**Energy network manager application**

Software that allows execute CRUD operations on hierarchical structure which represents Simple Network Model.

The goal is implement software that has API for Network creation and operation with it.

1. Implement Network model in Java (all information provided in Network model description)

2. Implement API that allows:

- Create whole network in one step

- Search for Networks

- Search for Network entries (children)

- Delete network

- Update whole Network

- Create network step by step (ability to add child to target entry eg to Network/Substation/Transformer/Feeder)

- Update entries (ability to update child of target entry eg: Substation/Transformer/Feeder/Resource)

- Delete entries (ability to remove child of target entry eg: from Network/Substation/Transformer/Feeder)

- Verifying that Network in consistent state (Hierarchy is valid, no duplication of ids)

3. We need to log all operations. In simple case use System out or configure and use any logger

4. \* Do persist network, this means after program restart the results will be restored

5. \* Web application with RESTful web services that allow us to manage network (create, list all, search by id, update and delete)

6. \* Documentation of web services. Any of suitable explanation of REST Api usage

\* Optional/advanced tasks

***Requirments***:

Software have to be developed on Java 11+

Application need to be build via Maven

***Result***: packed artifact - runnable jar/war

**Network model description**

Network is hierarchical structure (graph/tree), where each node has set of defined properties and strictly defined type

All nodes beside RESOURCE type can have children

All nodes have such properties:

id: unique identifier

type: defined node type, may be one of (NETWORK, SUBSTATION, TRANSFORMER, FEEDER, RESOURCE)

name: user would be able to define name for each node

description: user would be able to add description for each node

params: user would be able to define additional parameters for each node. It is Map/JSON

*Nodes has next hierarchy:*

NETWORK -> SUBSTATION -> TRANSFORMER -> FEEDER -> RESOURCE

Exmple in JSON format

{

"id": "nt-1",

"type": "NETWORK",

"name": "CE-network",

"description": "Central Europe network",

"params": {

"": ""

},

"children": [

{

"id": "sbst-1",

"type": "SUBSTATION",

"name": "Kyiv\_sub",

"description": "Kyiv substation",

"params": {

"lon": 50.45,

"lat": 30.52

},

"children": [

{

"id": "tf-1",

"type": "TRANSFORMER",

"name": "Dnipro-tf",

"description": "Dnipro transformer",

"params": {},

"children": [

{

"id": "fdr-1",

"type": "FEEDER",

"name": "Long feeder",

"description": "Feeder that belong to Dnipro-tf",

"params": {},

"children": [

{

"id": "rsc-1",

"type": "RESOURCE",

"name": "Big-consumer",

"description": "Load that belong to fdr-1",

"params": {

"consumes": 20,

"units": "MWatt"

}

},

{

"id": "rsc-2",

"type": "RESOURCE",

"name": "SP-saturn",

"description": "Solar panels, 10MWatt",

"params": {}

}

]

},

{

"id": "fdr-2",

"type": "FEEDER",

"name": "New feeder",

"description": "Feeder that belong to Dnipro-tf",

"params": {},

"children": [

{

"id": "rsc-10",

"type": "RESOURCE",

"name": "Consumer",

"description": "Load that belong to fdr-1",

"params": {

"consumes": 10,

"units": "MWatt"

}

}

]

}

]

}

]

}

]

}