**Authentication API Document**

***Overview***:

Document gives out a detailed technical aspects of each of the api’s which is listed in below overview table. Document is meant for developers who have knowledge in JOSN, HTTP-Request/Response.

***Table of API’s:***

|  |  |  |  |
| --- | --- | --- | --- |
| SL | API | URL | DESCRIPTION |
| *1* | ***Login*** | authentication/authenticationservice/login | Requested to perform user authenticated login. |
| *2* | ***Logout*** | authentication/authenticationservice/logout | Requested to perform user revoke authentication leads to logout. |
| *3* | ***Get-User-Info*** | authentication/authenticationservice/getUserInfo | Requested to getUserInfo from a valid token. |
| *4* | ***Validate-Token*** | authentication/authenticationservice/validateToken | Requested to validate a generated token. |
| *5* | ***RenewToken*** | authentication/authenticationservice/renewToken | Requested to renew the refresh\_token to access\_token |
| *6* | ***UserSelfRegistration*** | authentication/authenticationservice/selfRegisterUser | Requested to register user by themselves on to the userRepo |
| *7* | ***Get all userdomain*** | authentication/authenticationservice/getAllUserDomains | Requested to get all the domains where the user is existing or registered |
| *8* | ***GetDomainByAccount*** | authentication/authenticationservice/getDomainByAccount | Requested to get all the domains which is existing under a account will have stored. |
| *9* | ***Change-UserPassword*** | authentication/authenticationservice/changeUserPassword | Requested to change the password by user themselves. |
| *10* | ***Reset-Password*** | authentication/authenticationservice/resetPassword | Requested to reset the password at admin level. |
| *11* | ***UserExist*** | authentication/authenticationservice/userExist | Requested to check if user exists in wso2 or not |
| *12* | ***RegisterAccount*** | authentication/authenticationservice/registerNewAccount | Sore managed in external DB for mapping of account vs domain |
| *13* | ***Register Domain By Account*** | authentication/authenticationservice/registerDomainByAccount | Store mapped in external DB to map between account & domains. |
| *14* | ***GetUserInfo-By-Name & Domain*** | authentication/authenticationservice/ getUserInfoByNameAndDomain | One another api to get the user details by name and domain |
| *15* | ***GetAllDomains*** | authentication/authenticationservice/getAllDomains | Request to get All the Domains Registerd in IS |
| *16* | ***findUserByName*** | authentication/authenticationservice/findUsersByName | Request to get the Users By Name in the particular domain |
| *17* | ***findUserByEmail*** | authentication/authenticationservice/findUsersByEmail | Request to get the Users By Email in the particular domain |
| *18* | ***getAllUsers*** | authentication/authenticationservice/getAllUsers | Request to get All the Users associated to that domain |
| *19* | ***lockUser*** | authentication/authenticationservice/lockUser | Request to lock the user from Active state to inactive and vise versa |
| *20* | ***updateUser*** | authentication/authenticationservice/updateUser | Request to update the User information |
| *21* | ***updateUserStatus*** | authentication/authenticationservice/updateUserStatus | Request to Update the User Status |

NB: Above API’s services are assumed to be running in host: ‘localhost’ and port: ‘9060’, & all the needed configurations are setup in the WSO2 IS environment. Base url remains (eg: [http://localhost:9060/ecomm/rest/<*rest-from-table*](http://localhost:9060/ecomm/rest/%3crest-from-table)>). Or the ip/port will be dependent on the deployed environment.

API Manager environment : URL – http:// 52.201.18.79:5091/<api-name>/2.0.0/ <api-name>

Eg : http:// 52.201.18.79:5091/getUserInfoByNameAndDomain/2.0.0/ getUserInfoByNameAndDomain

***Explanation:***

1. **Login**

API to be requested to validate and request a SAML2 based login, and hence internally assertion being interchanged with the OAUTH 2 token and returned as response.

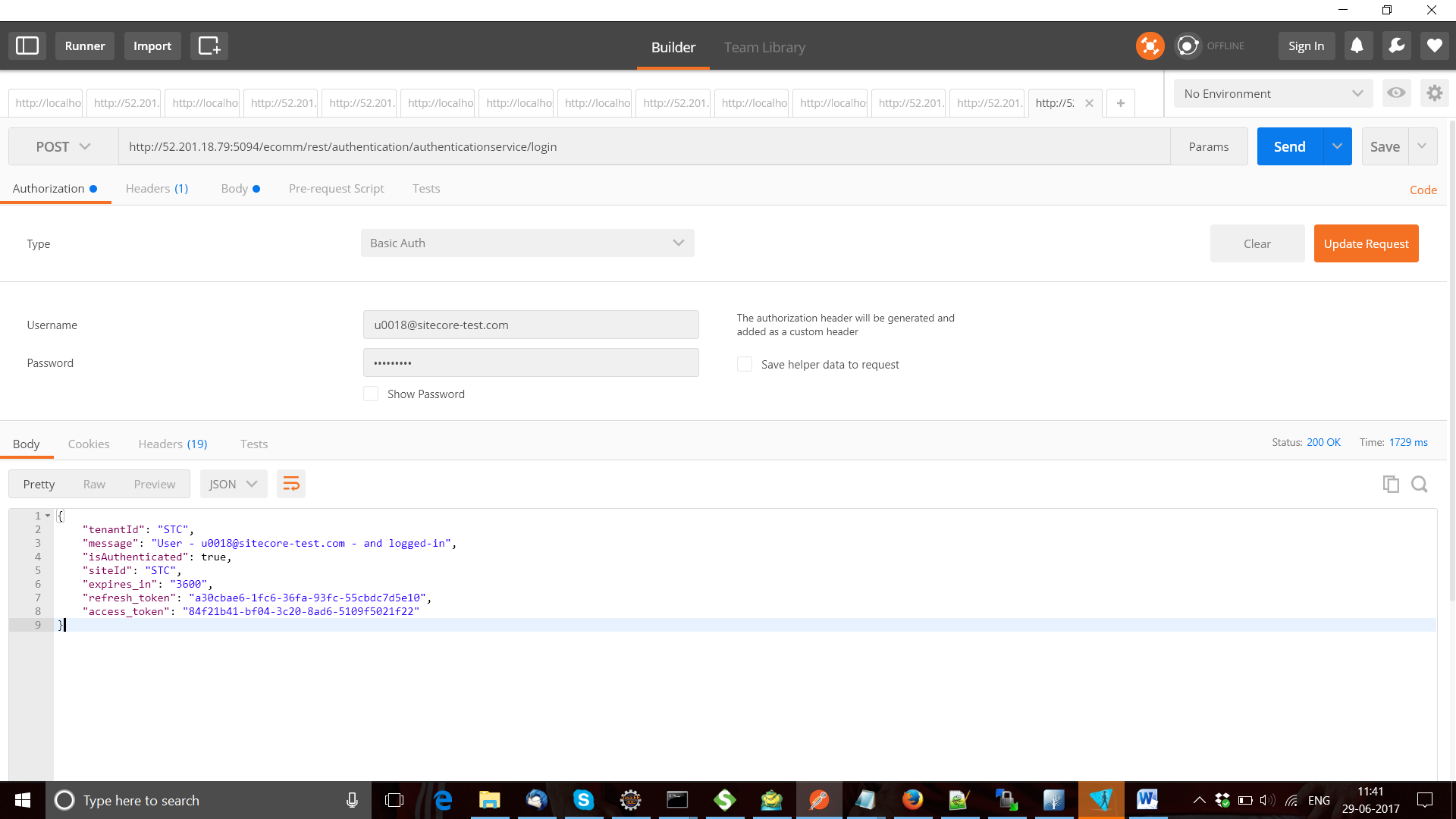
*How to request:*

**URL:** ... /authentication/authenticationservice/login

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| Domain as body | Eg:- {"domain":"carbon.super","userName":"admin@wso2.com","password":"admin"} | *A valid domain which is registered with WSO2 has to be requested, this is to identify to which domain the user belongs.* |
| Authorization headers | Eg:- accountId :ALL  siteId :all | *Valid accountId,siteId which is registered are expected to be passed, otherwise it will be unauthorized.* |

Sample Screenshot:



Response expected :

{"**access\_token**":"e34e11d5-f337-336d-8072-fa603b54cd50","**refresh\_token**":"8f263b4b-5a35-3964-8cfa-fe4e28b52782, ","**isAuthenticated**":true,"**message**":"User - u004@labs.com - and logged-in","**expires\_in**":"3600"}

1. **Logout**

API to be requested to in-validate, hence internally revoke the token, leading to logout.

