Media Scenes API

Documentation for API created

8/19/2016

Laitkor Infosolutions Pvt. Ltd.

LISP

Development Environment

1- Visual Studio 2015

2- .Net Framework 4.6.2

3- ASP.NET Web Api 2.0

4- Token Based Authentication

May be these changs will required in “web.config” for managing call of web api

<system.webServer>

<modules>

<remove name="WebDAVModule" />

<remove name="FormsAuthentication" />

<remove name="ApplicationInsightsWebTracking" />

</modules>

<handlers>

<remove name="WebDAV" />

</handlers>

</system.webServer>

Login API

## Argument

This method has the URL

[http://192.168.1.149:8000/token](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| username | [aa@demo.com](mailto:aa@demo.com) | Yes | User Login | |
| password | 123456 | Yes |  | |
| grant\_type | **password** | Yes |  | |

**Note :** In above arguments grant\_type must be password(word)

## Response

If successful, it will returns all roles

*For example:-*

## {

## "access\_token": "D6Mtu1f8S\_JsxZsfapPBGiLYruywi7mXEblaldg\_bWrRfs0twv2wuIXvZiamN8r1PHefX1YHYKBVQjDaS\_caxziqqu9G-1XXj1FdYkTQdXD\_Sh3-\_Oz-9whbKgyjrxhr1KD0SRl5oXGjoibIbpwbKlL543X1FMVcsjFKKcx-gxHLLrNRv7wjLtm1h4RRGWOal8Z\_ZkbzCvcwTKOOZAxG6eMu5vBVrwJEcXKsuLg4R-9wIS28B1ZArVMIBT9QaUyHJ85m0Lzlh78xoZXhBTmLApv0Xq9dk74iG5EKOIERDgNDZXtTxvAH8ZYnAAz5YzpZ",

## "token\_type": "bearer",

## "expires\_in": 86399,

## "userId": "1",

## "userName": "AA",

## "emailId": "aa@demo.com",

## "roleId": "6",

## "role": "Organization",

## "registrationType": "2",

## "databaseId": "1",

## ".issued": "Tue, 23 Oct 2018 11:09:11 GMT",

## ".expires": "Wed, 24 Oct 2018 11:09:11 GMT"

## }

## Errors

If not successful check token is passed or not.

{

"error": "invalid\_grant",

"error\_description": "Provided username or password is incorrect"

}

Master DB API Documentation

1- CRUD API for Role

API 1 : GetAllRole

This method is used for **GetAllRole**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/role/getallrole](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get all Roles | |

## Response

If successful, it will returns all roles

*For example:-*

[{"roleid":1,"role":"Clerk"},{"roleid":2,"role":"Employee"},{"roleid":3,"role":"Manager"},{"roleid":5,"role":"Admin"},{"roleid":6,"role":"Organization"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : GetRoleById

This method is used to **GetRoleById**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/role/getrolebyid/5>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get Role By Id |

## Response

If successful, it will return detail of Role.

*For Example:-*

## [{"roleid":5,"role":"Admin"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveRole

This method is used to **SaveRole**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/role/saverole>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Save Role |
| role | Admin | Yes |  |
| action | add | Yes |  |

## Response

If successful, it will return the id of inserted record.

*For Example:-*

9

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 4 : UpdateRole

This method is used to **UpdateRole**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/role/updaterole>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update selected Role |
| role | User | Yes |  |
| action | update | Yes |  |
| roleid | 1 | Yes |  |

## Response

If successfully update it will return the updated record Id

1

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

API 5 : DeleteRole

This method is used to **DeleteRole**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/role/deleterole>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete selected Role |
| action | delete | Yes |  |
| roleid | 9 | Yes |  |

## Response

If successfully deleted it will return the deleted record Id

9

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

2- CRUD API for Registration Type

API 1 : GetAllRegistrationType

This method is used for **GetAllRegistrationType**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/RegistrationType/GetAllRegistrationType](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get all Registration Types | |

## Response

If successful, it will returns all Registration Types

*For example:-*

## [{"regtypeid":1,"regtype":"Organization"},{"regtypeid":2,"regtype":"User"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : GetRegistrationTypeById

This method is used to **GetRegistrationTypeById**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/RegistrationType/GetRegistrationTypeById/2](http://192.168.1.149:8000/api/RegistrationType/GetRegistrationTypeById/2%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get Registration Type By Id |

## Response

If successful, it will return detail of Registration Type.

