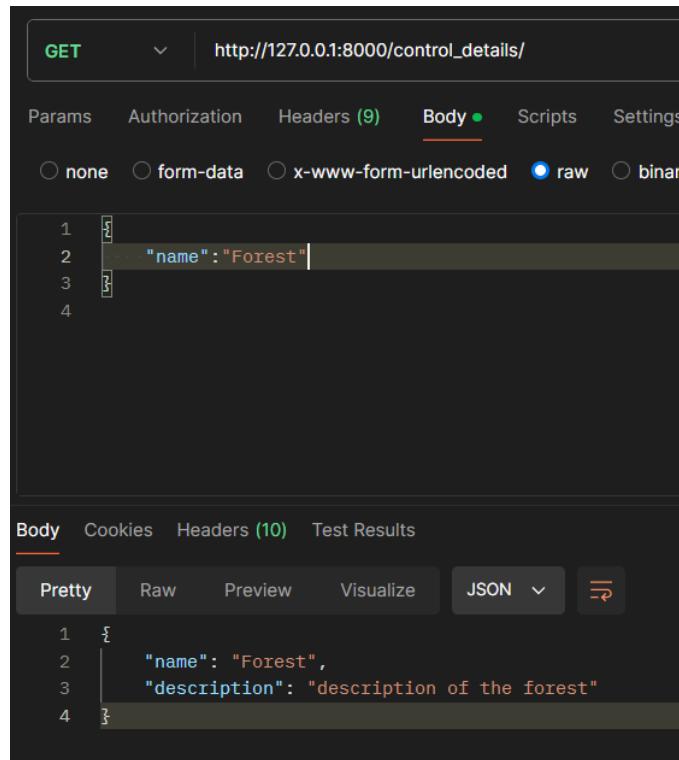
The screenshot shows a POST request to the URL `http://127.0.0.1:8000/control_create/`. The request body is a JSON object with fields `"name": "Forest"` and `"description": "description of the forest"`. The response body is identical to the request body, indicating a successful creation of the Control object.

The screenshot shows a POST request to the URL `http://127.0.0.1:8000/control_create/`. The request body is a JSON object with fields `"name": "Forest"` and `"description": "description of the flower"`. The response body contains an error message: `"error": "Failed to create Control, Control name already exist"`, indicating that a Control object with the name "Forest" already exists.

## 2. AllControlDetailsAPI

- Purpose: Retrieves details of all Control objects or a specific object by its name.
- Process:
  - If a name is provided, the specific Control object is retrieved.
  - If no name is provided, all Control objects are retrieved.
  - If the specified Control is not found, an error message is returned.

For calling this API use GET [http://127.0.0.1:8000/control\\_details/](http://127.0.0.1:8000/control_details/)



GET http://127.0.0.1:8000/control\_details/

Params Authorization Headers (9) Body Scripts Settings

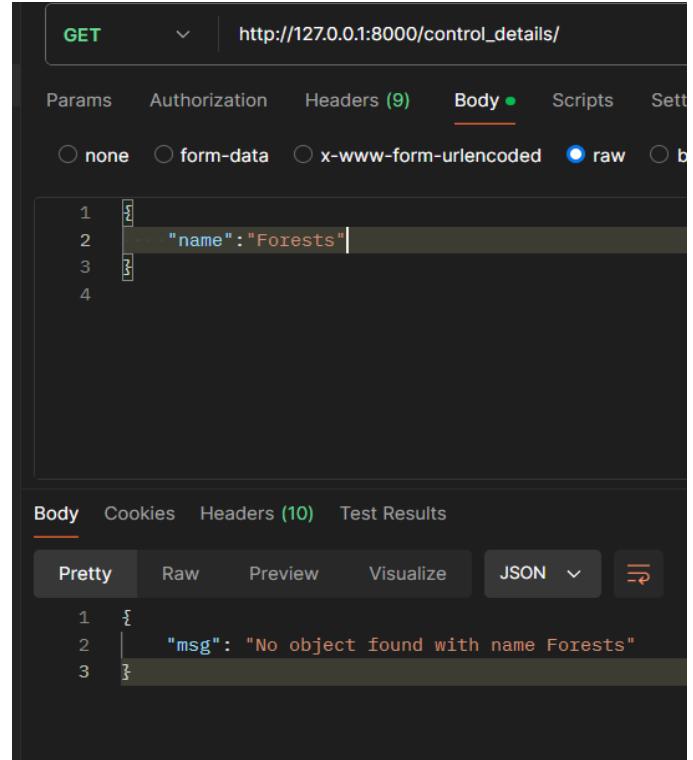
none  form-data  x-www-form-urlencoded  raw  binary

```
1 {  
2   "name": "Forest"  
3 }  
4
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "name": "Forest",  
3   "description": "description of the forest"  
4 }
```