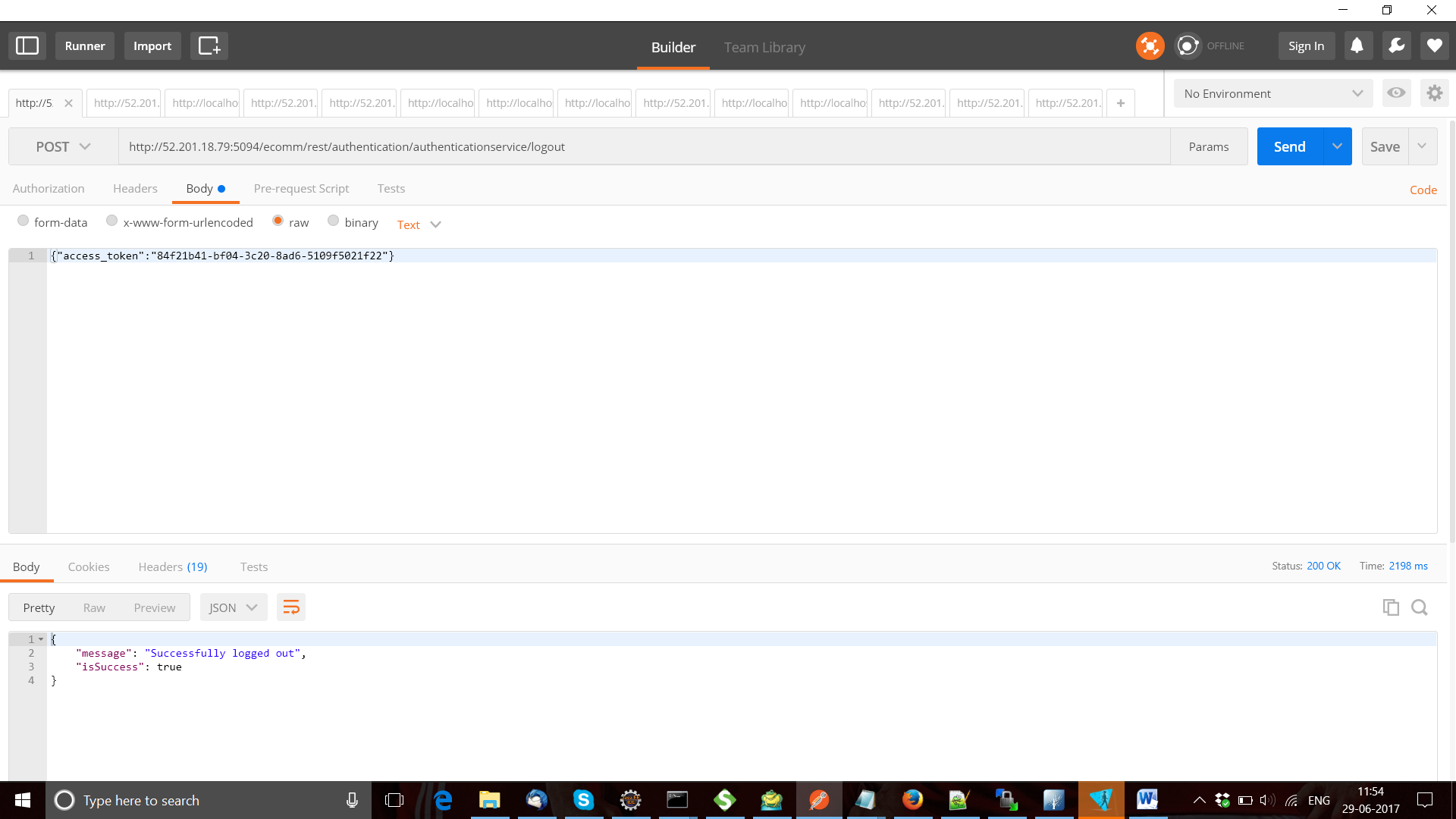
*How to request:*

**URL:** .../authentication/authenticationservice/logout

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| access\_token as body | Eg:- {“access\_token”:” 6964a976-7a67-3bfd-bad7-32292e60f87c”} | *A valid access\_token will be expected to be revoked. Internally it will revoke the dependent attributes if the token is valid, or requested before its expiry.* |
| Authorization headers | Eg:- accountId:ALL,siteId :all,tenant\_token:e34e11d5-f337-336d-8072-a603b54cd50, tenant\_token\_expiration :  1525876834 ,  access\_token : de11d5-f337-336d-8072-fa603b54cd50 | *Valid accountId,siteId,tenant\_token, tenan\_token\_expiration,*  *access\_token which is registered are expected to be passed, otherwise it will be unauthorized.* |

Sample Screenshot:



Img: *access\_token as body.*

Response expected :

{"message":"Successfully logged out","isSuccess":true}

1. **Get-User-Info**

API to be requested to getUserInfo, which internally identifies the ‘subject’ and queries the repository for the attributes.

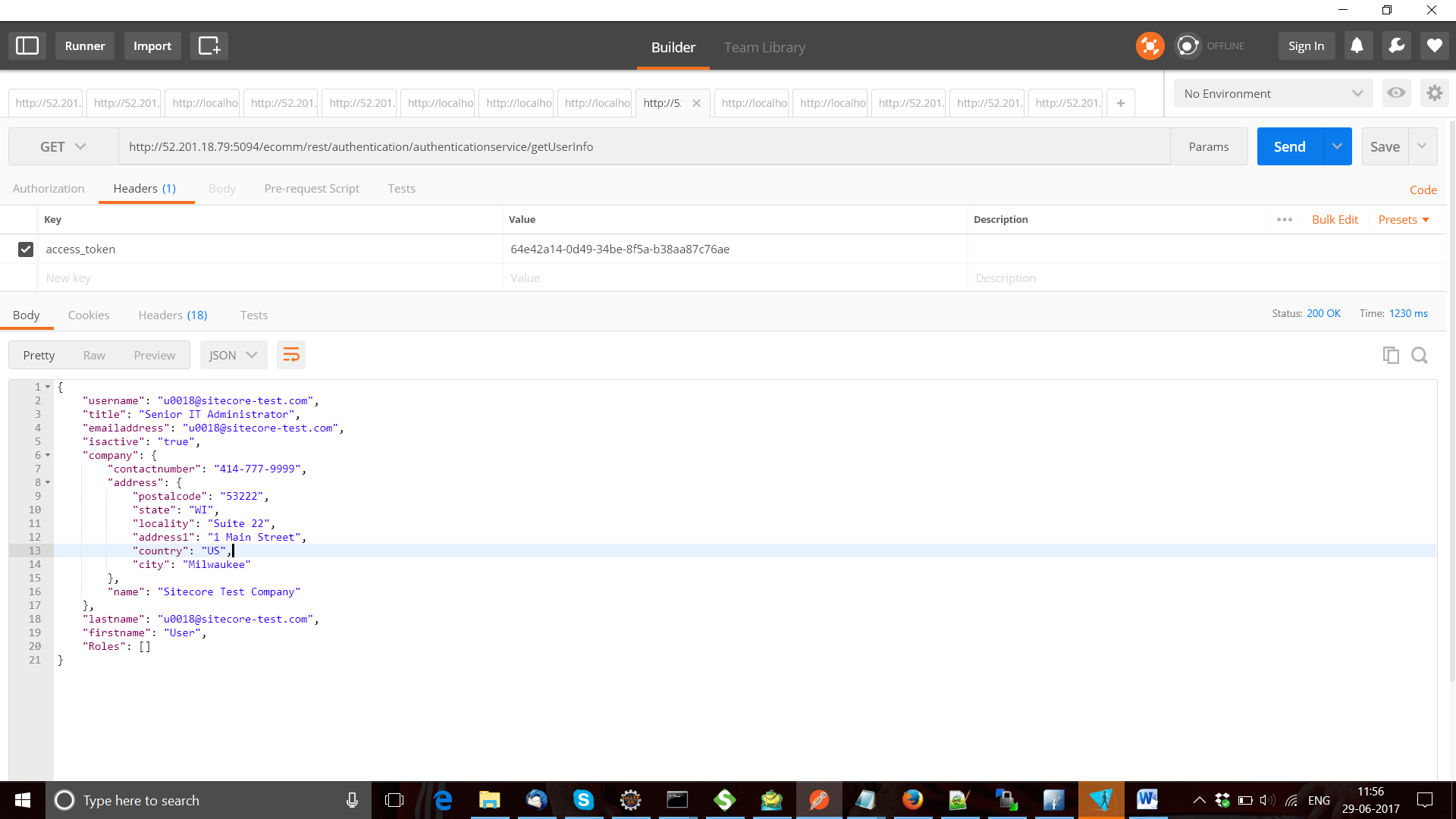
*How to request:*

**URL: …/**authentication/authenticationservice/getUserInfo

**REQUEST PARAMS: (***Methods-allowed***): (GET)**

|  |  |  |
| --- | --- | --- |
| Authorization headers | Eg:-  accountId : ALL,  siteId : all,  leap\_ auth\_token : | *Valid accountId,siteId,tenant\_token, tenan\_token\_expiration,*  *access\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



Img: *access\_token as header key:value pair*

Response expected:

{"username":"u0018@sitecore-test.com","title":"Senior IT Administrator","emailaddress":"u0018@sitecore-test.com","isactive":"true","company":{"contactnumber":"414-777-9999","address":{"postalcode":"53222","state":"WI","locality":"Suite 22","address1":"1 Main Street","country":"US","city":"Milwaukee"},"name":"Sitecore Test Company"},"lastname":"u0018@sitecore-test.com","firstname":"User","Roles":[]}

1. **Validate-Access-Token**

API to be requested to validate access\_token, exposed if developer needs to validate manually from any other external services.

