

## Integrating GoToMeeting with Teamchat

### Step-1: CREATING A USER ACCOUNT

Register for a GoToMeeting account at <http://www.gotomeeting.com>. This will be your account for creating and attending online meetings and conferences.

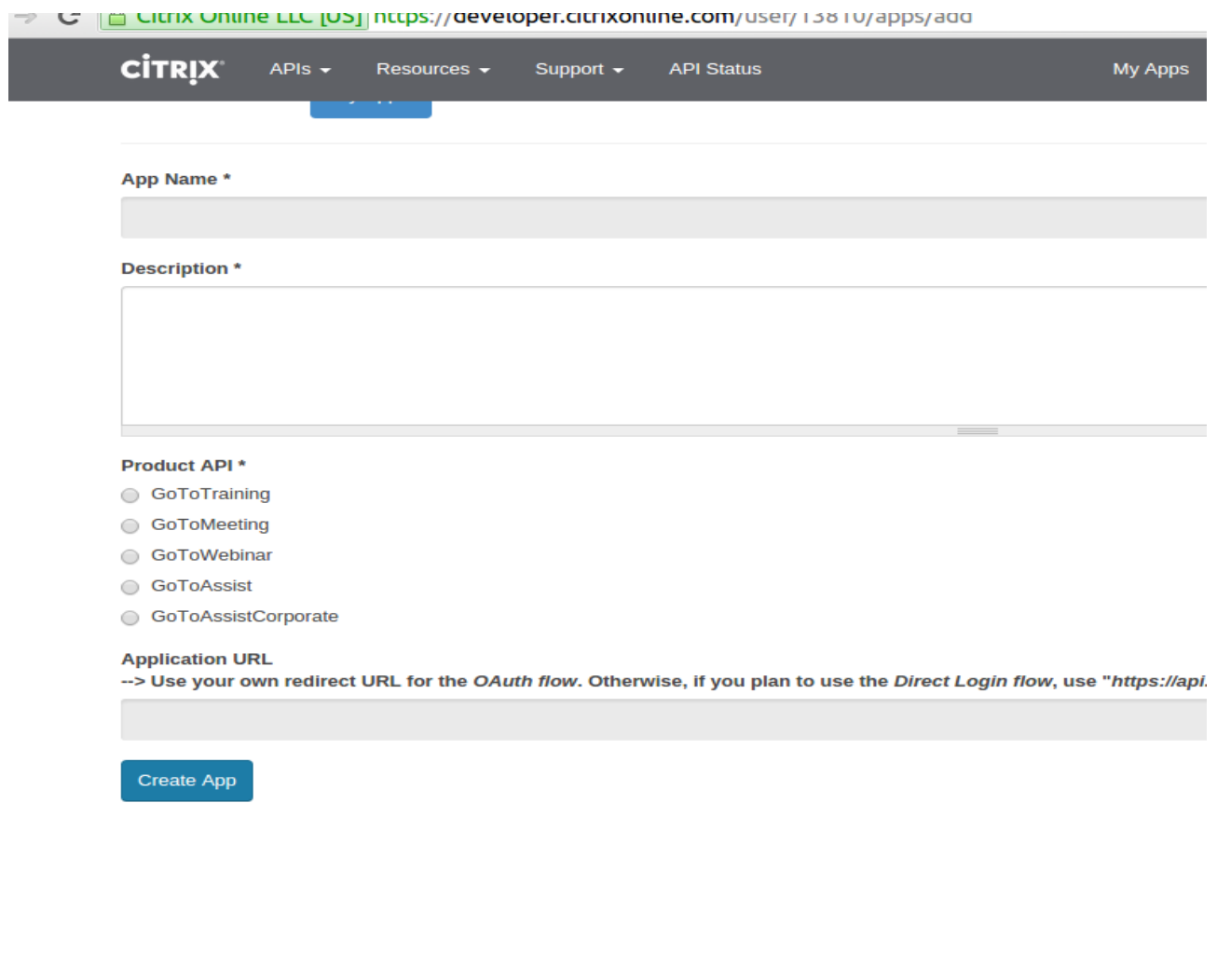
### Step-2: CREATING A DEVELOPER ACCOUNT

Now in order to intergrate GoToMeeting in Teamchat you need to create a developer account at <https://developer.citrixonline.com/>.

