ottuser Service Validations Test plan

# **Test info**

**This test is planned to check three main actions under ottuser service, and to verify it works as expected.**

# **Goals**

* + The system is able to register new user with specific information.
  + The system is able to update user information dynamically, and save the new updates correctly.
  + The system is able to login with users which added recently or updated by the automation tests.
  + The system is able to deal with irrelevant/ invalid inputs, and to return clear message to the users.

# Flow Name: check services commands

**register** Tests:

1) Register new user with valid information (input) and verify the user saved successfully in the system. By getting relevant user id in api response.

* + request link [https://resteus1.ott.kaltura.com/restful\_v5\_0/api\_v3/service/ottuser/action/registr](https://rest-eus1.ott.kaltura.com/restful_v5_0/api_v3/service/ottuser/action/register)

http method=POST

request body *=* {

"partnerId":181,

"user": {

"objectType": "KalturaOTTUser",

"username":"mp3kyymlh1537875168491",

"firstName": "mpky;l'ymlh",

"lastName": "m375168491",

"email": "qah153787591@gmail.com",

"address": "xympdpkyymlh fake address",

"city": "xympdpkyymlh fake city","countryId": 7,

"externalId": "esmpky37875168491" },

"password": "password"

}

2)register **exist** user, and check system return correct error message(User exists)

use the same input you insert in the step above.

**login** Tests:

1) use **exist** user , User which registered in the previous steps, and login with.

verify login action works as expected by checking that the command response.

Contains correct user id.

* + request link <https://resteus1.ott.kaltura.com/restful_v5_0/api_v3/service/ottuser/action/login>
  + http method=POST
  + request body *=* {

partnerId: 185,

username: " ehabmassad2",

password: "password",

extraParams: {

'item1': {

objectType: "KalturaStringValue",

description: "value",

value: "value"

}

},

udid: "value"

}

2) Use a **non-existent** user , for login action , Verify system return relevant error message (user is not exist ).

* + request link <https://resteus1.ott.kaltura.com/restful_v5_0/api_v3/service/ottuser/action/login>
  + http method=POST
  + request body *=* {

partnerId: 185,

username: " ",

password: " ",

extraParams: {

'item1': {

objectType: "KalturaStringValue",

description: "value",

value: "value"

}

},

udid: "value"

**update** Tests:

* + 1. update exist user , **first name** and **last name** values , verify the values are updated by getting relevant api response.
  + request link <https://resteus1.ott.kaltura.com/restful_v5_0/api_v3/service/ottuser/action/update>
  + http method=POST
    - * request body *=* {

user: {

objectType: "KalturaOTTUser",

username: " mp3kyymlh1537875168491",

**firstName: "value",**

**lastName: "value",**

email: "value",

address: "value",

city: "value",

countryId: value,

zip: "value",

phone: "value",

affiliateCode: "value",

externalId: "value",

userType: {

objectType: "KalturaOTTUserType",

description: "value"

},

dynamicData: {

'item1': {

objectType: "KalturaStringValue",

description: "value",

value: "value"

}

},

roleIds: "value"

},

id: "value"

}

* + 1. update a nonexistent user , **first name** and **last name** values , verify the system returned relevant error message (can’t update a non-exist user)
  + request link <https://resteus1.ott.kaltura.com/restful_v5_0/api_v3/service/ottuser/action/update>
  + http method=POST
    - * request body *=* {

user: {

objectType: "KalturaOTTUser",

username: "nonExist",

**firstName: "value",**

**lastName: "value",**

email: "value",

address: "value",

city: "value",

countryId: value,

zip: "value",

phone: "value",

affiliateCode: "value",

externalId: "value",

userType: {

objectType: "KalturaOTTUserType",

description: "value"

},

dynamicData: {

'item1': {

objectType: "KalturaStringValue",

description: "value",

value: "value"

}

},

roleIds: "value"

},

id: "value"

Note : the update action is forbidden , please check the action .

Api response: {

"executionTime": 0.0005671,

"result": {

"error": {

"code": "500004",

"message": "Service Forbidden",

"objectType": "KalturaAPIException"

}

}

}