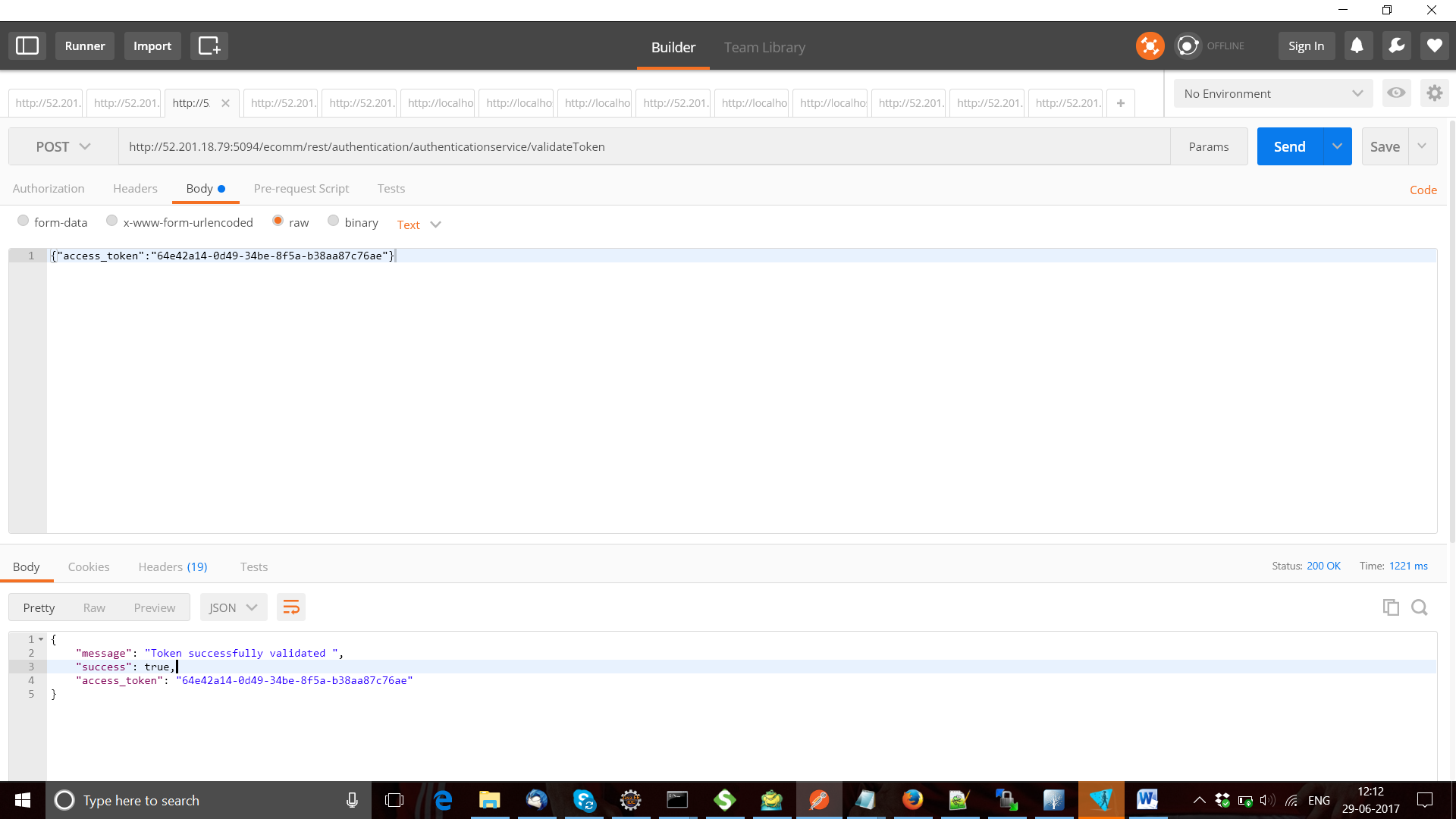
*How to request:*

**URL:** .../authentication/authenticationservice/validateToken *---* [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| access\_token as body (POST) | Eg:- {“access\_token”:”6964a976-7a67-3bfd-bad7-32292e60f87c”} | *A valid token is expected to be requested, else it can result in invalid token.* |
| Authorization headers | Eg:-  accountId : ALL,  siteId : all,  leap\_auth\_token: | *Valid accountId, siteId* |

Sample Screenshot:



Img: *POST with access\_token as body to validate*

Expected response:

{"message":"Token successfully validated ","success":”true"}

1. **Renew-Token**

API to be requested to renew access\_token, exposed if developer needs to pass in refresh\_token and re-generate the access\_token for the same user.

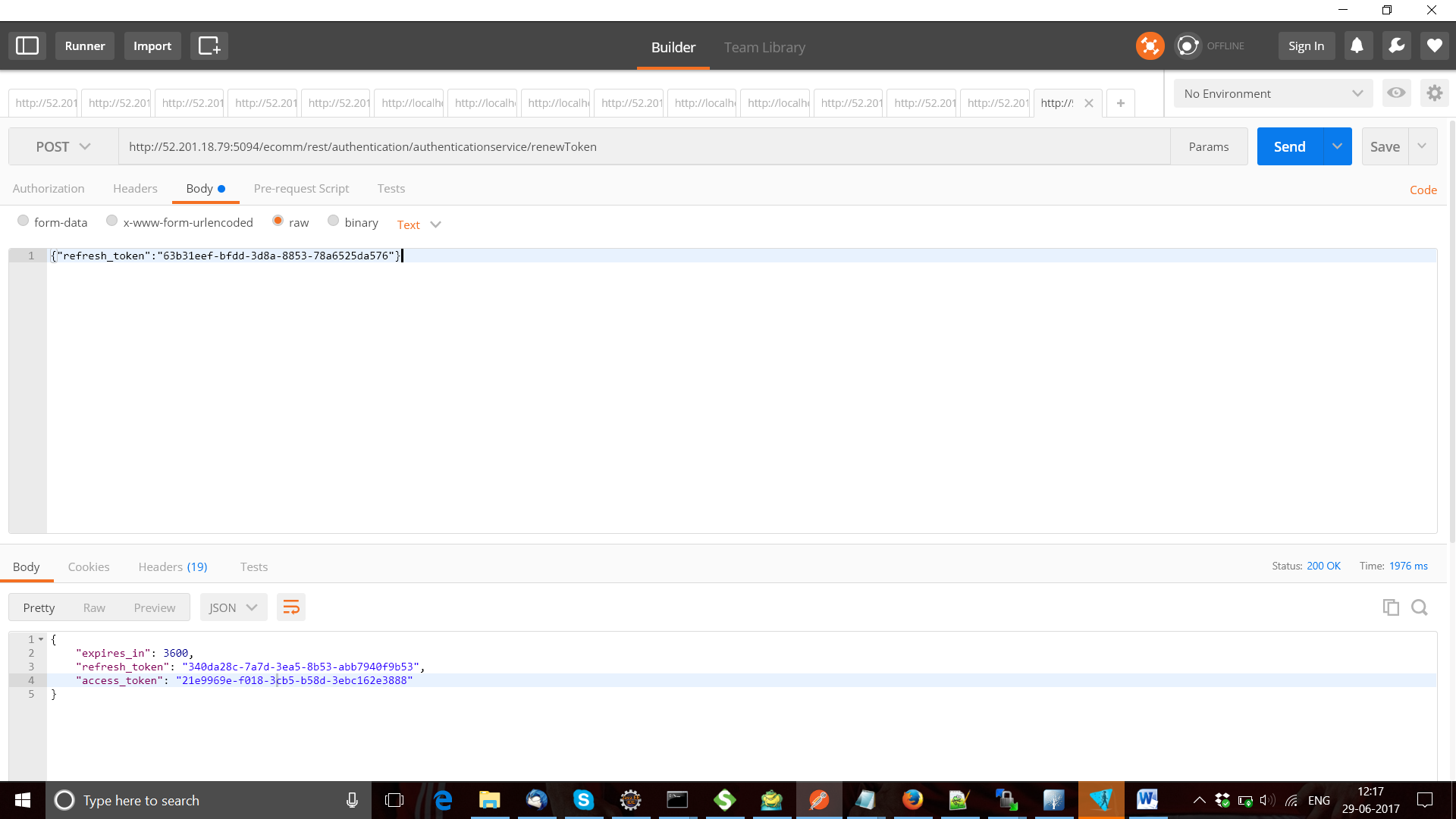
*How to request:*

**URL:** .../authentication/authenticationservice/renewToken --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| refresh\_token as body (POST) | Eg:-  {“refresh\_token”:” 63b31eef-bfdd-3d8a-8853-78a6525da576”} | *A valid* ***refresh*** *token is expected to be requested, else it can result in invalid token.* |
| Authorization headers | Eg:-  accountId : ALL,  siteId :all,  leap\_auth\_token | *Valid accountId, siteId , leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



Img: *a post request to the renewtoken*

Expected response:

{"expires\_in":3600,"refresh\_token":"340da28c-7a7d-3ea5-8b53-abb7940f9b53"}

1. **User-Self-Registration**

API to be requested to register new user and register few attributes within the user-repository.