*For Example:-*

## [{"regtypeid":2,"regtype":"User"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveRegistrationType

This method is used to **SaveRegistrationType**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/RegistrationType/SaveRegistrationType>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Save Registration Type |
| regtype | User | Yes |  |
| action | add | Yes |  |

## Response

If successful, it will return the id of inserted record.

*For Example:-*

3

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 4 : UpdateRegistrationType

This method is used to **UpdateRegistrationType**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/RegistrationType/UpdateRegistrationType>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update selected Registration Type |
| regtype | Organization | Yes |  |
| action | update | Yes |  |
| RegTypeId | 1 | Yes |  |

## Response

If successfully update it will return the updated record Id

1

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

API 5 : DeleteRegistrationType

This method is used to **DeleteRegistrationType**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/RegistrationType/DeleteRegistrationType](http://192.168.1.149:8000/api/RegistrationType/DeleteRegistrationType%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete selected Registration Type |
| action | delete | Yes |  |
| RegTypeId | 2 | Yes |  |

## Response

If successfully deleted it will return the deleted record Id

2

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

3- CRUD API for User

API 1 : GetAllUser

This method is used for **GetAllUser**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/register/getalluser](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get all Users | |

## Response

If successful, it will returns all Users

*For example:-*

## [{"userid":1,"username":"Amrish23423asda","roleid":6,"address":"Delhi","emailid":"c3qc@cc.com","contactno":"9874563210","password":"D5-36-FF-1B-AE-EF-53-69-49-02-09-EA-E4-C1-9F-20","registrationtype":2,"regtype":"User","role":"Organization","createdby":1,"createdon":"2018-10-15T15:26:40.357"},{"userid":2,"username":"Amrish23423asdasdfsdfsdasdfasdf","roleid":6,"address":"Mumbai","emailid":"c3q2c@cc.com","contactno":"9874563210","password":"00-CE-96-FB-B9-61-88-72-60-C6-56-11-D4-84-07-4A","registrationtype":2,"regtype":"User","role":"Organization","createdby":1,"createdon":"2018-10-15T16:12:41.01"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : GetUserById

This method is used to **GetUserById**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/register/getuserbyid/2](http://192.168.1.149:8000/api/register/getuserbyid/2%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get User By Id |

## Response

If successful, it will return detail of User.

*For Example:-*

## [{"userid":2,"username":"Amrish23423asdasdfsdfsdasdfasdf","roleid":6,"address":"Mumbai","emailid":"c3q2c@cc.com","contactno":"9874563210","password":"00-CE-96-FB-B9-61-88-72-60-C6-56-11-D4-84-07-4A","registrationtype":2,"regtype":"User","role":"Organization","createdby":1,"createdon":"2018-10-15T16:12:41.01"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveUser

This method is used to **SaveUser**

## Argument

This method has the URL

<http://192.168.1.149:8000/api/register/saveuser>

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| action | add | Yes | Save User |
| UserName | Albert | Yes |  |
| RoleId | 1 | Yes |  |
| Address | Delhi | Yes |  |
| EmailId | [albert@demo.com](mailto:albert@demo.com) | Yes |  |
| ContactNo | 2587413690 | Yes |  |
| DatabaseId | 2 | Yes |  |
| Password | Password | Yes |  |
| RegistrationType | 3 | Yes |  |
| CreatedBy | 2 | Yes |  |

## Note: If RoleId is of organization then a new database will be generated for that organization(e.g in our case organization id is 6. You can change it according to your requirement)

## Response

If successful, it will return the id of inserted record.

