**Synectiks Portal API Documentation**.

[1. Users 3](#_Toc14135)

[1.1. Create 3](#_Toc11627)

[1.2. Update 3](#_Toc5141)

[1.3. Delete 4](#_Toc5237)

[1.4. DeleteById 4](#_Toc1455)

[1.5. listAll 4](#_Toc27814)

[1.6. findById 5](#_Toc4797)

[2. Service 5](#_Toc11950)

[2.1. Create 5](#_Toc4778)

[2.2. Update 6](#_Toc22524)

[2.3. Delete 6](#_Toc10792)

[2.4. DeleteById 6](#_Toc11116)

[2.5. listAll 7](#_Toc26032)

[2.6. findById 7](#_Toc19556)

[3. Subscription 7](#_Toc10427)

[3.1. Create 7](#_Toc12177)

[3.2. Update 8](#_Toc8173)

[3.3. Delete 8](#_Toc30447)

[3.4. DeleteById 8](#_Toc5527)

[3.5. listAll 9](#_Toc15046)

[3.6. findById 9](#_Toc17202)

[4. StateMachine 9](#_Toc4022)

[4.1. Create 9](#_Toc20417)

[4.2. Update 10](#_Toc23057)

[4.3. Delete 11](#_Toc27197)

[4.4. DeleteById 11](#_Toc16656)

[4.5. listAll 11](#_Toc10367)

[4.6. findById 12](#_Toc26793)

[4.7. currentState 12](#_Toc22074)

[4.8. listStates 12](#_Toc3849)

[Response: StateName, StateName, … 13](#_Toc32005)

[4.9. sendEvent 13](#_Toc21257)

[5. ElasticSearch 13](#_Toc30090)

[5.1. elsQuery 13](#_Toc21264)

[5.2. query 13](#_Toc4944)

[5.3. list 14](#_Toc4163)

[5.4. count 15](#_Toc18370)

[5.5. aggregateCounts 15](#_Toc8176)

[6. JackrabbitOak 16](#_Toc15955)

[6.1. Create 16](#_Toc17815)

[6.2. Update 16](#_Toc17143)

[6.3. Delete 16](#_Toc8706)

[6.4. DeleteById 17](#_Toc3409)

[6.5. listAll 17](#_Toc18500)

[6.6. findById 17](#_Toc7737)

Services **error response code is** 417 with exception details json object in error stream.

## Users

### 1.1. Create

Api to create a new entity

**Url:** /api/v1/users/create

**Method:** POST

**Request:**

Body: {

"type": "CUSTOMER | USER | SUPER",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

**Response:** {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

### 1.2. Update

Api to update existing entity

**Url:** /api/v1/users/update

**Method:** POST

**Request:**

Body: {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

**Response:** {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

### 1.3. Delete

Api to delete existing entity

**Url:** /api/v1/users/delete

**Method:** POST

**Request:**

Body: {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

**Response:** User removed Successfully

### 1.4. DeleteById

Api to delete existing entity by id

**Url:** /api/v1/users/delete/[id]

**Method:** POST

**Request:**

**Response:** User removed Successfully

### 1.5. listAll

Api to list all entity

**Url:** /api/v1/users/listAll

**Method:** POST | GET

**Request:**

**Response:** [{

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}, {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}]

### 1.6. findById

Api to find an entity by id

**Url:** /api/v1/users/[id]

**Method:** POST

**Request:**

**Response:** {

"id": "",

"type": "",

"**username**": "",

"**password**": "",

"**email**": "",

"roles": [{"role": ""}]

}

## Service

### 2.1. Create

Api to create a new entity

**Url:** /api/v1/service/create

**Method:** POST

**Request:**

Body: {

"**name**": "",

"**category**": "SERVICE | RESOURCE",

"description": "",

"SLA": ""

}

**Response:** {

"id": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

### 2.2. Update

Api to update existing entity

**Url:** /api/v1/service/update

**Method:** POST

**Request:**

Body: {

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

**Response:** {

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

### 2.3. Delete

Api to delete existing entity

**Url:** /api/v1/service/delete

**Method:** POST

**Request:**

Body: {

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

**Response:** Service Deleted Successfully

### 2.4. DeleteById

Api to delete existing entity by id

**Url:** /api/v1/service/delete/[id]

**Method:** POST

**Request:**

**Response:** {

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

### 2.5. listAll

Api to list all entity

**Url:** /api/v1/service/listAll

**Method:** POST | GET

**Request:**

**Response:** [{

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}, … ]

### 2.6. findById

Api to find an entity by id

**Url:** /api/v1/service/[id]

**Method:** POST

**Request:**

**Response:** {

"**id**": "",

"**name**": "",

"**category**": "",

"description": "",

"SLA": ""