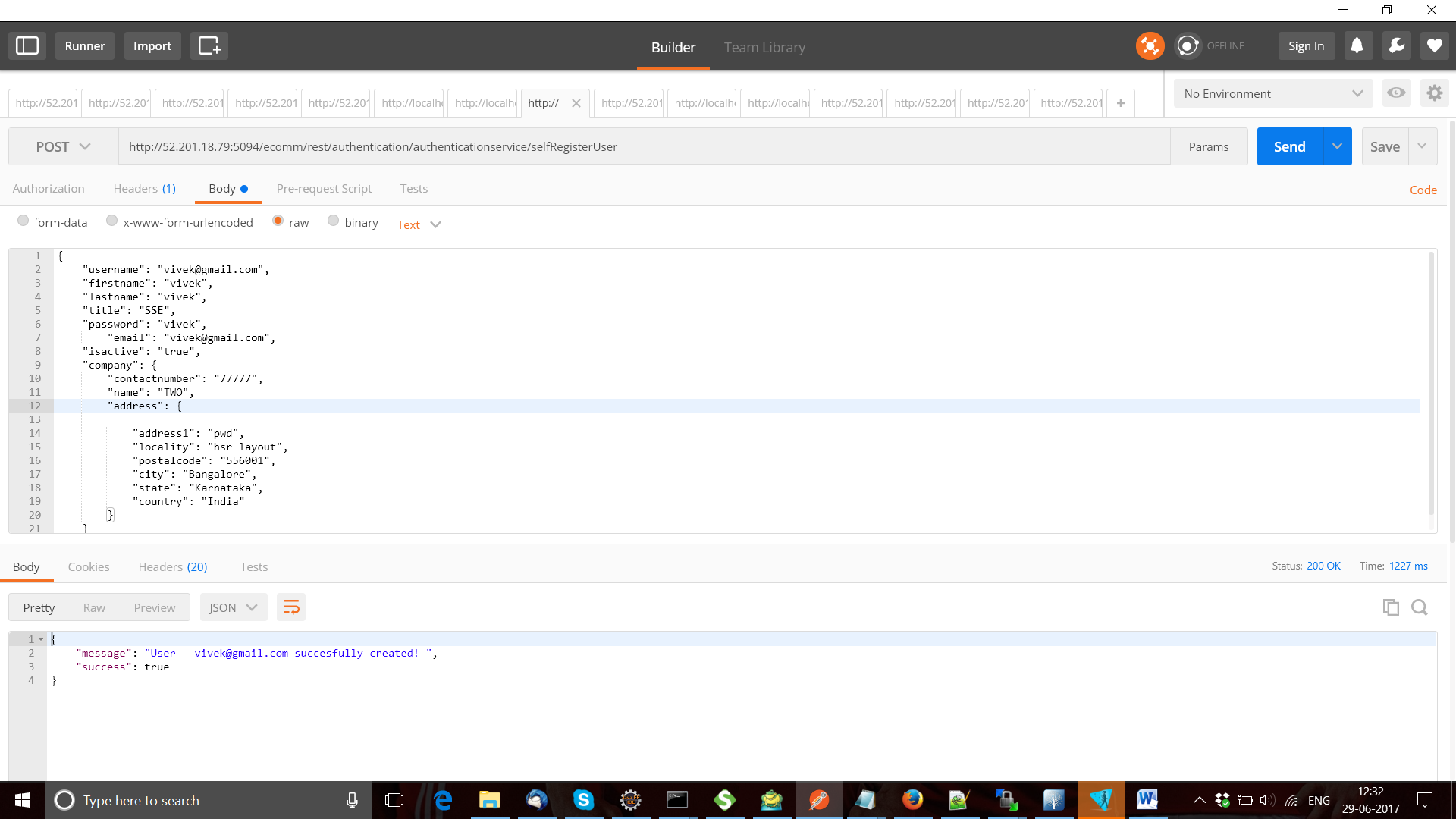
*How to request:*

**URL:** .../authentication/authenticationservice/selfRegisterUser ---[POST]

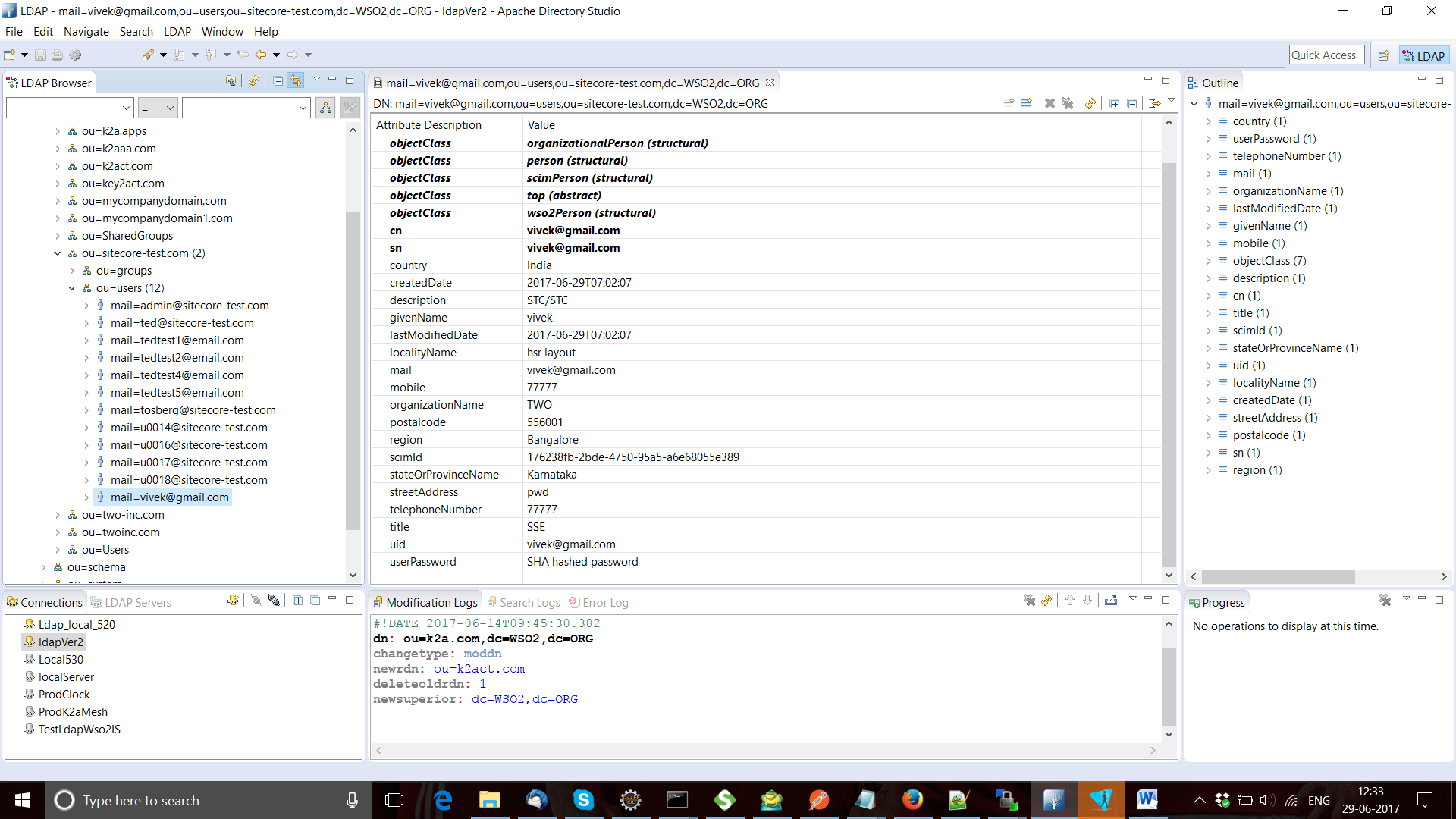
**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **username** | Eg:- u001@twoinc.com | *User name is now same as the email address, hence we don’t need to request for email* |
| **firstname** | Eg:- u001 | *Internally it is the given name* |
| **lastname** | Eg:- u001@twoinc.com | *Lastname is currently the same as email address which is username as well* |
| **password** | Eg:- u001@123 | *This will get registered in reference to the same user.* |
| **title** | Eg:- Software Engineer | *This is the tile of the user to be registered in the repo* |
| **isactive** | Eg:- true | *This is the key to identify whether the user account it enabled / disabled, accepts true/false* |
| **email** | Eg:- u001@twoinc.com | *Username, lastname & email are currently the same value* |
| **company/name** | Eg:- Two inc | *This is registers in repo against one user* |
| **company/contactnumber** | Eg:- 4444444444 | *This is registers in repo against one user* |
| **company/address/ address1** | Eg:- Street 1 | *This is registers in repo against one user* |
| **company/address/locality** | *---//has-to-be-done//---* | *Nill* |
| **Company/address/postalcode** | *---//has-to-be-done//---* | *Nill* |
| **company/address/city** | Eg:- City 1 | *This is registers in repo against one user* |
| **company/address/state** | Eg:- Wisconsin | *This is registers in repo against one user* |
| **company/address/country** | Eg:- USA | *This is registers in repo against one user* |
| **domain (as header)** | Eg:-test.com | *This will identify the domain where the user has to be added* |
| Authorization headers | Eg:-  domain:test.com, accountId:ALL,  siteId : all,  leap\_auth\_token : | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