GET http://127.0.0.1:8000/control\_details/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary

```
1 {  
2   "name": "Forests"  
3 }  
4
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2   "msg": "No object found with name Forests"  
3 }
```

The screenshot shows a Postman interface with a GET request to `http://127.0.0.1:8000/control_details/`. The Body tab is selected, showing a raw JSON response. The response is a list of three objects:

```
1  [
2    {
3      "name": "Forest",
4      "description": "description of the forest"
5    },
6    {
7      "name": "Flower",
8      "description": "description of the flower"
9    },
10   {
11     "name": "Animal",
12     "description": "description of the animal"
13   }
14 ]
```

### 3. ControlUpdateAPI

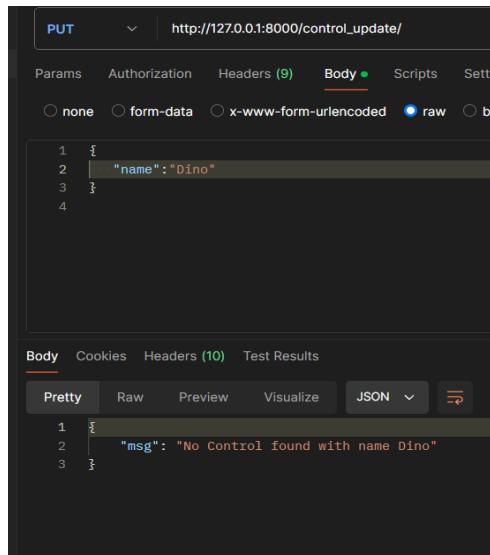
- Purpose: Updates an existing Control object.
- Process:

The specified Control object is updated with the provided data.

Any associated ControlSetReference objects are also updated.

If no Control object with the specified name is found, an error message is returned.

For calling this API use [PUT `http://127.0.0.1:8000/control\_update/`](http://127.0.0.1:8000/control_update/)



PUT | http://127.0.0.1:8000/control\_update/

Params Authorization Headers (9) **Body** Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary

```
1 {
2   "name": "Dino"
3 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize **JSON**

```
1 {}
2   "msg": "No Control found with name Dino"
3 }
```

#### 4. ControlDeleteAPI

- Purpose: Deletes a Control object by its name.
- Process:

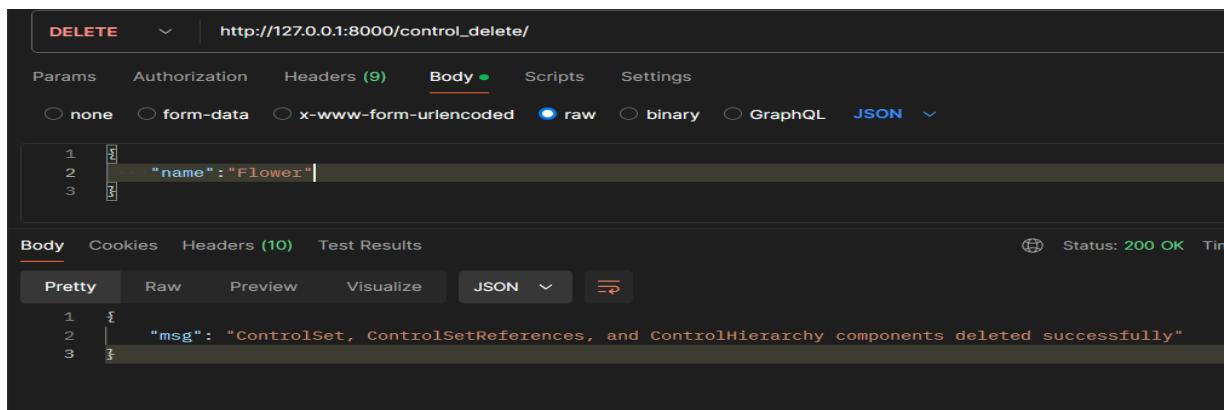
The specified Control object is deleted.

Any associated ControlSetReference objects are also deleted.

The Control object is removed from any ControlHierarchy components where it is referenced.

If no Control object with the specified name is found, an error message is returned.

For calling this API use [DELETE http://127.0.0.1:8000/control\\_delete/](http://127.0.0.1:8000/control_delete/)



DELETE | http://127.0.0.1:8000/control\_delete/

Params Authorization Headers (9) **Body** Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**