}

## Subscription

### 3.1. Create

Api to create a new entity

**Url:** /api/v1/subscription/

**Method:** POST

**Request:**

Body: {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

**Response:** {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

### 3.2. Update

Api to update existing entity

**Url:** /api/v1/subscription/update

**Method:** POST

**Request:**

Body: {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

**Response:** {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

### 3.3. Delete

Api to delete existing entity

**Url:** /api/v1/subscription/delete

**Method:** POST

**Request:**

Body: {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

**Response:** Sucscription removed successfully

### 3.4. DeleteById

Api to delete existing entity by id

**Url:** /api/v1/subscription/delete/[id]

**Method:** POST

**Request:**

**Response:** Subscription removed successfully

### 3.5. listAll

Api to list all entity

**Url:** /api/v1/subscription/listAll

**Method:** POST | GET

**Request:**

**Response:** [{

"id": "",

"**serviceId**": "",

"**customerId**": ""

}, … ]

### 3.6. findById

Api to find an entity by id

**Url:** /api/v1/subscription/[id]

**Method:** POST

**Request:**

**Response:** {

"id": "",

"**serviceId**": "",

"**customerId**": ""

}

## StateMachine

### 4.1. Create

Api to create a new entity

**Url:** /api/v1/states/create

**Method:** POST

**Request:**

Body: {

"**name**": "",

"**parent**": "",

"**target**": "",

"**event**": "",

"**action**": "",

"guard": "",

"guardExpression": "",

"initial": true|false,

"end": true|false

}

**Response:** {

"id": "",

"name": "",

"parent": "",

"target": "",

"event": "",

"action": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}

### 4.2. Update

Api to update existing entity

**Url:** /api/v1/states/update

**Method:** POST

**Request:**

Body: {

"**id**": "",

"**name**": "",

"**parent**": "",

"**target**": "",

"**event**": "",

"**action**": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}

**Response:** {

"id": "",

"name": "",

"parent": "",

"target": "",

"event": "",

"action": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}

### 4.3. Delete

Api to delete existing entity

**Url:** /api/v1/states/delete

**Method:** POST

**Request:**

Body: {

"**id**": "",

"name": "",

"parent": "",

"target": "",

"event": "",

"action": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}

**Response:** State removed successfully.

### 4.4. DeleteById

Api to delete existing entity by id

**Url:** /api/v1/states/delete/[id]

**Method:** POST

**Request:**

**Response:** State removed successfully

### 4.5. listAll

Api to list all entity

**Url:** /api/v1/states/listAll

**Method:** POST | GET

**Request:**

**Response:** [{

"id": "",

"name": "",

"parent": "",

"target": "",

"event": "",

"action": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}, … ]

### 4.6. findById

Api to find an entity by id

**Url:** /api/v1/states/[id]

**Method:** POST

**Request:**

**Response:** {

"id": "",

"name": "",

"parent": "",

"target": "",

"event": "",

"action": "",

"guard": "",

"guardExpression": "",

"initial": ,

"end":

}

### 4.7. currentState

Api to get current state name from state machine

**Url:** /api/v1/states/currentState

**Method:** POST

**Request:**

**Parameter** orderID = “”

**Response:** StateName

### 4.8. listStates

List all states of state machine by id

**Url:** /api/v1/states/listStates

**Method:** POST

**Request:**

**Parameter** orderID = “”

### **Response:** StateName, StateName, …

### 4.9. sendEvent

Api to send an event to change statemachine state and perform state action

**Url:** /api/v1/states/sendEvent

**Method:** POST

**Request:**

**Parameter** orderID = “”

**Response:** “true” for successfully action and state changes, false otherwise

## ElasticSearch

### 5.1. elsQuery

Api to perform elastic search queries directly on server, avoid to use it as API, use it for internal access.

query = “String needs to be search”

pageNo = “page number 1 based indexed i.e. 1 for first and so on”

pageSize = “The size of results list in one set keep it short for better performance”

Elastic query API returns 10 results at a time by default.

<https://www.elastic.co/guide/en/elasticsearch/reference/current/_introducing_the_query_language.html>

Link to follow and get idea about string query of elastic search

**Url:** /api/v1/search/elsQuery

**Method:** POST

**Request Parameter:**

**query** =

pageNo =

pageSize =

**Response:** [org](eclipse-javadoc:%E2%98%82=mysynectiksportal/C:\\/Users\\/Rajesh\\/.m2\\/repository\\/org\\/elasticsearch\\/elasticsearch\\/2.4.5\\/elasticsearch-2.4.5.jar<org).[elasticsearch](eclipse-javadoc:%E2%98%82=mysynectiksportal/C:\\/Users\\/Rajesh\\/.m2\\/repository\\/org\\/elasticsearch\\/elasticsearch\\/2.4.5\\/elasticsearch-2.4.5.jar<org.elasticsearch).[action](eclipse-javadoc:%E2%98%82=mysynectiksportal/C:\\/Users\\/Rajesh\\/.m2\\/repository\\/org\\/elasticsearch\\/elasticsearch\\/2.4.5\\/elasticsearch-2.4.5.jar<org.elasticsearch.action).[search](eclipse-javadoc:%E2%98%82=mysynectiksportal/C:\\/Users\\/Rajesh\\/.m2\\/repository\\/org\\/elasticsearch\\/elasticsearch\\/2.4.5\\/elasticsearch-2.4.5.jar<org.elasticsearch.action.search).SearchResponse