**NOTE:** Your user account should be different from your developer account.

### Step-3: CREATING A NEW APP

When you log in using your developer account, at the home page you can see the option “Add a new App”. Click on it, the follwing screen appears:



The screenshot shows the 'Add App' page in the Citrix Online Developer Portal. The browser address bar shows the URL: <https://developer.citrixonline.com/user/13810/apps/add>. The page has a dark header with the Citrix logo and navigation links: APIs, Resources, Support, API Status, and My Apps. The main form contains the following fields and options:

- App Name \***: A text input field.
- Description \***: A large text area for a description.
- Product API \***: A list of radio button options:
  - ☐ GoToTraining
  - ☐ GoToMeeting
  - ☐ GoToWebinar
  - ☐ GoToAssist
  - ☐ GoToAssistCorporate
- Application URL**: A text input field with a note below it: "--> Use your own redirect URL for the OAuth flow. Otherwise, if you plan to use the Direct Login flow, use "https://api".
- Create App**: A blue button at the bottom of the form.

Fill the following details and in the option “Product API”, select GoToMeeting. Redirect URL can be any URL (like [www.google.com](http://www.google.com)) even your localhost URL.

Once the app is created, your app is now assigned a consumer-key and consumer-secret.

These are your apps! Explore them!

TeamMeet	
Keys	Products
Details	Analytics
Edit "TeamMeet"	Delete "TeamMeet"
TeamMeet's Keys	
Consumer Key	I3t1BHRqzL4tuUcdCHPJUULRzCVHGIUr
Consumer Secret	7INDReHEAJKla42L

#### Step-4: FETCHING ACCESS TOKEN

Once you have the client-id you can now fetch the access token using the url:

[https://api.citrixonline.com/oauth/access\\_token?grant\\_type=password&user\\_id=<put\\_developer\\_email\\_here>&password=<put\\_developer\\_password\\_here>&client\\_id=<put\\_consumer\\_key\\_here>](https://api.citrixonline.com/oauth/access_token?grant_type=password&user_id=<put_developer_email_here>&password=<put_developer_password_here>&client_id=<put_consumer_key_here>)

To check the output of this URL, run this URL in your web browser and you will see the following response:

```
1 {
2   "access_token": "K2AQucgycaBIuEnpFKh53rfTOFGD",
3   "expires_in": "30758399",
4   "refresh_token": "ePsB1rbZfg6cS5gUCByzddKalzGcHR28",
5   "organizer_key": "4369960952971322118",
6   "account_key": "4756497564250858246",
7   "account_type": "",
8   "firstName": "Prakhar",
9   "lastName": "Vashisht",
10  "email": "prakhar888.vegmail.com",
11  "platform": "GLOBAL",
12  "version": "2"
13 }
```

The above step is just to give an idea for the output of the given URL.

You need to fetch the access token using a JAVA code with HTTP GET Request.

The ouput is a JSON object. This object is parsed and the value of the required key (i.e. "access\_token") is extracted.

The following JAVA method fetches the access token from the given URL and returns it to the calling function.

```
public String getAccessToken(String email, String pwd,String clientid) throws IOException, JSONException
{
    URL url = new URL("https://api.citrixonline.com/oauth/access_token?
grant_type=password&user_id=" + email + "&password=" + pwd +
"&client_id="+clientid);

    URLConnection yc = url.openConnection();
    BufferedReader in = new BufferedReader(new
InputStreamReader(yc.getInputStream()));

    StringBuilder response = new StringBuilder();
    String inputLine;
    while ((inputLine = in.readLine()) != null)
        response.append(inputLine);

    String out = response.toString();
    in.close();

    JSONObject jobject = new JSONObject(out);
    String accTok = (String) jobject.get("access_token");

    return (accTok);
}
```

### Step-5: USING ACCESS TOKEN TO CALL FUNCTIONS

Once the access token is fetched, it can be used to call various functions provided by GoToMeeting API.