```
1 {}
2   "name": "Flower"
3 }
```

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 0ms

Pretty Raw Preview Visualize **JSON**

```
1 {}
2   "msg": "ControlSet, ControlSetReferences, and ControlHierarchy components deleted successfully"
3 }
```

The image shows two side-by-side screenshots of a REST API testing tool interface, likely Postman. Both screenshots display a GET request to `http://127.0.0.1:8000/control_details/`.

**Left Screenshot:**

- Body:** Contains the JSON payload:
 

```

1 {
2   "name": "Flower"
3 }
```
- Response Body:** Shows the response in JSON format:
 

```

1 {
2   "msg": "No object found with name Flower"
3 }
```

**Right Screenshot:**

- Body:** Contains the same JSON payload as the left screenshot.
- Response Body:** Shows the response in JSON format:
 

```

1 {
2   "msg": "No object found with name Flower"
3 }
```

The image shows a screenshot of a REST API testing tool interface, likely Postman. The request URL is `http://127.0.0.1:8000/controlhierarchies_details/`.

**Body:** Contains the JSON payload:

- 1 [
- 2 {
- 3 "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",
- 4 "control\_set": [
- 5 {
- 6 "name": "Forest",
- 7 "reference\_id": "FR20",
- 8 "control\_name": "Forest",
- 9 "description": "description of the forest"
- 10 }
- 11 ],
- 12 "parents": [
- 13 "None"
- 14 ],
- 15 "children": [
- 16 "None"
- 17 ],
- 18 "control\_set\_name": "Science"
- 19 },
- 20 ]

(Deletion of flower in Control is reflected in all the other tables also)

## 5. ControlsetRefUpdateAPI

- Purpose: Updates a ControlSetReference object.
- Process:

The specified ControlSetReference object is updated with the new reference ID.

The update is reflected in all related ControlHierarchy objects.

For calling this API use POST [http://127.0.0.1:8000/controlsetreference\\_update/](http://127.0.0.1:8000/controlsetreference_update/)

Request body

```
{  
    "name": "Forest",  
    "reference_id": "FR20"  
}
```

Response

```
{  
    "msg": "ControlSetReference and ControlHierarchy components updated successfully"  
}
```

The screenshot shows a Postman interface with the following details:

- Method:** PUT
- URL:** http://127.0.0.1:8000/controlsetreference\_update/
- Body (JSON):**

```
1 {  
2     "name": "Forest",  
3     "reference_id": "FR20"  
4 }
```
- Response (Pretty):**

```
1 {  
2     "msg": "ControlSetReference and ControlHierarchy components updated successfully"  
3 }
```

## 6. AllControlsetRefDetailsAPI

- Purpose: Retrieves details of all ControlSetReference objects or a specific object by its name.
- Process:
  - If a name is provided, the specific ControlSetReference object is retrieved.
  - If no name is provided, all ControlSetReference objects are retrieved.
  - If the specified ControlSetReference is not found, an error message is returned.

For calling this API use GET [http://127.0.0.1:8000/controlsetreference\\_details/](http://127.0.0.1:8000/controlsetreference_details/)

The image shows two side-by-side Postman request panels. Both requests are for the URL `http://127.0.0.1:8000/controlsetreference_details/`.

**Request 1 (Left):** The body contains a JSON object with a single key-value pair: `"name": "Animal"`. The response (Pretty) shows a JSON object with keys `"name": "Animal"` and `"reference_id": "AL20"`.

```
1 {  
2   "name": "Animal"  
3 }  
1 {  
2   "name": "Animal",  
3   "reference_id": "AL20"  
4 }
```

**Request 2 (Right):** The body contains a JSON object with a single key-value pair: `"name": "Animals"`. The response (Pretty) shows a JSON object with a single key `"msg": "No object found with name Animals"`.

