**API DOCUMENTATION:**

**USER:**

1. **Method:** POST

**End Point:** <http://localhost:8080/user/register>

**Description:** Registers the user if user doesn’t already exist.

**Input:** JSON.

**Output:** String.

**Input Format:**

{

“userName”:”String”,

“password”:”String”

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“userName”:”SomeOne”,

“password”:”xyz”

}

**Output:**

**Case 1:** “User Created Successfully”

**Case 2:** “User Already Exist”

**Explanation:**

**Case 1:** If the username is new then returns the string “User Created Successfully“ and changes the Http status code to “201 Created”.

**Case 2:** If the username is already present returns the string “User Already Exist” and the Http status code remains “200 Ok”.

1. **Method:** POST

**End Point:** <http://localhost:8080/user/login>

**Description:** User login by providing valid credentials.

**Input:** JSON.

**Output:** JSON.

**Input Format:**

{

“userName”:”String”,

“password”:”String”

}

**Output Format:**

{

“userId”:Integer,

“userName”:”String”,

“status”:”String”

}

**Example:**

**Input:**

{

“userName”:”SomeOne”,

“password”:”xyz”

}

**Output:**

**Case 1:**

{

“userId”:1,

“userName”:”SomeOne”,

“status”:”Success”

}

**Case 2:**

{

“userId”:null,

“userName”:null,

“status”:”Invalid credentials”

}

**Explanation:**

**Case 1:**

If the given credentials are correct and the http status code changes to “202 Accepted”

**Case 2:**

If the given credentials are invalid and the http status code changes to “404 Not Found”

**GROUP:**

1. **Method:** GET

**End Point:** [http://localhost:8080/group/{userId}](http://localhost:8080/group/%7buserId%7d)

**Description:** Fetches all the groups corresponding to the given userId.

**Input:** Integer.

**Output:** JSON List.

**Input Format:**

userId appended with the url.

**Output Format:**

[

{

"groupId": Integer,

"groupName": "String"

},

{

"groupId": Integer,

"groupName": "String"

},

.

.

.

{

"groupId": Integer,

"groupName": "String"

}

]

**Example:**

**Input:**

<http://localhost:8080/group/1>

**Output:**

**Case 1:**

[

{

"groupId": 1,

"groupName": "Group 1"

},

{

"groupId": 2,

"groupName": "Group 2"

},

]

**Case 2:**

[]

**Explanation:**

**Case 1:**

Displays all the groups of the provided userId and changes the http status code to “202 Accepted”.

**Case 2:**

The provided userId doesn’t have any groups and changes the http status code to “404 Not Found”.

1. **Method:** POST

**End Point:** <http://localhost:8080/group>

**Description:** Creates the group with the given name for the given userId.

**Input:** JSON

**Output:** String

**Input Format:**

{

“userId”:Integer,

“groupName”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“userId”:1,

“groupName”:”Something”

}

**Output:**

**Case 1:** “Group Created Successfully”

**Case 2:** “Error Creating Group”

**Explanation:**

**Case 1:** Group is created with the given group name if the userId is valid and the http status code is changed to “201 Created”.

**Case 2:** Given userId is not valid so group is not created and the http status code is changed to “404 Not Found”.

1. **Method:** PUT

**End Point:** <http://localhost:8080/group>

**Description:** Updates the group name provided the groupId is valid.

**Input:** JSON.

**Output:** String.

**Input Format:**

{

“groupId”:Integer,

“groupName”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“groupId”:1,

“groupName”:”Something”

}

**Output:**

**Case 1**: Group Updated Successfully.

**Case 2:** Error Updating Group.

**Explanation:**

**Case 1:** Group is updated with the given group name if the groupId is valid and the http status code is changed to “202 Accepted”.

**Case 2:** Given groupId is not valid so the group name is not updated and the http status code is changed to “404 Not Found”.

1. **Method:** DELETE

**End Point:** [http://localhost:8080/group/{groupId}](http://localhost:8080/group/%7bgroupId%7d)

**Description:** Deletes the group provided the groupId is valid.

**Input:** Integer

**Output:** String.

**Input Format:**

groupId appended with the url.

**Output Format:**

“String”.

**Example:**

**Input:**

<http://localhost:8080/group/1>

**Output:**

**Case 1:** “Group Deleted Successfully”.

**Case 2:** “Error Deleting Group”.

**Explanation:**

**Case 1:** Group is deleted if the groupId is valid and the http status code is changed to “204 No Content”.

**Case 2:** Given groupId is not valid so the group is not deleted and the http status code is changed to “404 Not Found”.