Img : Successful POST request to registerUser.



Img: Successful repo update of the user

Sample User Registration Request data:

{

"username": "vijay@gmail.com",

"firstname": "vijay",

"lastname": "vijay",

"title": "SSE",

"password": "vijay",

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

"isactive": "true",

"islocked": "false",

"company": {

"contactnumber": "77777",

"name": "TWO",

"address": {

"address1": "pwd",

"locality": "hsr layout",

"postalcode": "556001",

"city": "Bangalore",

"state": "Karnataka",

"country": "India"

}

}

}

1. **Get All User Domain**

API to be requested to get all domain for user

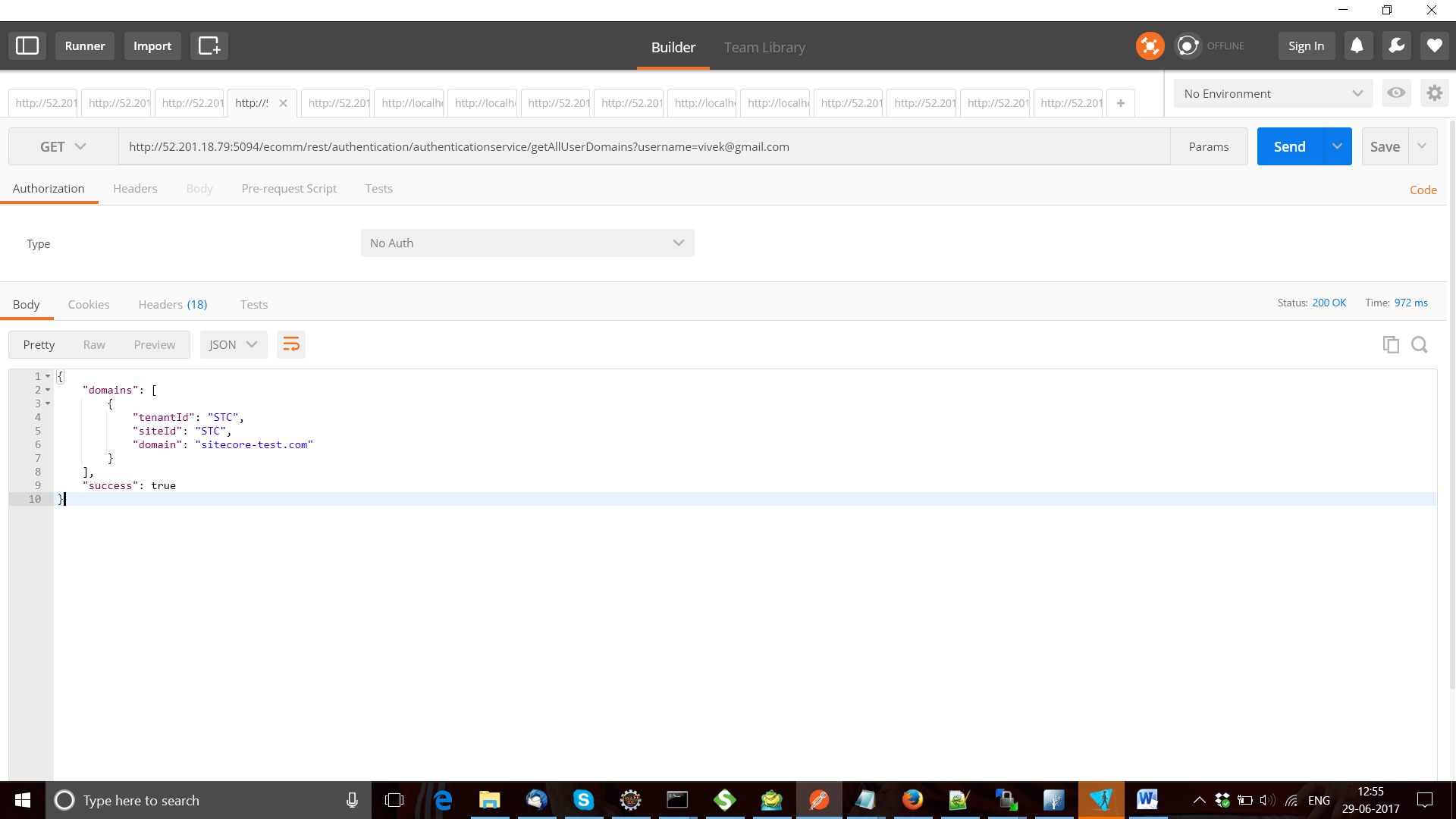
*How to request:*

**URL:** ... …/authentication/authenticationservice/getAllUserDomains? username=<username> --- [GET]

**REQUEST PARAMS: (***Methods-allowed***): (GET)**

|  |  |  |
| --- | --- | --- |
| username as request parameter (GET) | Eg:- username = u001@email.com | *A valid* ***username*** *is expected to get the list of domains the user is registered.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

Screenshot:



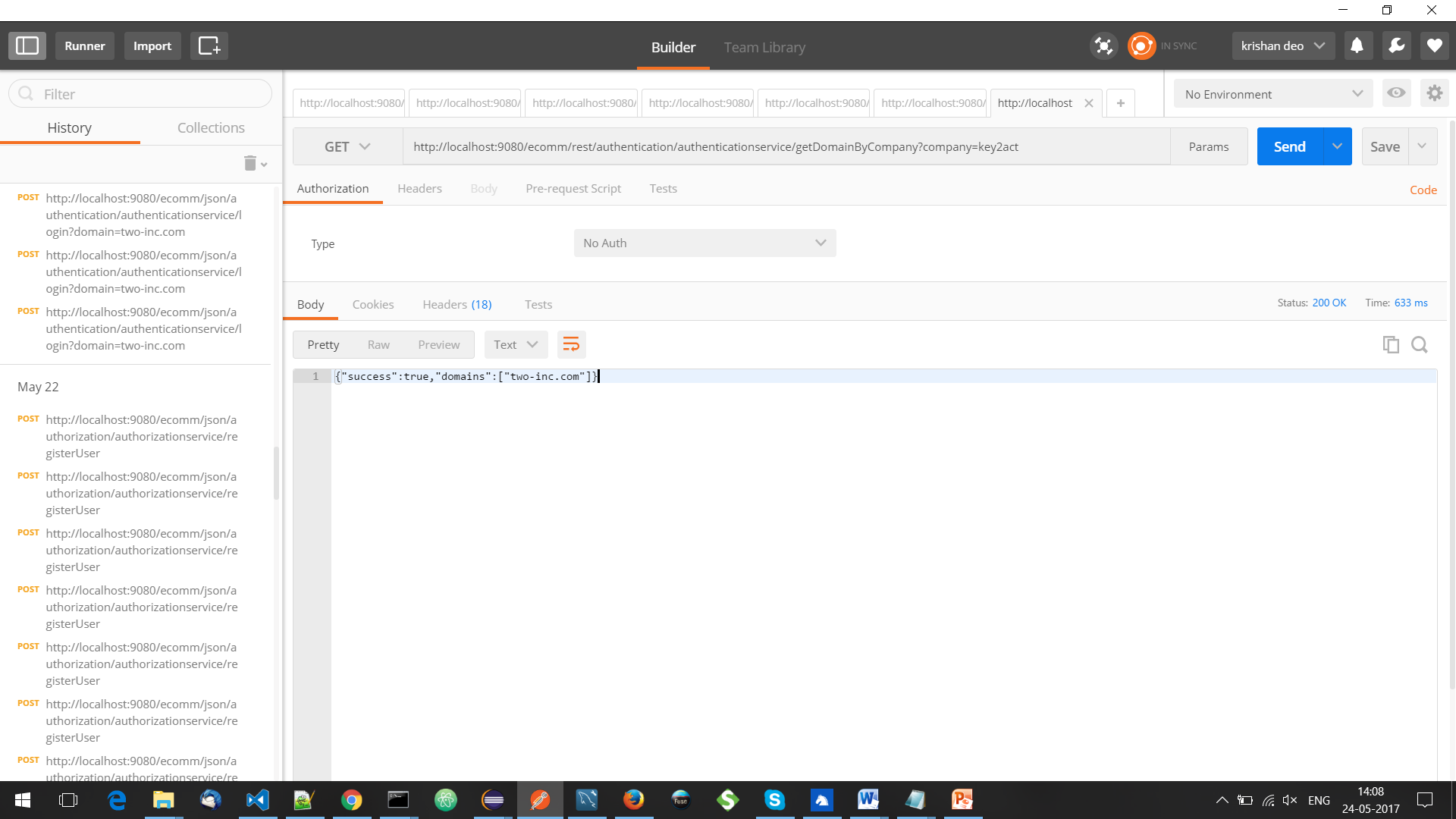
1. **Get Domain Name By AccountName**

API to be requested to get all domain names by Account name

**URL:**

…/authentication/authenticationservice/getDomainByAccount? company=<account \_name> --- [GET]

|  |  |  |
| --- | --- | --- |
| Authorization headers | Eg:-  accountId:ALL,  siteId : all,  leap\_auth\_token | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |



1. **Change-User-Password**

API to be requested to change the registered user’s password, unless the user account is enabled/un-locked and valid old password in request falls in place this service will not replace the password to the latest being requested.

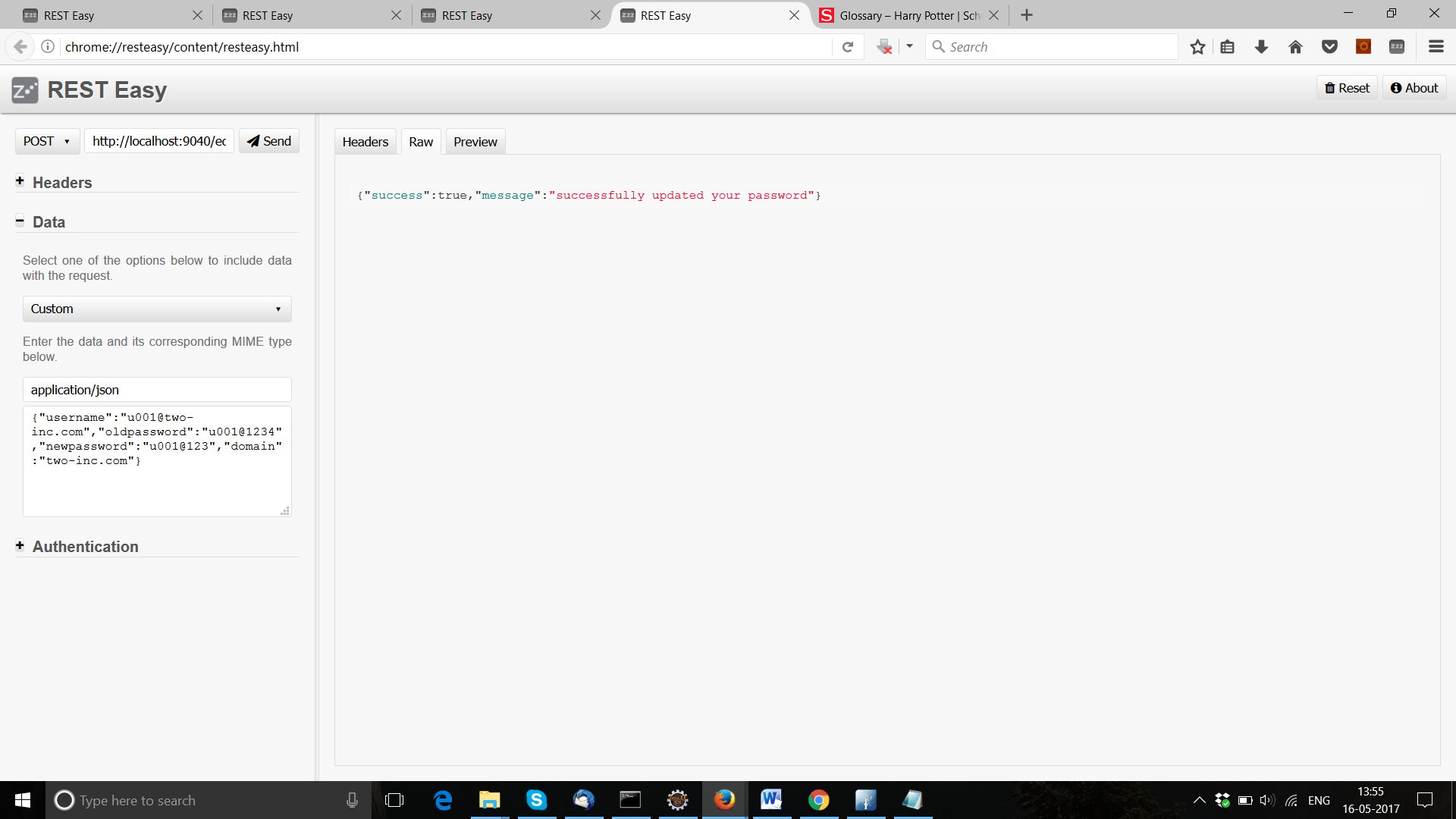
*How to request:*

**URL:** .../authentication/authenticationservice/changeUserPassword ---[POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| Username, old password, new password & domain as body (POST) | Eg:- {"username":"ted@twoinc.com","oldpassword":"ted@13","newpassword":"ted@123","domain":"two-inc.com"} | *A valid* ***request params*** *expected to be requested, else it can result in no-change of old password.* |
| Authorization headers | Eg:-  accountId : ALL,  siteId : all,  leap\_auth\_token : | *Valid accountId,siteId,tenant\_token, tenan\_token\_expiration,*  *access\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



Img : *post-data to change the password by user.*

Expected Response:

{"**success**":”true”,"**message**":"successfully updated your password"}

1. **Reset-User Password**

API to be requested to reset the registered users password, unless the admin(tenant-admin) requests with their credentials this service will not replace the password to the latest being requested.