```
1 {  
2   "name": "Animals"  
3 }  
1 {  
2   "msg": "No object found with name Animals"  
3 }
```

The image shows a single Postman request panel for the URL `http://127.0.0.1:8000/controlsetreference_details/`.

The body contains a JSON array with three objects, each representing a ControlSetReference:

```
1 [  
2   {  
3     "name": "Forest",  
4     "reference_id": "FR20"  
5   },  
6   {  
7     "name": "Flower",  
8     "reference_id": "FL20"  
9   },  
10  {  
11    "name": "Animal",  
12    "reference_id": "AL20"  
13  }  
14 ]
```

The response status is 200 OK, time is 757 ms, size is 445 B, and it includes a "Save as example" button.

## 7. ControlSetCreateAPI

- Purpose: Creates a new ControlSet object.
- Process:

A ControlSet object is created using the data provided.

If the creation is successful, the response includes the details of the created ControlSet object.

If an error occurs, an error message is returned.

For calling this API use POST [http://127.0.0.1:8000/controlset\\_create/](http://127.0.0.1:8000/controlset_create/)

Request:

```
{  
    "name": "Biology",  
    "hierarchy_depth": 1  
}
```

Response:

```
{  
    "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",  
    "name": "Biology",  
    "hierarchy_depth": 1  
}
```

POST http://127.0.0.1:8000/controlset\_create/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary

```
1 {  
2     "name": "Science",  
3     "hierarchy_depth": 0  
4 }  
5
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2     "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",  
3     "name": "Science",  
4     "hierarchy_depth": 0  
5 }
```

POST http://127.0.0.1:8000/controlset\_create/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  G

```
1 {  
2     "name": "Social",  
3     "hierarchy_depth": -10  
4 }  
5
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```
1 {  
2     "hierarchy_depth": [  
3         "Ensure this value is greater than or equal to 0."  
4     ]  
5 }
```

## 8. ControlSetUpdateAPI

- Purpose: Updates an existing ControlSet object.
- Process:

The specified ControlSet object is updated with the provided data.

If no ControlSet object with the specified name is found, an error message is returned.

For calling this API use PUT [http://127.0.0.1:8000/controlset\\_update/](http://127.0.0.1:8000/controlset_update/)

### Request

```
{  
    "name": "Science",  
    "hierarchy_depth": 2  
}
```

### Response

```
{  
    "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",  
    "name": "Science",  
    "hierarchy_depth": 2  
}
```

PUT http://127.0.0.1:8000/controlset\_update/

Body

```
{  
  "name": "Science",  
  "hierarchy_depth": 2  
}
```

Body

```
{  
  "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",  
  "name": "Science",  
  "hierarchy_depth": 2  
}
```

PUT http://127.0.0.1:8000/controlset\_update/

Body

```
{  
  "name": "Sciences",  
  "hierarchy_depth": 2  
}
```

Body

```
{"msg": "No ControlSet found with name Sciences"}
```

## 9. AllControlSetDetailsAPI

- Purpose: Retrieves details of all ControlSet objects or a specific ControlSet by its name.
- Process:
  - If a name is provided, the specific ControlSet object is retrieved.
  - If no name is provided, all ControlSet objects are retrieved.
  - If the specified ControlSet is not found, an error message is returned.

For calling this API use GET [http://127.0.0.1:8000/controlset\\_details/](http://127.0.0.1:8000/controlset_details/)

### Request

```
{  
}
```

### Response

```
[  
  {  
    "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",  
    "name": "Science",  
    "hierarchy_depth": 0  
  },  
  {  
    "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",  
    "name": "Biology",  
    "hierarchy_depth": 1  
  },  
  {  
    "slug": "639e5457-66a6-4035-b3e2-5ff577096052",  
    "name": "Botony",  
    "hierarchy_depth": 2  
  },  
  {  
    "slug": "5a2ce284-385c-4f27-bdfe-ceefb106b2f6",  
    "name": "History",  
    "hierarchy_depth": 1  
  },  
  {  
    "slug": "f80b70a9-75e5-4d63-b324-d6f658463944",  
    "name": "World War",  
    "hierarchy_depth": 2  
  }
```

```
}
```

GET http://127.0.0.1:8000/controlset\_details/

Params Authorization Headers (9) Body Scripts Settings

None form-data x-www-form-urlencoded raw binary Graph

```
1 [ ]  
2 [ ]  
3 [ ]  
4 [ ]
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
2 {  
3   "slug": "86877d42-c0d6-4895-9bb3-0e6ddde36e27e",  
4   "name": "Science",  
5   "hierarchy_depth": 0  
6 },  
7 {  
8   "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",  
9   "name": "Biology",  
10  "hierarchy_depth": 1  
11 },  
12 {  
13   "slug": "639e5457-66a6-4035-b3e2-5ff577096052",  
14   "name": "Botony",  
15   "hierarchy_depth": 2  
16 },  
17 {  
18   "slug": "5a2ce284-385c-4f27-bdfe-ceefb106b2f6",  
19   "name": "History",  
20   "hierarchy_depth": 1  
21 },
```

GET http://127.0.0.1:8000/controlset\_details/

Params Authorization Headers (9) Body Scripts Settings

None form-data x-www-form-urlencoded raw binary Graph