The general information about GoToMeeting API can be found at:

<https://developer.citrixonline.com/gotomeeting-api-overview>

Registration at developer.citrixonline.com generates the following critical values:

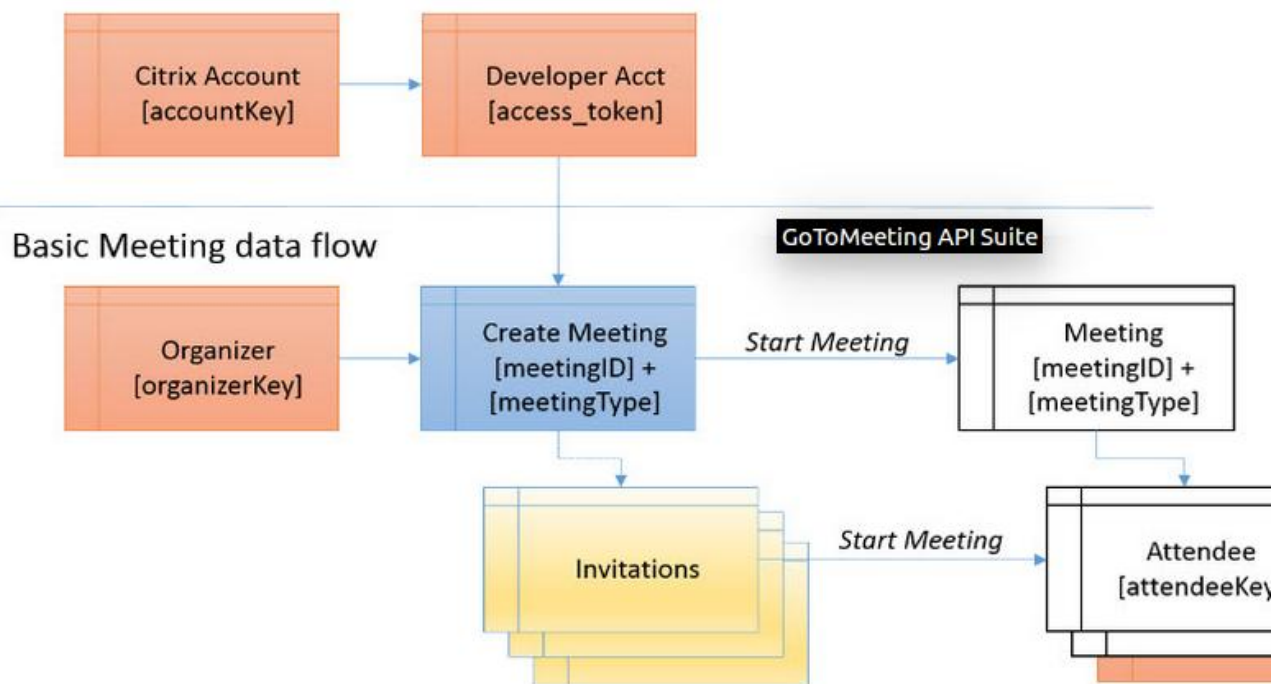
- **access\_token** (AKA oauth\_token) Alphanumeric string used to authorize your requests to Citrix servers. This is passed in every HTTP header for every API call.
- **organizer\_key** (AKA organizerKey) Numeric string that represents your organizer and presenter status and allows you create, edit, delete, and launch Meetings. This is passed in many API request URLs to validate or identify the individual person responsible.
- **account\_key** (AKA accountKey) Numeric string denoting your overall account. This is a numeric string that is passed in request APIs where you are requesting, for instance, a list of all organizers on the account.

The corresponding Data Model for REST API is shown:

## Data model

The underlying GoToMeeting data objects consist of organizers, meetings, meeting types and attendees. Each meeting event is recorded with the meetingType (meet now, scheduled, or recurring), meetingID, and start and end times. Reporting against meeting history provides all past, completed sessions. Reporting on scheduled meetings provides all scheduled and recurring meetings. In general, data values are 19-plus alphanumeric characters with case sensitivity. However, the length and definition of data keys may change at any time.

### Preliminary requirements



Registration at [developer.citrixonline.com](https://developer.citrixonline.com) generates the following critical values:

- **access\_token** (AKA oauth\_token) Alphanumeric string used to authorize your requests to Citrix servers. This is passed in every HTTP header for every API call.
- **organizer\_key** (AKA organizerKey) Numeric string that represents your organizer and presenter status and allows you create, edit, delete, and launch Meetings. This is passed in many API request URLs to validate or identify the individual

Once you complete the authentication flow and have an access token, you can make API calls. You can make the calls from a cURL client, an API client such as Postman or your own code.

