

Higher Logic Push API Integration

1.0 Overview

This document details the Higher Logic Push API Integration methodology. This type of integration allows the system of record (your CRM) to send updates directly to Higher Logic. This data can be pushed in real-time or done in batches.

Using the Push API methodology is much more efficient than having the Higher Logic system continuously looking for changed records in a CRM and updating them when found.

The integration between your CRM and your Higher Logic Community site drives object creation in the community site from the vast amount of up to date data that you already track in the CRM system. The integration ensures that your members interact with the most up to date member data, boosting member participation, collaboration and retention.

The Push API has two endpoints, one for updating community member information, and one for updating event information for the community calendar. Member information sent via the Push API should be updated on a member's profile almost instantaneously. Event information may take up to 30 minutes to show on the community calendar after being sent via the Push API.

Note: Single Sign-On (SSO) and Activity Sync are not within the scope of the Push API methodology and are done with different processes.

2.0 Push API for Community Member Information

The Push API Endpoint URL for community member information can be found at

<https://data.higherlogic.com/push/v1/members>

The Higher Logic Push API integration has several elements that are passed to Higher Logic to keep community member records current. The sections below described the data that is sent with Higher Logic's Push API method. When the method is called, the full dataset must be populated, and not only the changed values.

Authentication

An API Gateway Key and Tenant Code are needed for the Push API method. The API Gateway Key is generated by Higher Logic and the Tenant Code is specific to the client's Higher Logic's instance. The API Gateway Key provided by Higher Logic should be sent in a header called "x-api-key". The TenantCode is passed with the actual request.

MemberDetails

This section of Higher Logic's Push API method is for **Member Details**. This contains the basic profile demographic fields such a name and contact information. Additional demographic information can be passed in the **Demographics** section of the Push API method. This section has unique elements for a constituent and should only be included once.

Field	Description	Type & Size
LegacyContactKey (required)	The unique ID from the CRM system. Note: The RegEx that describes this field needs to be given to Higher Logic so we know what to expect.	String(100)
MemberID	A readable ID to be show to constituents. It can be the same as the LegacyContactKey.	String(50)
PrefixCode	A name prefix (Mr., Ms., Mrs., Dr., etc.)	String(20)
FirstName	The constituent's first name	String(50)
MiddleName	The constituent's middle name	String(50)
LastName (required for Individuals)	The constituent's last name	String(50)
SuffixCode	A name suffix (Jr., Sr., III, etc.)	String(20)
Designation	A designation or credential included in the constituent's name.	String(50)
InformalName	An informal or nickname for the constituent	String(50)
Gender	Gender of the constituent	String(50)
Ethnicity	Ethnicity of the constituent	String(100)
Age ¹	The age of the constituent.	Integer
Birthday ¹	The birthday of the constituent.	Date-time
MemberSince	The date the constituent joined the organization.	Date-time
MemberExpiresOn	The date the current membership expires.	Date-time