```
1 [ ]  
2 [ "name": "Math" ]  
3 [ ]  
4 [ ]
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "msg": "No object found with name Math"  
3 }
```

## 10. ControlSetDeleteAPI

- Purpose: Deletes a ControlSet object by its name.

- Process:

The specified ControlSet object is deleted.

Any associated ControlHierarchy components are updated accordingly.

If no ControlSet object with the specified name is found, an error message is returned.

For calling this API use DELETE [http://127.0.0.1:8000/controlset\\_delete/](http://127.0.0.1:8000/controlset_delete/)

The image displays two side-by-side Postman requests. The left request is a DELETE to `http://127.0.0.1:8000/controlset_delete/` with a raw JSON body containing `{"name": "Science"}`. The right request is a GET to `http://127.0.0.1:8000/controlhierarchies_details/`, also with a raw JSON body containing `{"name": "Science"}`. Both requests show a successful response with a message indicating the object was deleted.

(Deletion of object in Control Set is reflected in Control Hierarchy)

## 11. ControlHierarchyUpdateAPI

- Purpose: Updates a ControlHierarchy object.
- Process:

The specified ControlHierarchy object is updated with the provided data.

If no ControlHierarchy object with the specified slug is found, an error message is returned.

For calling this API use PUT [http://127.0.0.1:8000/controlhierarchies\\_update/](http://127.0.0.1:8000/controlhierarchies_update/)

### Request

```
{
  "name": "Biology",
  "control_set": [
    {"reference_id": "FR20"}, {"reference_id": "FL20"}
  ],
  "parents": ["Science"],
  "children": ["Botony"]
}
```

### Response

```
{
  "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",
  "control_set": [
```

```
{  
    "name": "Flower",  
    "reference_id": "FL20"  
,  
{  
    "name": "Forest",  
    "reference_id": "FR20"  
}  
,  
"parents": [  
    "Science"  
,  
    "children": [  
        "Botony"  
    ]  
}
```

PUT [http://127.0.0.1:8000/controlhierarchies\\_update/](http://127.0.0.1:8000/controlhierarchies_update/)

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQL [JSON](#)

```
1 {  
2     "name": "Biology",  
3     "control_set": [  
4         {"reference_id": "FT20"},  
5         {"reference_id": "FL28"}  
6     ],  
7     "parents": ["Biology", "Science", "Zoology"],  
8     "children": ["Botony", "History"]  
9 }
```

Body Cookies Headers (10) Test Results

Status: 400 Bad Request Time: 1505 ms Size: 458 B

Pretty Raw Preview Visualize [JSON](#)

```
1 {  
2     "non_field_errors": [  
3         "Parent ControlSet 'Biology' has greater hierarchy depth than the current ControlSet(It can't be a parent)."  
4     ]  
5 }
```

The image shows two side-by-side Postman request panels. Both requests are to `http://127.0.0.1:8000/controlhierarchies_update/` using the `PUT` method.

**Request 1 (Left):**

- Body:** Raw JSON (Pretty printed)

```

1 {
2   "name": "Biology",
3   "control_set": [
4     {"reference_id": "FT20"}, 
5     {"reference_id": "FL28"}
6   ],
7   "parents": ["Science", "Zoology"],
8   "children": ["Botony", "History"]
9 }
```

**Request 2 (Right):**

- Body:** Raw JSON (Pretty printed)

```

1 {
2   "name": "Biology",
3   "control_set": [
4     {"reference_id": "FT20"}, 
5     {"reference_id": "FL28"}
6   ],
7   "parents": ["Science"], 
8   "children": ["Botony"]
9 }
```

**Test Results (Left):**

```

1 {
2   "non_field_errors": [
3     "Parent ControlSet 'Zoology' does not exist."
4   ]
5 }
```

**Test Results (Right):**

```

1 [
2   "ControlSetReference FT20 does not exist"
3 ]
```

The image shows a single Postman request panel for `http://127.0.0.1:8000/controlhierarchies_update/` using the `PUT` method.

**Body:** Raw JSON (Pretty printed)

```

1 {
2   "name": "Biology",
3   "control_set": [
4     {"reference_id": "FT20"}, 
5     {"reference_id": "FL28"}
6   ],
7   "parents": ["Science"], 
8   "children": ["Botony", "History"]
9 }
```

**Status Bar:** Status: 400 Bad Request Time: 2.07 s Size

**Test Results:**