### 5.1 GET HISTORICAL MEETINGS:

Gets historical meetings for the organizer which is the owner of the passed access token. History searches will contain the parameter "meetingInstanceKey" which is used in conjunction with the call "Get Attendees by Meeting" to get attendee information for a past meeting.

URL : <https://api.citrixonline.com/G2M/rest/meetings>

Method : GET

Parameters : paramName="Auhorization", value="access\_token"

Output:

```
[ { "lastName": "string",
    "groupName": "string",
    "numAttendees": 0,
    "passwordRequired": true,
    "status": "string",
    "subject": "string",
    "endTime": "2015-06-11T10:44:47.579Z",
    "date": "2015-06-11T10:44:47.579Z",
    "conferenceCallInfo": "string",
    "startTime": "2015-06-11T10:44:47.579Z",
    "organizerkey": "string",
    "meetingInstanceKey": 0,
    "newOrganizerKey": "string",
    "duration": 0,
    "newMeetingId": "string",
    "sessionId": 0,
    "email": "string",
    "meetingId": 0,
    "organizerKey": "string",
    "meetingKey": 0,
    "meetingType": "string",
    "firstName": "string",
    "uniqueMeetingId": 0
} ]
```

Code:

The following code gives a JSON Output which contains information about all previous meetings. You can parse this JSON Output and extract necessary information and display it to the user.

```
public void viewMeetings(TeamchatAPI api, String accTok) throws
ClientProtocolException, IOException, JSONException
{
    HttpClient client = new DefaultHttpClient();
    HttpGet request = new
    HttpGet("https://api.citrixonline.com:443/G2M/rest/meetings");
    request.setHeader("Authorization", accTok);
    HttpResponse response = client.execute(request);
    BufferedReader rd = new BufferedReader(new
    InputStreamReader(response.getEntity().getContent()));
    String line = "";
    StringBuilder sb = new StringBuilder();
    while ((line = rd.readLine()) != null)
        sb.append(line);
    sb.insert(0, "{meetings:");
    sb.insert(sb.length(), "}");
    String output = sb.toString();
    System.out.println(output);
    JSONObject j=new JSONObject(output);
    JSONArray jarray=j.getJSONArray("meetings");
    int length=jarray.length();
    int[] id= new int[length];
    String[] status =new String[length],
        subject =new String[length];

    for(int i=0;i<jarray.length();i++){
        JSONObject j1=jarray.getJSONObject(i);
        id[i]=j1.getInt("meetingid");
        subject[i]=j1.getString("subject");
        status[i]=j1.getString("status");
    }
}
```

## 5.2 Create a Meeting

Create a new meeting based on the parameters specified.

URL : <https://api.citrixonline.com/G2M/rest/meetings>

Method : POST

Parameters :

paramName="Authorization", value="access\_token"

paramName="body", value=

```
{
    "subject": "string",
    "starttime": "2015-06-11T10:44:47.411Z",
    "endtime": "2015-06-11T10:44:47.411Z",
    "passwordrequired": false,
    "conferencecallinfo": "string",
    "timezonekey": "",
    "meetingtype": "immediate" or "scheduled" or
    "recurring"
}
```

Output:

```
[ {
    "joinURL": "string",
    "meetingid": 0,
    "maxParticipants": 0,
    "uniqueMeetingId": 0,
    "conferenceCallInfo": "string"
} ]
```

Code:

The following code creates a meeting using an access token and returns two values i.e. joinURL and meetingID.

```
public String[] createMeeting(TeamchatAPI api, String accTok) throws ClientProtocolException, IOException,
JSONException
{
    HttpClient client = new DefaultHttpClient();
    HttpPost post = new
HttpPost("https://api.citrixonline.com:443/G2M/rest/meetings");
    post.setHeader("Authorization", accTok);
    JSONObject json = new JSONObject();

    json.put("subject", "Report Meeting");
    json.put("starttime", "2015-06-02T08:54:32.156Z");
    json.put("endtime", "2015-06-02T08:54:32.157Z");
    json.put("passwordrequired", false);
    json.put("conferencecallinfo", "VoIP");
    json.put("timezonekey", "");
    json.put("meetingtype", "immediate");

    StringEntity se = new StringEntity(json.toString());
    post.setEntity(se);
    HttpResponse response = client.execute(post);
    BufferedReader rd = new BufferedReader(new

    InputStreamReader(response.getEntity().getContent()));
    StringBuilder sb = new StringBuilder();
    String line = "";
    while ((line = rd.readLine()) != null)
        sb.append(line);
}
```

```

        sb.deleteCharAt(0);
        sb.deleteCharAt(sb.length() - 1);
        String output = sb.toString();
        JSONObject jobject = new JSONObject(output);
        values[0] = (String) jobject.get("joinURL");
        values[1] = jobject.get("meetingid").toString();
        return values;
    }