1. **Method:** GET

**End Point**: [http://localhost:8080/group/search/{search}](http://localhost:8080/group/search/%7bsearch%7d)

**Description:** Searches the string appended with the url and gives the list of group id and group name which contains the search string.

**Input:** String.

**Output**: JSON List.

**Input Format:**

The string to be searched should be appended with url.

**Output Format:**

[

{

“groupId”: Integer,

“groupName”: String

},

.

.

.

{

“groupId”: Integer,

“groupName”: String

}

]

**Example:**

**Input:**

<http://localhost:8080/group/search/something>

**Output:**

**Case 1:**

[

{

“groupId”:1,

“groupName”:”Something”

},

{

“groupId”:2,

“groupName”:”Some”

}

]

**Case 2:**

[]

**Explanation:**

**Case 1:** Gives the list of group id and group name in which group name contains the search string.

**Case 2:** No group name contains the given search string.

**TITLE**:

1. **Method:** GET

**End Point:** [http://localhost:8080/title/{groupId}](http://localhost:8080/title/%7bgroupId%7d)

**Description:** Fetches all the titles corresponding to the given groupId.

**Input:** Integer.

**Output:** JSON List.

**Input Format:**

groupId appended with the url.

**Output Format:**

[

{

"titleId": Integer,

"titleName": "String"

},

{

"titleId": Integer,

"titleName": "String"

},

.

.

.

{

"titleId": Integer,

"titleName": "String"

}

]

**Example:**

**Input:**

<http://localhost:8080/title/1>

**Output:**

**Case 1:**

[

{

"titleId": 1,

"titleName": "Title 1"

},

{

"titleId": 2,

"titleName": "Title 2"

},

]

**Case 2:**

[]

**Explanation:**

**Case 1:**

Displays all the titles of the provided groupId and changes the http status code to “202 Accepted”.

**Case 2:**

The provided groupId doesn’t have any groups and changes the http status code to “404 Not Found”.

1. **Method:** POST

**End Point:** <http://localhost:8080/title>

**Description:** Creates the title with the given name for the given groupId.

**Input:** JSON

**Output:** String

**Input Format:**

{

“groupId”:Integer,

“titleName”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“groupId”:1,

“titleName”:”Something”

}

**Output:**

**Case 1:** “Title Created Successfully”

**Case 2:** “Error Creating Title”

**Explanation:**

**Case 1:** Title is created with the given title name if the groupId is valid and the http status code is changed to “201 Created”.

**Case 2:** Given groupId is not valid so title is not created and the http status code is changed to “404 Not Found”.

1. **Method:** PUT

**End Point:** <http://localhost:8080/title>

**Description:** Updates the title name provided the titleId is valid.

**Input:** JSON.

**Output:** String.

**Input Format:**

{

“titleId”:Integer,

“titleName”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“titleId”:1,

“titleName”:”Something”

}

**Output:**

**Case 1**: Title Updated Successfully.

**Case 2:** Error Updating Title.

**Explanation:**

**Case 1:** Title is updated with the given title name if the titleId is valid and the http status code is changed to “202 Accepted”.

**Case 2:** Given titleId is not valid so the group name is not updated and the http status code is changed to “404 Not Found”.

1. **Method:** DELETE

**End Point:** [http://localhost:8080/title/{titleId}](http://localhost:8080/title/%7btitleId%7d)

**Description:** Deletes the title provided the titleId is valid.

**Input:** Integer

**Output:** String.

**Input Format:**

titleId appended with the url.

**Output Format:**

“String”.

**Example:**

**Input:**

<http://localhost:8080/title/1>

**Output:**

**Case 1:** “Title Deleted Successfully”.

**Case 2:** “Error Deleting Title”.

**Explanation:**

**Case 1:** Title is deleted if the titleId is valid and the http status code is changed to “204 No Content”.

**Case 2:** Given titleId is not valid so the group is not deleted and the http status code is changed to “404 Not Found”.

1. **Method:** GET

**End Point**: [http://localhost:8080/title/search/{search}](http://localhost:8080/title/search/%7bsearch%7d)

**Description:** Searches the string appended with the url and gives the list of title id and group name which contains the search string.

**Input:** String.

**Output**: JSON List.

**Input Format:**

The string to be searched should be appended with url.

**Output Format:**

[

{

“titleId”: Integer,

“titleName”: String

},

.

.

.

{

“titleId”: Integer,

“titleName”: String

}

]

**Example:**

**Input:**

<http://localhost:8080/title/search/something>

**Output:**

**Case 1:**

[

{

“titleId”:1,

“titleName”:”Something”

},

{

“titleId”:2,

“titleName”:”Some”

}

]

**Case 2:**

[]

**Explanation:**

**Case 1:** Gives the list of title id and title name in which title name contains the search string.

**Case 2:** No title name contains the given search string.

**NOTE:**

1. **Method:** GET

**End Point:** [http://localhost:8080/note/{titleId}](http://localhost:8080/note/%7btitleId%7d)

**Description:** Fetches the note corresponding to the given titleId.

**Input:** Integer.

**Output:** JSON.

**Input Format:**

titleId appended with the url.