```

1 {
2   "non_field_errors": [
3     "Child ControlSet 'History' has lesser hierarchy depth than the current ControlSet(It can't be a child)."
4   ]
5 }
```

PUT http://127.0.0.1:8000/controlhierarchies\_update/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary

```
1 {  
2   "name": "Biology",  
3   "control_set": [  
4     {"reference_id": "FR20"},  
5     {"reference_id": "FL20"}  
6   ],  
7   "parents": ["Science"],  
8   "children": ["Botany"]  
9 }
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↗

```
2 "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",  
3 "control_set": [  
4   {  
5     "name": "Flower",  
6     "reference_id": "FL20"  
7   },  
8   {  
9     "name": "Forest",  
10    "reference_id": "FR20"  
11  }  
12 ],  
13 "parents": [  
14   "Science"  
15 ],  
16 ...
```

```
        ],
      "children": [
        "Botony"
      ]
    }
}
```

## 12. AllControlHierarchiesDetailsAPI

- Purpose: Retrieves details of all ControlHierarchy objects or a specific ControlHierarchy by its name.
  - Process:
    - If a name is provided, the specific ControlHierarchy object is retrieved. The ControlSet details are included in the response, along with the Control name and description.
    - If no name is provided, all ControlHierarchy objects are retrieved.

For calling this API use GET [http://127.0.0.1:8000/controlhierarchies\\_details/](http://127.0.0.1:8000/controlhierarchies_details/)

## Request

```
{  
}  
}
```

## Response

```
[  
 {  
   "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",  
   "control_set": [  
     {  
       "name": "Animal",  
       "reference_id": "AL20",  
       "control_name": "Animal",  
       "description": "description of the animal"  
     },  
     {  
       "name": "Flower",  
       "reference_id": "FL20",  
       "control_name": "Flower",  
       "description": "description of the flower"  
     },  
     {  
       "name": "Forest",  
       "reference_id": "FR20",  
       "control_name": "Forest",  
       "description": "description of the forest"  
     }  
   ],  
   "parents": [  
     "None"  
   ],  
   "children": [  
     "None"  
   ],  
   "control_set_name": "Science"  
 },  
 {  
   "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",  
   "control_set": [  
     {  
       "name": "Animal",  
       "reference_id": "AL20",  
       "control_name": "Animal",  
       "description": "description of the animal"  
     },  
     {  
       "name": "Flower",  
       "reference_id": "FL20",  
       "control_name": "Flower",  
       "description": "description of the flower"  
     },  
     {  
       "name": "Forest",  
       "reference_id": "FR20",  
       "control_name": "Forest",  
       "description": "description of the forest"  
     }  
   ],  
   "parents": [  
     "None"  
   ],  
   "children": [  
     "None"  
   ],  
   "control_set_name": "Science"  
 }]
```

```
        "reference_id": "FL20",
        "control_name": "Flower",
        "description": "description of the flower"
    },
    {
        "name": "Forest",
        "reference_id": "FR20",
        "control_name": "Forest",
        "description": "description of the forest"
    }
],
"parents": [
    "Science"
],
"children": [
    "Botony"
],
"control_set_name": "Biology"
},
{
    "slug": "639e5457-66a6-4035-b3e2-5ff577096052",
    "control_set": [
        {
            "name": "Animal",
            "reference_id": "AL20",
            "control_name": "Animal",
            "description": "description of the animal"
        }
    ],
    "parents": [
        "Science",
        "Biology"
    ],
    "children": [
        ""
    ],
    "control_set_name": "Botony"
},
{
    "slug": "5a2ce284-385c-4f27-bdfe-ceefb106b2f6",
    "control_set": [],
    "parents": [
```

```

        "None"
    ],
    "children": [
        "None"
    ],
    "control_set_name": "History"
},
{
    "slug": "f80b70a9-75e5-4d63-b324-d6f658463944",
    "control_set": [],
    "parents": [
        "None"
    ],
    "children": [
        "None"
    ],
    "control_set_name": "World War"
}
]

```

GET http://127.0.0.1:8000/controlhierarchies\_details/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  GraphQ

```

1 {
2     "name": "Biology"
3 }

