

Python Test Assignment

Categories API

Introduction

Create a simple Categories API that stores category tree to database and returns category parents, children and siblings by category id.

Requirements

Use Python 3.4+ and Django Framework (or Django Rest Framework).

Use of any other third-party libraries or Django extensions (mptt, treebread, etc) is prohibited.

Categories Endpoint

Create **POST /categories/** API endpoint. Endpoint should accept **json** body (see example **Request**), validate input data (see Request) and save categories to database (category name should be unique).

Example.

Request:

```
{
  "name": "Category 1",
  "children": [
    {
      "name": "Category 1.1",
      "children": [
        {
          "name": "Category 1.1.1",
          "children": [
            {
              "name": "Category 1.1.1.1"
            },
            {
              "name": "Category 1.1.1.2"
            },
            {
              "name": "Category 1.1.1.3"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
},
{
  "name": "Category 1.1.2",
  "children": [
    {
      "name": "Category 1.1.2.1"
    },
    {
      "name": "Category 1.1.2.2"
    },
    {
      "name": "Category 1.1.2.3"
    }
  ]
}
]
},
{
  "name": "Category 1.2",
  "children": [
    {
      "name": "Category 1.2.1"
    },
    {
      "name": "Category 1.2.2",
      "children": [
        {
          "name": "Category 1.2.2.1"
        },
        {
          "name": "Category 1.2.2.2"
        }
      ]
    }
  ]
}
]
}
]
}

```

Category Endpoint

Create **GET /categories/<id>/** API endpoint. Endpoint should retrieve category name, parents (and their parents), children and siblings (see examples) by primary key (<id>) in json format.

Example 1.

GET /categories/2/

Response:

```
{
  "id": 2,
  "name": "Category 1.1",
  "parents": [
    {
      "id": 1,
      "name": "Category 1"
    }
  ],
  "children": [
    {
      "id": 3,
      "name": "Category 1.1.1"
    },
    {
      "id": 7,
      "name": "Category 1.1.2"
    }
  ],
  "siblings": [ // sisters and brothers
    {
      "id": 11,
      "name": "Category 1.2"
    }
  ]
}
```

Example 2.

GET /categories/8/

Response:

```
{
  "id": 8,
  "name": "Category 1.1.2.1",
  "parents": [
    {
      "id": 7,
      "name": "Category 1.1.2"
    }
  ],
}
```

```
{
  "id": 2,
  "name": "Category 1.1"
},
{
  "id": 1,
  "name": "Category 1"
},
],
"children": [],
"siblings": [
  {
    "id": 9,
    "name": "Category 1.1.2.2"
  },
  {
    "id": 10,
    "name": "Category 1.1.2.3"
  }
]
}
```