*For Example:-*

3

## Errors

If not successful

For Example :

{"Message":"Enter EmailId. Enter Contact Number. Enter Database Id. Enter Password. Enter Registration Type Id. "}

API 4 : UpdateUser

This method is used to **UpdateUser**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/register/updateuser](http://192.168.1.149:8000/api/RegistrationType/UpdateRegistrationType)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update User |
| action | update | Yes |  |
| UserId | 2 | Yes |  |
| UserName | Albert123 | Yes |  |
| RoleId | 1 | Yes |  |
| Address | Agra | Yes |  |
| EmailId | [albert@demo.com](mailto:albert@demo.com) | Yes |  |
| ContactNo | 2587413690 | Yes |  |
| DatabaseId | 2 | Yes |  |
| Password | Password123 | Yes |  |
| RegistrationType | 3 | Yes |  |
| CreatedBy | 2 | Yes |  |

## Response

If successfully update it will return the updated record Id

2

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Enter EmailId. Enter Contact Number. Enter Database Id. Enter Password. Enter Registration Type Id. "}

API 5 : DeleteUser

This method is used to **DeleteUser**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/register/deleteuser/3](http://192.168.1.149:8000/api/register/deleteuser/3%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete selected User |

## Response

If successfully deleted it will return the deleted record Id

3

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

4- CRD API for Upload Document

API 1 : ViewAllDocument

This method is used for **ViewAllDocument**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/document/viewalldocument](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get all Documents | |

## Response

If successful, it will returns all Documents

*For example:-*

## [{"documentid":2,"documentpath":"test1.csv","createdon":"2018-10-03T15:28:19.35","createdby":2},{"documentid":5,"documentpath":"2018\_10\_17\_17\_7\_29\_69\_DemoSchool.sql","createdon":"2018-10-16T12:09:43.03","createdby":3}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : ViewDocumentById

This method is used to **ViewDocumentById**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/document/viewdocumentbyid/2](http://192.168.1.149:8000/api/document/viewdocumentbyid/2%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get Document By Id |

## Response

If successful, it will return detail of Document.

*For Example:-*

## [{"documentid":2,"documentpath":"test1.csv","createdon":"2018-10-03T15:28:19.35","createdby":2}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveDocument

This method is used to **SaveDocument**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/document/savedocument](http://192.168.1.149:8000/api/RegistrationType/SaveRegistrationType)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Save Document |
| file | file | Yes |  |
| CreatedBy | 2 | Yes |  |
| action | add | Yes |  |

## Note: Data will be sent in the form of FormData (multipart/form-data).

## Response

If successful, it will return the id of inserted record.

*For Example:-*

10

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 4 : UpdateDocument

This method is used to **UpdateDocument**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/document/updatedocument](http://192.168.1.149:8000/api/RegistrationType/UpdateRegistrationType)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update Document |
| file | file | Yes |  |
| CreatedBy | 2 | Yes |  |
| action | update | Yes |  |
| DocumentId | 3 | Yes |  |

## Response

If successfully update it will return the updated record Id

3

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

API 5 : DeleteDocument

This method is used to **DeleteDocument**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/document/deletedocument](http://192.168.1.149:8000/api/document/deletedocument%20%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete selected Document |
| action | delete | Yes |  |
| DocumentId | 10 | Yes |  |

## Response

If successfully deleted it will return the deleted record Id

10

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

5- CRUD API for User Activity

API 1 : GetAllActivity

This method is used for **GetAllActivity**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/activity/GetAllActivity](http://192.168.1.149:8000/api/node/getMediaByFolderId?api_token=qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmlkjhgfds&fid=37)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get all Activity | |

## Response

If successful, it will returns all Activities

*For example:-*

## [{"activity\_id":1,"activity":"View"},{"activity\_id":2,"activity":"download"},{"activity\_id":4,"activity":"Data"},{"activity\_id":5,"activity":"view11"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : GetActivityById

This method is used to **GetActivityById**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/activity/GetActivityById/2](http://192.168.1.149:8000/api/document/viewdocumentbyid/2%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get Activity By Id |

## Response

If successful, it will return detail of Activity.