```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1 {
2     "slug": "5fc4e295-010f-48c7-b47c-84809e6db419",
3     "control_set": [
4         {
5             "name": "Animal",
6             "reference_id": "AL20",
7             "control_name": "Animal",
8             "description": "description of the animal"
9         },
10        {
11            "name": "Flower",
12            "reference_id": "FL20",
13            "control_name": "Flower",
14            "description": "description of the flower"
15        },
16        {
17            "name": "Forest",
18            "reference_id": "FR20",
19            "control_name": "Forest",
20            "description": "description of the forest"
21        }
22 }

```

GET http://127.0.0.1:8000/controlhierarchies\_details/

Params Authorization Headers (9) Body Scripts Settings

none  form-data  x-www-form-urlencoded  raw  binary  Gra

```

1 {}
2 ...
3 ...

```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON ↻

```

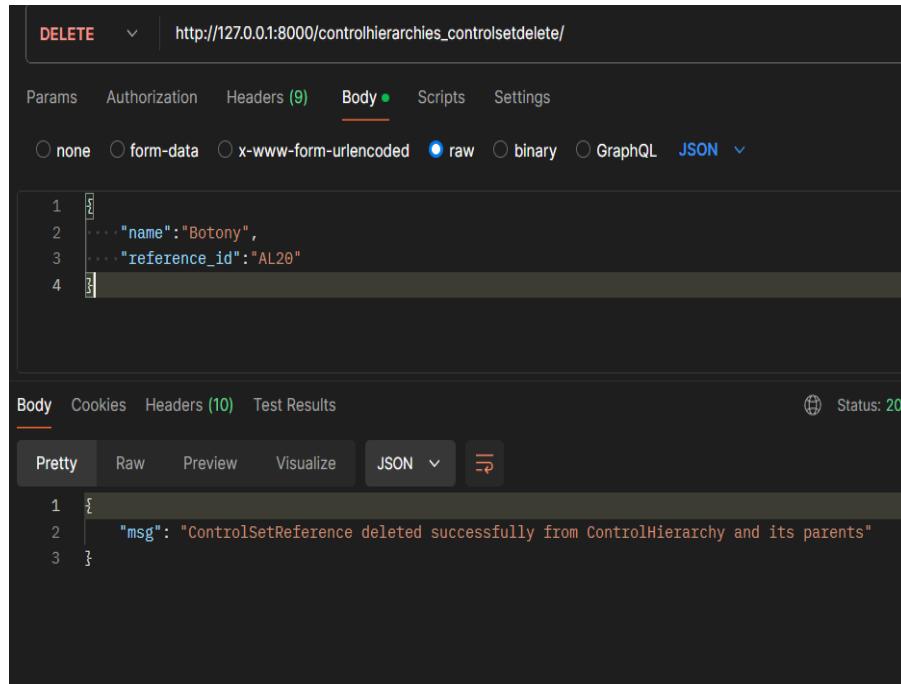
1 {
2     "msg": "No Control Set found with name Biology"
3 }

```

### 13. ControlHierarchyControlsetDeleteAPI

- Purpose: Deletes a ControlSetReference object from ControlHierarchy by its name and reference ID.
- Process: The specified ControlSetReference object is removed from the ControlHierarchy and its parents.

For calling this API use PUT [http://127.0.0.1:8000/controlhierarchies\\_controlsetdelete/](http://127.0.0.1:8000/controlhierarchies_controlsetdelete/)



DELETE [http://127.0.0.1:8000/controlhierarchies\\_controlsetdelete/](http://127.0.0.1:8000/controlhierarchies_controlsetdelete/)

Params Authorization Headers (9) Body Scripts Settings

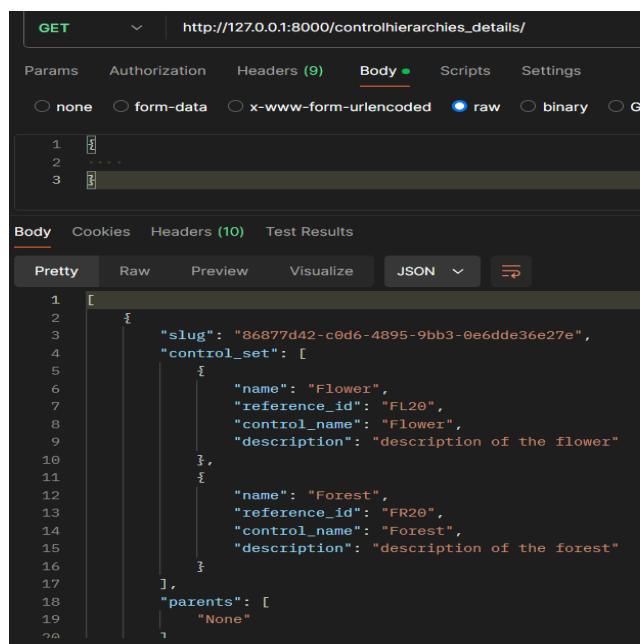
Body (raw JSON)