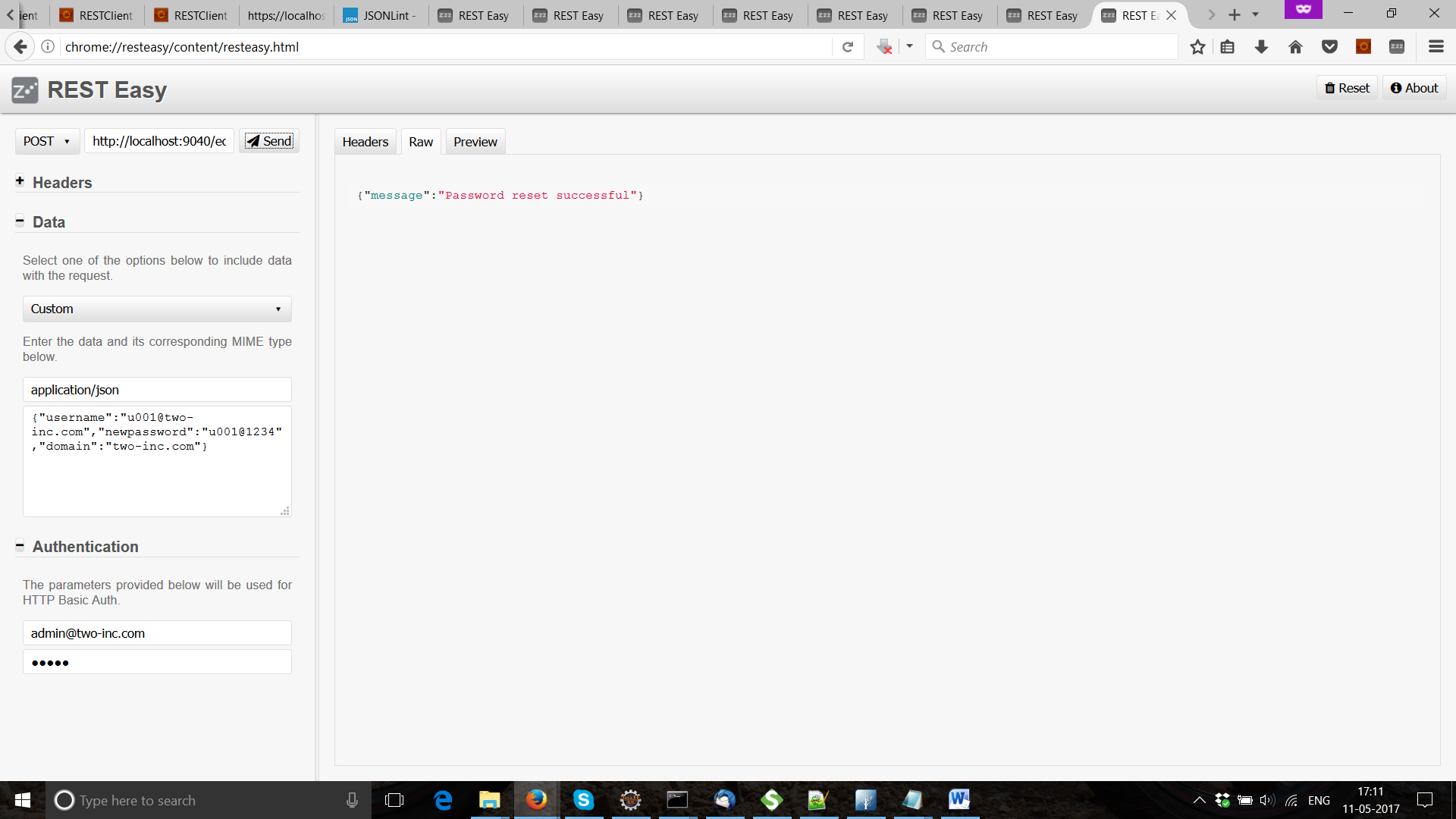
*How to request:*

**URL:** .../authentication/authenticationservice/resetPassword --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| Username, newpassword & domain as body (POST) | Eg:- {"username":"ted@twoinc.com","newpassword":"ted@1234","domain":"two-inc.com"} | *A valid* ***request params*** *expected to be requested, else it can result in no-change of old password.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId : all,  leap\_auth\_token : | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



1. **User Exist**

API to be requested to check if user already exist in repo or not.

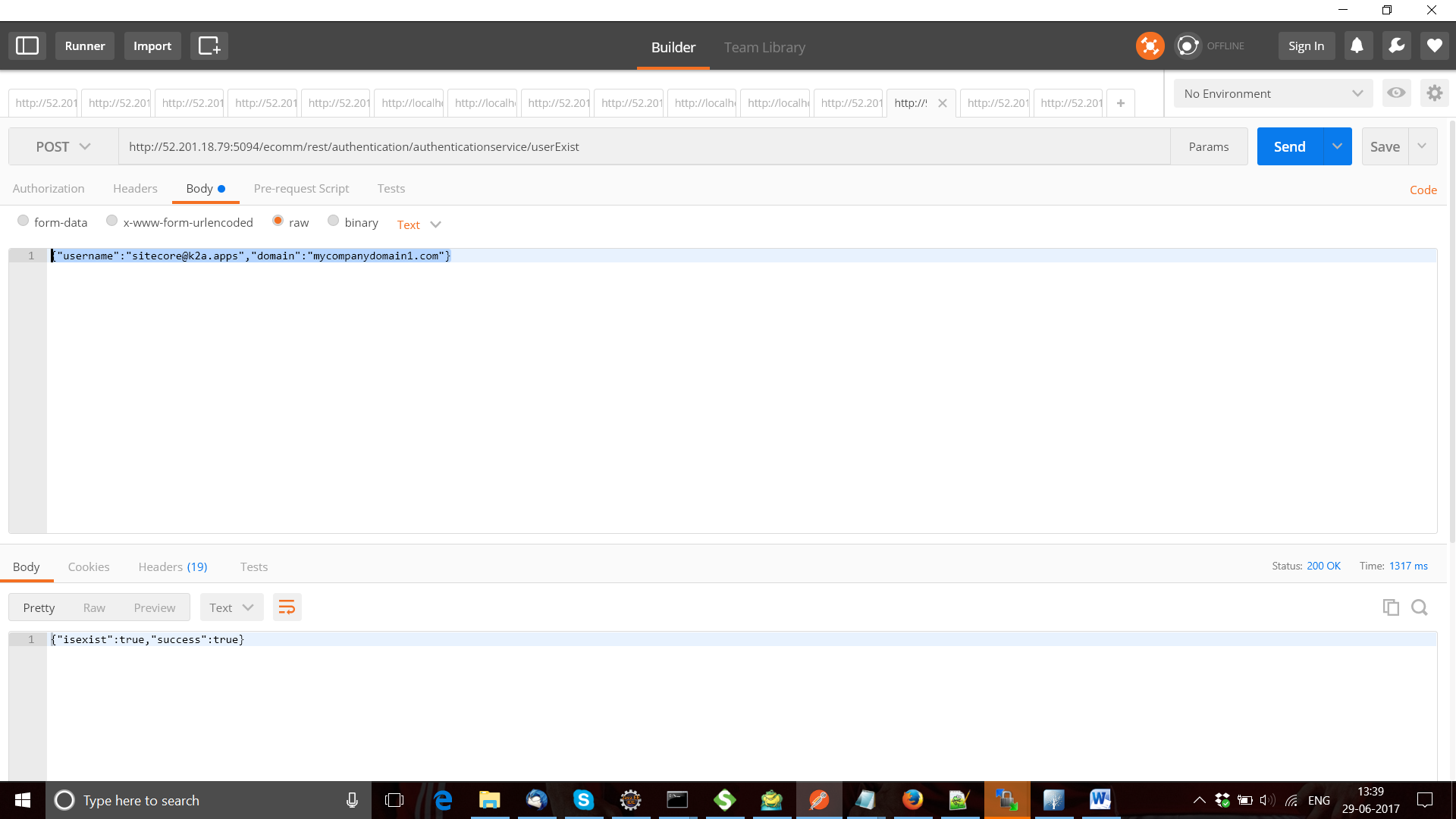
*How to request:*

**URL: ../**authentication/authenticationservice/userExist --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **Username as body** | Eg:- {"username":"sitecore@k2a.apps","domain":"mycompanydomain1.com"} | *User name is now same as the email address, hence we don’t need to request for email* |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token : | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

**Sample Screenshot:**



1. **Register Account**

API to request to register new account which will have domains associated.

*How to request:*

**URL:** .../authentication/authenticationservice/registerNewAccount ---[POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **Body** | Eg:-  {"accountName " : "GetUsRoi",  “internalTenantId” : “getusroi”,  “tenantToken\_expiration\_ini\_in\_seconds”: “3600”} | *Tenant token expiration valid time in seconds* |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

SAMPLE REQUEST:

{

"accountName": "testattuendlabs1",

"internalTenantId":"testattuendlabs1, Inc.",

"tenantToken\_expiration\_init\_in\_seconds":"3600"

}

Expected Response:

{"message":"Successfully registerd - mycompany-codetest - as anew company!","success":”true”}

1. **RegisterDomainByAccount**

API to request to register new account which will have domains associated.

*How to request:*

**URL:** .../authentication/authenticationservice/registerDomainByAccount ---[POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **accountname** | Eg:- {"newdomain":{"companyname":"mycompany-codetest","domainname":"mycompanydomain3.com","tenantid":"MY","siteid":"MY","adminfirstname":"admin","adminlastname":"admin","adminusername":"admin@email.com","adminpassword":"admin","adminemail":[admin@email.com](mailto:admin@email.com), ","timezone":"IST”}} | *Companyname should be valid to get mapped to, i.e.; company should already exist in order to register the domain under that Company* |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

SAMPLE REQUEST:

{"newdomain":{

"accountname":"testattuendlabs1",

"domainname":"labs1.com",

"siteid":"labs1",

"adminfirstname":"admin",

"adminlastname":"admin",

"adminusername":"admin@labs1.com",

"adminpassword":"admin",

"adminemail":"admin@labs1.com",

"timezone":"IST”

}

}

1. **GetUserInfoByNameAndDomain**

API to request to get user information based on registered user name and associated domain.

*How to request:*

**URL: ...**/authentication/authenticationservice/ getUserInfoByNameAndDomain --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **Username, domain name as body** | Eg:- {"**username**":"vijay.paranthaman@gmail.com","**domain**":"mycompanydomain3.com"} | *Username should be valid to get mapped to, ie.; username should already exist inorder to get user information associated with particular domain.* |
| Authorization headers | Eg:-  accountId : ALL,  siteId :all,  leap\_auth\_token : | *Valid accountId,*  *siteId,*  *leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |
|  |  |  |

1. **GetAllDomains**

API to request to get all domains based on registered account and associated site.

