Implement storage array simulator as a web service by python. It takes Python Django framework to implement REST API of LUN management.

For LUN, there are five attributes:

Id: Integer

Name: string

Initiator: string

Target: string

Size: string

Main functions:

1. Retrieve the information of all LUNs
2. Retrieve the information of one specific LUN
3. Retrieve the size of one specific LUN
4. Create one LUN
5. Create multiple LUNs
6. Resize a LUN
7. Remove a LUN

* Use threading module to handle the concurrent requests
* Use lun.db file to implement Data persistence
* Error Handle:

1. User can only modify the size of the existing LUN, otherwise, it will be handled by status code:400 error “Bad request”
2. If user operate on the non-existing LUN, status code:404 error: “the resource is not found” error will prompt
3. If URL doesn’t match the pattern in urls.py, status code:404 error: “the resource is not found” error will prompt

For each fucntion, here is the detail message:

#### URL: / or /luns/

Description: Retrieve the information of all LUNs

## Method: GET

## Parameters: N/A

## Example: *curl <http://127.0.0.1:8000/>*

## *curl <http://127.0.0.1:8000/luns/>*

## Responses:

### *"luns": [*

### *{*

### *"id": 1,*

### *"initiator": "hostname-ig",*

### *"target": "hostname-tg",*

### *"name": "test1",*

### *"size": “22TB”*

### *},*

### *{*

### *"id": 2,*

### *"initiator": "default",*

### *"target": "default",*

### *"name": "test2",*

### *"size": “100GB”*

### *}*

### *]*

### *"status\_code": 200*

#### URL: / luns/id

Description: Retrieve the information of one specific LUN

## Method: GET

## Parameters: N/A

### Example: <http://127.0.0.1:8000/luns/1/>

### Responses:

### *{*

### *"status\_code": 200,*

### *"lun": {*

### *"id": 1,*

### *"initiator": "hostname-ig",*

### *"target": "hostname-tg",*

### *"name": "test1",*

### *"size": “22TB”*

### *}*

### *"status\_code": 200*

#### URL: / luns/id/size/

Description: Retrieve the size of one specific LUN

## Method: GET

## Parameters: id

### Example: *curl http://127.0.0.1:8000/luns/2/size/*

### Responses:

### {

### "status\_code": 200,

### "lun": “100GB” }

### "status\_code": 200

#### URL: /luns/

Description : Create one LUN

## Method: PUT

## Parameters: [{"name”: string,”size”: string}]

### Example:

### *curl -H "Content-Type:application/json" -X POST -d '[{"name":"test3","size":"100GB"}]' http://127.0.0.1:8000/luns/*

### Responses:

### "luns": [

### {

### "initiator": "default",

### "size": "100GB",

### "target": "default",

### "id": 3,

### "name": "test3"

### }

### ],

### "status\_code": 200

#### URL: /luns/

Description : Create multiple LUNs

## Method: PUT

## Parameters:[{"name”: string, “size”: string}, {"name”: string, “size”: string}]

Example:

*curl -H "Content-Type:application/json" -X POST -d '[{"name":"test5","size":700MB},{"name":"test6","size":100TB}]' http://127.0.0.1:8000/luns/*

### Responses:

### *"luns": [*

### *{*

### *"initiator": "default",*

### *"size": "700MB",*

### *"target": "default",*

### *"id": 4,*

### *"name": "test5"*

### *},*

### *{*

### *"initiator": "default",*

### *"size": "100TB",*

### *"target": "default",*

### *"id": 5,*

### *"name": "test6"*

### *}*

### *],*

### *"status\_code": 200*

#### URL: /luns/id/

Description: Resize a LUN

## Method: POST

## Parameters: '{"size":string}'

### Example:

### *curl -H "Content-Type:application/json" -X PUT -d '{"size":"25GB"}' http://127.0.0.1:8000/luns/5/*

### Responses:

### "luns": [

### {

### "initiator": "default",

### "size": "25GB",

### "target": "default",

### "id":5,

### "name": "test5"

### }

### ],

### "status\_code": 200

#### URL: /luns/id/

Description: Remove a LUN

## Method: DELETE

## Parameters: N/A

### Example:

### *curl X DELETE -d http://127.0.0.1:8000/luns/6/*

### Responses:

### "status\_code": 200