**Output Format:**

[

{

"noteId": Integer,

"message": "String"

}

]

**Example:**

**Input:**

<http://localhost:8080/note/1>

**Output:**

**Case 1:**

[

{

"noteId": 1,

"message": "Title 1"

}

]

**Case 2:**

[]

**Explanation:**

**Case 1:**

Displays the note of the provided titleId and changes the http status code to “202 Accepted”.

**Case 2:**

The provided titleId doesn’t have any note and changes the http status code to “404 Not Found”.

1. **Method:** POST

**End Point:** <http://localhost:8080/note>

**Description:** Creates the note with the given name for the given titleId.

**Input:** JSON

**Output:** String

**Input Format:**

{

“titleId”:Integer,

“message”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“titleId”:1,

“message”:”Something”

}

**Output:**

**Case 1:** “Note Created Successfully”

**Case 2:** “Error Creating Note”

**Explanation:**

**Case 1:** Note is created if the titleId is valid and the http status code is changed to “201 Created”.

**Case 2:** Given titleId is not valid so note is not created and the http status code is changed to “404 Not Found”.

1. **Method:** PUT

**End Point:** <http://localhost:8080/note>

**Description:** Updates the note provided the noteId is valid.

**Input:** JSON.

**Output:** String.

**Input Format:**

{

“noteId”:Integer,

“noteName”:String

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“noteId”:1,

“noteName”:”Something”

}

**Output:**

**Case 1**: Note Updated Successfully.

**Case 2:** Note Not Found, Error Updating Note.

**Explanation:**

**Case 1:** Note is updated with the given message if the noteId is valid and the http status code is changed to “202 Accepted”.

**Case 2:** Given noteId is not valid so the note is not updated and the http status code is changed to “404 Not Found”.

**CHECK LIST:**

1. **Method:** GET

**End Point:** [http://localhost:8080/checklist/{titleId}](http://localhost:8080/checklist/%7btitleId%7d)

**Description:** Fetches all the check lists corresponding to the given titleId.

**Input:** Integer.

**Output:** JSON List.

**Input Format:**

titleId appended with the url.

**Output Format:**

[

{

"listId": Integer,

"message": "String",

“checkbox”: 0 or 1

}

.

.

.

{

“listId”: Integer,

“message”: ”String”,

“checkbox”:0 or 1

}

]

**Example:**

**Input:**

<http://localhost:8080/checklist/1>

**Output:**

**Case 1:**

[

{

"listId": 1,

"listName": "Title 1"

“checkbox”: 0

},

{

“listId”:2,

“listName”: “Title 2”,

“checkbox”: 1

}

]

**Case 2:**

[]

**Explanation:**

**Case 1:**

Displays all the check lists of the provided titleId and changes the http status code to “202 Accepted”.

**Case 2:**

The provided titleId doesn’t have any check list and changes the http status code to “404 Not Found”.

1. **Method:** POST

**End Point:** <http://localhost:8080/checklist>

**Description:** Creates the checklist with the given name for the given titleId.

**Input:** JSON

**Output:** String

**Input Format:**

{

“titleId”:Integer,

“listName”:String,

“checkbox”: 0 or 1

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“titleId”:1,

“listName”:”Something”

“checkbox: 0

}

**Output:**

**Case 1:** “Check List Created Successfully”

**Case 2:** “Error Creating Check List”

**Explanation:**

**Case 1:** Check List is created if the titleId is valid and the http status code is changed to “201 Created”.

**Case 2:** Given titleId is not valid so Check List is not created and the http status code is changed to “404 Not Found”.

1. **Method:** PUT

**End Point:** <http://localhost:8080/checklist>

**Description:** Updates the checklist provided the noteId is valid.

**Input:** JSON.

**Output:** String.

**Input Format:**

{

“noteId”:Integer,

“listName”:String,

“checkbox”:0 or 1

}

**Output Format:**

“String”

**Example:**

**Input:**

{

“noteId”:1,

“listName”:”Something”,

“checkbox”:1

}

**Output:**

**Case 1**: Check List Updated Successfully.

**Case 2:** Check List Not Found, Error Updating Check List.

**Explanation:**

**Case 1:** Check List is updated if the noteId is valid and the http status code is changed to “202 Accepted”.

**Case 2:** Given noteId is not valid so the Check List is not updated and the http status code is changed to “404 Not Found”.