*How to request:*

**URL: ...**/authentication/authenticationservice/ getAllDomains --- [GET]

**REQUEST PARAMS: (***Methods-allowed***): (GET)**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token: | *Valid accountId,*  *siteId,*  *leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

1. **FindUserByName**

API to request to get user information based on registered user name and associated domain.

*How to request:*

**URL: ...**/authentication/authenticationservice/findUserByName --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **namekeyword** | Eg:- {"namekeyword":"\*","pageindex":"0","pagesize":"20","domain":"labs1.com"} | *namekeyword should be valid to get mapped to, i.e.; get user information associated with particular domain.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId:all,  leap\_auth\_token | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

1. **FindUserByEmail**

API to request to get user information based on registered user email and associated domain.

*How to request:*

**URL: ...**/authentication/authenticationservice/findUserByEmail --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **body** | Eg:- {"emailkeyword":"\*","pageindex":"0","pagesize":"20","domain":"labs1.com"} | *email keyword should be valid to get mapped to, ie.; get user information associated with particular domain.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId:all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

1. **GET ALLUSERS**

API to request to get all users information based on registered domain and associated domain.

*How to request:*

**URL: ...**/authentication/authenticationservice/getAllUsers --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| body | Eg:- {"pageindex":"0","pagesize":"20","domain":"labs1.com"} | *pageindex to display number of users* |
| Authorization headers | Eg:-  accountId:ALL,  siteId :all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |
|  |  |  |

1. **LOCKUSER**

API to request to lock the user from the active state to inactive state and vise versa

*How to request:*

**URL: ...**/authentication/authenticationservice/lockUser --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| body | Eg:- {"domain":"labs.com","username":"u004@labs.com","lock": "false"} | *Domain should be valid to get mapped to, i.e.; domain should already exist in order to get information.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId:all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token which is registered are expected to be passed, otherwise it will be unauthorized* |

1. **Update User**

API to be requested to update user information

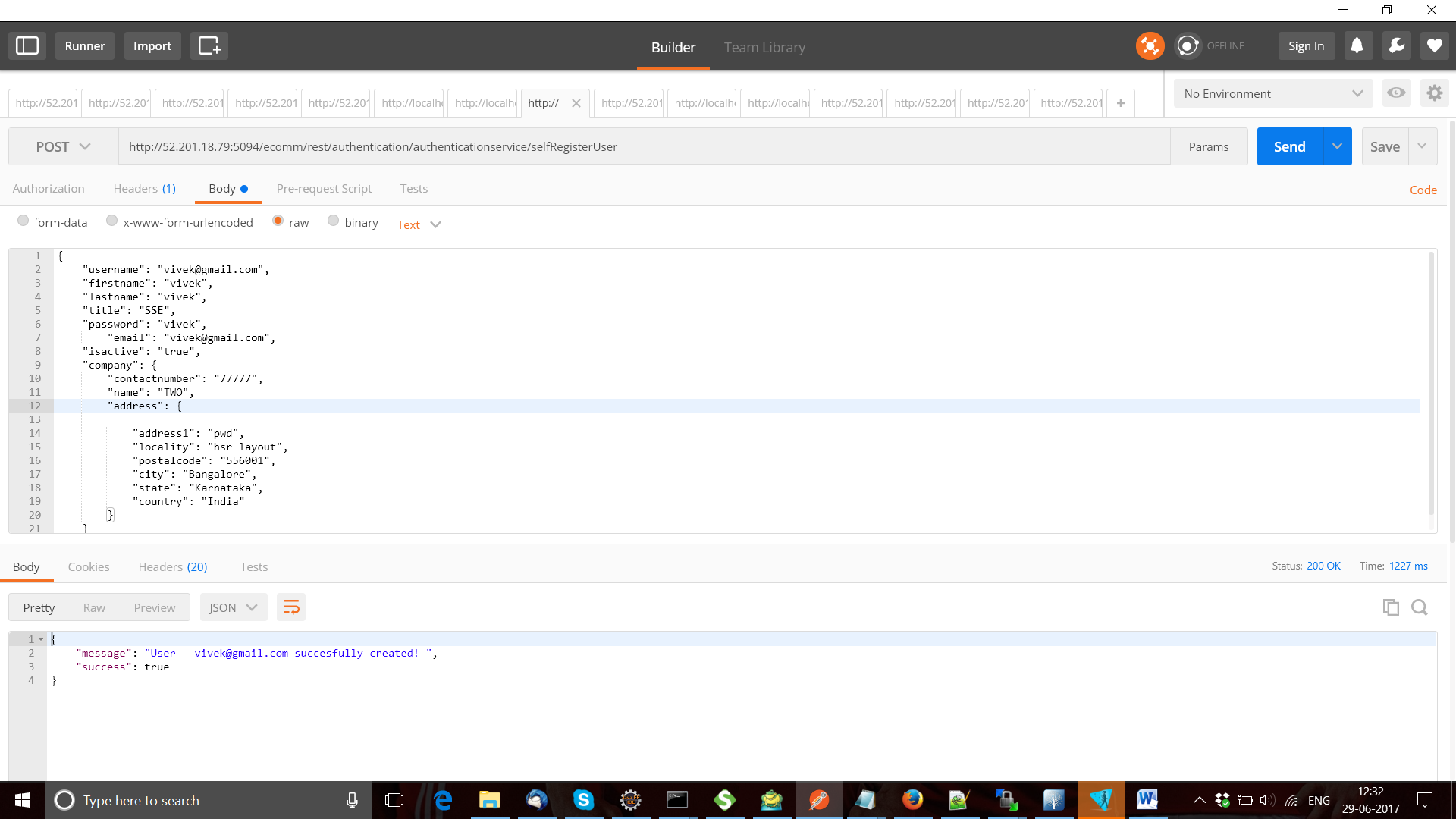
*How to request:*

**URL:** .../authentication/authenticationservice/updateUser --- [POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| **username** | Eg:- u001@twoinc.com | *User name is now same as the email address, hence we don’t need to request for email* |
| **firstname** | Eg:- u001 | *Internally it is the given name* |
| **lastname** | Eg:- u001@twoinc.com | *Lastname is currently the same as email address which is username as well* |
| **password** | Eg:- u001@123 | *This will get registered in reference to the same user.* |
| **title** | Eg:- Software Engineer | *This is the tile of the user to be registered in the repo* |
| **isactive** | Eg:- true | *This is the key to identify whether the user account it enabled / disabled, accepts true/false* |
| **email** | Eg:- u001@twoinc.com | *Username, lastname & email are currently the same value* |
| **company/name** | Eg:- Two inc | *This is registers in repo against one user* |
| **company/contactnumber** | Eg:- 4444444444 | *This is registers in repo against one user* |
| **company/address/ address1** | Eg:- Street 1 | *This is registers in repo against one user* |
| **company/address/locality** | *---//has-to-be-done//---* | *Nill* |
| **Company/address/postalcode** | *---//has-to-be-done//---* | *Nill* |
| **company/address/city** | Eg:- City 1 | *This is registers in repo against one user* |
| **company/address/state** | Eg:- Wisconsin | *This is registers in repo against one user* |
| **company/address/country** | Eg:- USA | *This is registers in repo against one user* |
| **domain (as header)** | Eg:-test.com | *This will identify the domain where the user has to be added* |
| Authorization headers | Eg:-  domain:test.com, accountId:ALL,  siteId:all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |

Sample Screenshot:



Img: Successful POST request to registerUser.

Sample Update User Request data:

{

"username": "vijay@gmail.com",

"firstname": "vijay",

"lastname": "vijay",

"title": "SSE",

"password": "vijay",

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

"isactive": "true",

"islocked": "false",

"company": {

"contactnumber": "77777",

"name": "TWO",

"address": {

"address1": "pwd",

"locality": "hsr layout",

"postalcode": "556001",

"city": "Bangalore",

"state": "Karnataka",

"country": "India"

}

}

}

1. **UpdateUserStatus**

API to request to update the User Status information

*How to request:*

**URL: ...**/authentication/authenticationservice/ updateUserStatus---[POST]

**REQUEST PARAMS: (***Methods-allowed***): (POST)**

|  |  |  |
| --- | --- | --- |
| body | Eg:- {"domain":"labs.com","username":"u004@labs.com","email": "u004@gmail-labs.com","isactive": "true"} | *Domain should be valid to get mapped to, i.e.; domain should already exist in order to get information.* |
| Authorization headers | Eg:-  accountId:ALL,  siteId:all,  leap\_auth\_token: | *Valid accountId, siteId, leap\_auth\_token*  *which is registered are expected to be passed, otherwise it will be unauthorized* |