```

### 5.3 View Meetings by ID

Returns the meeting details for the specified meeting.

URL : <https://api.citrixonline.com/G2M/rest/meetings/{meetingid}>

Method : POST

Parameters : paramName="Auhorization", value="access\_token"  
 paramName="meetingid", value="meetingid"

Ouput:

```

[ {
    "createTime": "2015-06-11T10:44:47.600Z",
    "passwordRequired": true,
    "status": "string",
    "subject": "string",
    "endTime": "2015-06-11T10:44:47.600Z",
    "conferenceCallInfo": "string",
    "startTime": "2015-06-11T10:44:47.600Z",
    "duration": 0,
    "maxParticipants": 0,
    "meetingId": 0,
    "meetingKey": 0,
    "meetingType": "string",
    "uniqueMeetingId": 0
} ]

```

Code:

The following code extracts a given meeting details when the meeting ID is passed in it as an argument.

```

public void getMeetingByID(TeamchatAPI api,String id) throws IOException,
JSONException
{
    System.out.println("check");
    HttpClient client = new DefaultHttpClient();
    HttpGet request = new
    HttpGet("https://api.citrixonline.com:443/G2M/rest/meetings/" + id);
    request.setHeader("Authorization", accTok);
    HttpResponse response = client.execute(request);
    BufferedReader rd = new BufferedReader(new
    InputStreamReader(response.getEntity().getContent()));
    String line = "";
    StringBuilder sb = new StringBuilder();
    while ((line = rd.readLine()) != null)
        sb.append(line);
}

```

```

        sb.deleteCharAt(0);
        sb.deleteCharAt(sb.length() - 1);
        String output = sb.toString();

        JSONObject j = new JSONObject(output);
        String start = (String) j.get("startTime");
        int part = (int) j.get("maxParticipants");
        String status = (String) j.get("status");
        String subject = (String) j.get("subject");
        String meetingType = (String) j.get("meetingType");
        String end = (String) j.get("endTime");
    }

```

## 5.4 Start a Meeting

Returns a host URL that can be used to start a meeting. When this URL is opened in a web browser, the GoToMeeting client will be downloaded and launched and the meeting will start. The end user is not required to login to a client.

URL : <https://api.citrixonline.com/G2M/rest/meetings/{meetingid}/start>  
 Method : GET

Parameters : paramName="Auhorization", value="access\_token"  
 paramName="meetingid", value="meetingid"

Output:

```

{
    "hostURL": "string"
}

```

Code:

The following code can be used to start a meeting for a given meeting-id. This method takes meetingID and access token as arguments and returns the hosting URL which can be used to start a meeting and others can join using the URL extracted in the above example.

```

public String initializeMeeting(TeamchatAPI api, String accTok, String meetingid)
throws ClientProtocolException, IOException,
        URISyntaxException, JSONException

{
    HttpClient client = new DefaultHttpClient();
    HttpGet request = new
    HttpGet("https://api.citrixonline.com/G2M/rest/meetings/" +
    meetingid + "/start");

    request.setHeader("Authorization", accTok);
    HttpResponse response = client.execute(request);
    BufferedReader rd = new BufferedReader(new
    InputStreamReader(response.getEntity().getContent()));
    StringBuilder sb = new StringBuilder();
    String line = "";
    while ((line = rd.readLine()) != null)
        sb.append(line);
    String output = sb.toString();

    JSONObject jobject = new JSONObject(output);
}

```



```
String hostURL = (String) jobject.get("hostURL");
```

```
return hostURL;
```

```
}
```