```
1 {
2   ...
3   "name": "Botony",
4   ...
5   "reference_id": "AL20"
6 }
```

Body Cookies Headers (10) Test Results Status: 200

Pretty Raw Preview Visualize JSON

```
1 {
2   ...
3   "msg": "ControlSetReference deleted successfully from ControlHierarchy and its parents"
4 }
```



GET [http://127.0.0.1:8000/controlhierarchies\\_details/](http://127.0.0.1:8000/controlhierarchies_details/)

Params Authorization Headers (9) Body Scripts Settings

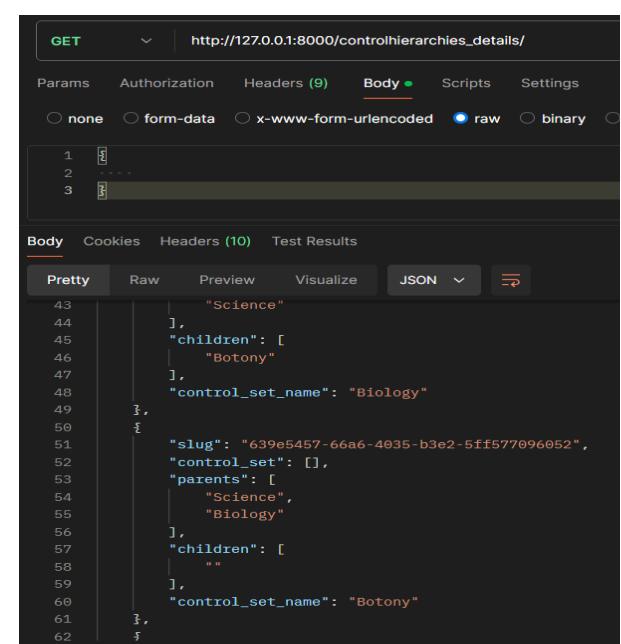
Body (raw JSON)

```
1 [
2   {
3     "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",
4     "control_set": [
5       {
6         "name": "Flower",
7         "reference_id": "FL20",
8         "control_name": "Flower",
9         "description": "description of the flower"
10      },
11      {
12        "name": "Forest",
13        "reference_id": "FR20",
14        "control_name": "Forest",
15        "description": "description of the forest"
16      }
17    ],
18    "parents": [
19      "None"
20    ]
21  }
22 ]
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "slug": "86877d42-c0d6-4895-9bb3-0e6dde36e27e",
4     "control_set": [
5       {
6         "name": "Flower",
7         "reference_id": "FL20",
8         "control_name": "Flower",
9         "description": "description of the flower"
10      },
11      {
12        "name": "Forest",
13        "reference_id": "FR20",
14        "control_name": "Forest",
15        "description": "description of the forest"
16      }
17    ],
18    "parents": [
19      "None"
20    ]
21  }
22 ]
```



GET [http://127.0.0.1:8000/controlhierarchies\\_details/](http://127.0.0.1:8000/controlhierarchies_details/)

Params Authorization Headers (9) Body Scripts Settings

Body (raw JSON)

```
1 [
2   {
3     "slug": "639e5457-66a6-4035-b3e2-5ff577096052",
4     "control_set": [],
5     "parents": [
6       "Science",
7       "Biology"
8     ],
9     "children": [
10       "Botony"
11     ],
12     "control_set_name": "Biology"
13   },
14   {
15     "slug": "639e5457-66a6-4035-b3e2-5ff577096052",
16     "control_set": [],
17     "parents": [
18       "Science",
19       "Biology"
20     ],
21     "children": [
22       ""
23     ],
24     "control_set_name": "Botony"
25   }
26 ]
```

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "slug": "639e5457-66a6-4035-b3e2-5ff577096052",
4     "control_set": [],
5     "parents": [
6       "Science",
7       "Biology"
8     ],
9     "children": [
10       "Botony"
11     ],
12     "control_set_name": "Biology"
13   },
14   {
15     "slug": "639e5457-66a6-4035-b3e2-5ff577096052",
16     "control_set": [],
17     "parents": [
18       "Science",
19       "Biology"
20     ],
21     "children": [
22       ""
23     ],
24     "control_set_name": "Botony"
25   }
26 ]
```

(Animal is deleted from botany and its parents also)