*For Example:-*

## [{"activity\_id":2,"activity":"download"}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveActivity

This method is used to **SaveActivity**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/activity/SaveActivity](http://192.168.1.149:8000/api/RegistrationType/SaveRegistrationType)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Save Activity |
| Activity | View | Yes |  |
| action | add | Yes |  |

## Response

If successful, it will return the id of inserted record.

*For Example:-*

1

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 4 : UpdateActivity

This method is used to **UpdateActivity**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/activity/UpdateActivity](http://192.168.1.149:8000/api/RegistrationType/UpdateRegistrationType)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update Activity |
| Activity | Data123 | Yes |  |
| action | update | Yes |  |
| ActivityId | 2 | Yes |  |

## Response

If successfully update it will return the updated record Id

2

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

API 5 : DeleteActivity

This method is used to **DeleteActivity**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/activity/DeleteActivity](http://192.168.1.149:8000/api/activity/DeleteActivity%20%20%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete selected Activity |
| action | delete | Yes |  |
| ActivityId | 10 | Yes |  |

## Response

If successfully deleted it will return the deleted record Id

10

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

7- CRUD API for FQuestion

API 1 : GetAllFQuestion

This method is used for **GetAllFQuestion**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/FQuestion/getallFQuestion](http://192.168.1.149:8000/api/FQuestion/getallFQuestion%20)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Argument | Example | Required | | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get All Frequent Questions | |

## Response

If successful, it will returns all Frequent Questions.

*For example:-*

## [{"fquestionid":2,"fquestion":"Best","createdon":"2018-10-18T16:09:43.47","createdby":0},{"fquestionid":3,"fquestion":"Rest","createdon":"2018-10-18T16:09:49.98","createdby":0}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 2 : GetFQuestionById

This method is used to **GetFQuestionById**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/FQuestion/getFQuestionbyid/2](http://192.168.1.149:8000/api/FQuestion/getFQuestionbyid/2%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Get Frequent Question By Id |

## Response

If successful, it will return detail of Frequent Question.

*For Example:-*

## [{"fquestionid":2,"fquestion":"Best","createdon":"2018-10-18T16:09:43.47","createdby":0}]

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 3 : SaveFQuestion

This method is used to **SaveFQuestion**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/FQuestion/saveFQuestion](http://192.168.1.149:8000/api/FQuestion/saveFQuestion%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Save Frequent Question |
| FQuestion | Question | Yes |  |
| action | add | Yes |  |
| CreatedBy | 1 | Yes |  |

## Response

If successful, it will return the id of inserted record.

*For Example:-*

2

## Errors

If not successful check token is passed or not.

{"Message":"Authorization has been denied for this request."}

API 4 : UpdateFQuestion

This method is used to **UpdateFQuestion**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/FQuestion/updateFQuestion](http://192.168.1.149:8000/api/FQuestion/updateFQuestion%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Update Frequent Question |
| FQuestion | Question11 | Yes |  |
| action | update | Yes |  |
| CreatedBy | 1 | Yes |  |
| FQuestionId | 2 | Yes |  |

## Response

If successfully update it will return the updated record Id

2

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}

API 5 : DeleteFQuestion

This method is used to **DeleteFQuestion**

## Argument

This method has the URL

[http://192.168.1.149:8000/api/FQuestion/deleteFQuestion](http://192.168.1.149:8000/api/MsgRequestType/deleteMsgRequestType%20%20%20%20)

|  |  |  |  |
| --- | --- | --- | --- |
| Argument | Example | Required | Description |
| access\_token | xxxx-xxxx-xxxx-xxxx | Yes | Delete Frequent Question |
| action | delete | Yes |  |
| FQuestionId | 2 | Yes |  |

## Response

If successfully deleted it will return the deleted record Id

2

## Errors

If not successful check token is passed or not.

For Example :

{"Message":"Authorization has been denied for this request."}