### 5.2. query

Api to search a string into elastic server indexes.

q = “String which needs to be searched”

cls = “fully qualified class name which object are requested”

fields = “index field names (comma separated) where q must be searched”

**Url:** /api/v1/search/query

**Method:** POST

**Request Parameters:**

**q** =

cls =

fields =

pageNo =

pageSize =

**Response:** List of JSONObject or cls class objects

### 5.3. list

Api to list the entity objects which match query params.

filters = “its a json query object in following format for searching entities in specified class type objects.”

{

"and": [{

"field-name-1": "value-1",

...

}],

"or": [{

"field-name-1": "value-1",

"field-name-2": "value-2",

...

}],

"not": [{

"field-name-1": "value-1",

...

}],

"filters": [{

"field-name-1": "value-1",

"ranges": [{

"**type**": "[Number|Date|String]",

"**fieldName**": "field-name",

"**from**": "start from value",

"**to**": "end to value",

"locale": "local-value if any i.e **en\_IN** for india",

"format": "value format if date string is not in format

**yyyy-MM-dd'T'HH:mm:ss.SSSz**"

}]

}]

}

**Url:** /api/v1/search/list

**Method:** POST

**Request Parameter:**

**cls** =

filters =

pageNo =

pageSize =

**Response:** List of cls class objects

### 5.4. count

Api to get entities count vased on filter search like above.

**Url:** /api/v1/search/count

**Method:** POST

**Request Parameter:**

**cls** =

filters =

**Response:** count of entities

### 5.5. aggregateCounts

Api to run aggregate queries to get counts

aggregator = a json object in following format with filters and class object like above.

{

**"aggreType"**: "count | terms | avg | ranges | min | max | sum",

**"fieldType"**: "field-type",

**"fieldName"**: "field-name",

"interval": "1d" -- "[\d+][s|m|h|d|w|M|q|y]"

"ranges": [{"from": X, "to": Y}], -- only used for numbers

"values": ["ABC", "DEF"],

"locale": "local-value if any",

"format": "value format if needs formated key"

}

**Url:** /api/v1/search/aggregateCounts

**Method:** POST

**Request Parameters:**

cls =

filters =

Aggregator = {}

**Response:** Returns a json map object with aggregate key and counts

## JackrabbitOak

### 6.1. list

Api to get the list of child nodes by absolute node path.

**Url:** /api/v1/oakRepo/list

**Method:** POST

**Request Parameter:**

path =

**Response:** string of child node objects

### 6.2. createNode

Api to create a new entity in jackrabbit repository at specified path. if path exists then node get created as child with new random id.

**Parameters:**

**parentPath** node path or node parent path to create node at

**json** json string with node properties to create new node.

**cls** specify fully qualified class name for json to type cast in

**nodeName** optional specify if don't want random id as node name

**Returns:**

json of newly created object in repository.

**Url:** /api/v1/oakRepo/createNode

**Method:** POST

**Request Parameters:**

parentPath =

Json =

cls =

nodeName =

**Response:** json of newly created object in repository.

### 6.3. moveNode

Api to move or rename node or tree from one to another path. Make sure both path must exists

**Parameters:**

**src** source node path

**dest** destination node path

**Returns:**

moved node object

**Url:** /api/v1/oakRepo/moveNode

**Method:** POST

**Request Parameters:**

src =

dest =

**Response:** moved node json object

### 6.4. similar

Api to find similar nodes as input node path

**Url:** /api/v1/oakRepo/similar

**Method:** POST

**Request Parameters:**

absPath =

**Response:** list of json objects format.

### 6.5. suggestions

Api to get suggestions for input string.

**Url:** /api/v1/oakRepo/suggestions

**Method:** POST | GET

**Request Parameters:**

input =

**Response:** List of suggestion as json object.

### 6.6. spellcheck

Api to find correct spell for input string.

**Url:** /api/v1/oakRepo/spellcheck

**Method:** POST

**Request Parameters:**

**input =**

**Response:** List of correct spells

### 6.7. search

Api to search input query in jackrabbit repository.

**Parameters:**

**path** absolute node path to search in repository.

**cols** fields to be searched for query

**query** string to search

**orderBy** comma separate list of columns for result sorting.

**Returns:**

json List object of search result nodes as string

**Url:** /api/v1/oakRepo/search

**Method:** POST

**Request parameters:**

**path =**

**cols =**

**query =**

**orderBy =**

**Response:** json string list of search result nodes.