ExcludeFromDirectory	Do not show this record in Directory searches (True or False).	Boolean
IsActive	Is an active record (True or False).	Boolean
IsMember	Receives member benefits (True or False).	Boolean
Title	Job Title	String(150)
Company Name (required for Organizations)	A company name or employer	String(300)
ParentMemberKey	A unique ID from the CRM system for the Parent company or organization	String(100)
Bio	The text of a constituent's biography; HTML allowed.	String(unlimited)
ProfileImageURL	A publicly accessible URL to the constituent's profile picture. Images resized to 200x200.	String(unlimited)
EmailAddress (required)	The primary email address of the constituent.	String(100)
Phone1 ²	A formatted phone number	String(50)
Phone1Type ³	A type or location for Phone1	String(30)
Phone2 ²	A formatted phone number	String(50)
Phone2Type ³	A type or location for Phone2	String(30)
Phone3 ²	A formatted phone number	String(50)
Phone3Type ³	A type or location for Phone3	String(30)
Phone4 ²	A formatted phone number	String(50)
Phone4Type ³	A type or location for Phone4	String(30)
Address1	Address line 1	String(100)
Address2	Address line 2	String(100)
Address3	Address line 3	String(100)
City	City	String(100)
State	State (2-character abbreviation preferred)	String
PostalCode	Postal Code	String(20)
Country	Country (ISO ALPHA-2 & ALPHA-3 are acceptable values)	String
WebsiteURL	Full web site URL	String(255)
YouTubeURL ⁴	Full YouTube profile URL	String(1000)
FacebookURL ⁴	Full Facebook profile URL	String(1000)
TwitterURL ⁴	Full Twitter profile URL	String(1000)
LinkedInURL ⁴	Full LinkedIn profile URL	String(1000)
GooglePlusURL ⁴	Full Google Plus profile URL	String(1000)
WordPressURL ⁴	Full WordPress site URL	String(1000)
BloggerURL ⁴	Full Blogger site URL	String(1000)
OtherBlogURL ⁴	Full other web site URL	String(1000)
IsOrganization	Is a Company or Organization record (True or False). If False , this record is an individual.	Boolean
PrimaryContactLegacyContact Key	A unique ID from the CRM system for a primary contact if this record is an organization.	String(100)

¹Age is not calculated from Birthday by Higher Logic

²Phone numbers are not formatted by Higher Logic

³Phone Types should be consistent across all profiles, ie Phone1Type is always **Home**, Phone2Type is always **Work**

⁴All social profile links should be the full URL. For example, <https://twitter.com/rkaighn> not [@rkaighn](https://twitter.com/rkaighn) or twitter.com/rkaighn

In general, these fields will be displayed on a profile as they were sent over via the Push API. There are three **Configuration Parameters** that must be set by HL Staff to **True** for the related items to be brought into the profile:

- MembershipIntegrateBio
- IntegrateProfileImage
- IntegrateSocialMediaSites

CommunityGroups

This section of Higher Logic’s Push API method is for **Community Groups**. The Community object in your community site drives membership and access to many of the features on the site including Discussion Groups, Libraries and Microsites. Each integrated Community has a unique identifier, display name and limiting logic on the Higher Logic side that controls which users can view and access community content. Only the members of a Community as determined via the integration can view the affiliated Resource Library and Discussion Group.

Communities are driven by Community-Memberships determined and should have a term component to manage who is a member and for what specific period of time. Each Community Membership has a begin date, end date, and community role.

Communities are of a certain Community Type (such as Committees, Chapters, Special Interest Groups, etc.) and integration logic is implemented per-Community Type.

This section has several elements that are repeated for each community group the constituent belongs

Field	Description	Type & Size
SinceDate	The initial date-time when the constituent became part of this group, or join date.	Date-time
BeginDate	The begin date-time of the current membership to this group.	Date-time
EndDate	The end date-time of the current membership to this group. If an end date is not included, this group assignment will not end.	Date-time
GroupKey (required)	A unique id for this group. For example, a chapter or committee ID. This value must not change over time.	String(100)
ParentCommunityKey	If this community group has a parent community group, this is the key to that parent community. This value is not required, but must not change if provided.	String(100)
GroupName (required)	The name of this community group.	String(200)
GroupType	Community Types are the core organizational construct for Communities. Community Type settings	String(100)

	establish defaults for the products supporting Communities, access permissions, display options, and more. These are defined by the client for their tenant.	
GroupSubType	[UNUSED]	String(100)
RoleDescription	A role given to this user in this group.	String(100)

Members of communities can be automatically subscribed to the community’s discussion. Thus when someone becomes a member, starts a committee term, etc., they will start receiving discussion messages based settings for that specific community.

New community groups are created automatically when users with them are pushed to Higher Logic. These groups may still need to be configured in Higher Logic for the correct use, but they will not need to be created. The **GroupType** can be used to determined how this new community will be configured within Higher Logic. For examples, chapters and committees can be sent over and will be created based upon a **GroupType** that is used for all committees and chapters. If the **GroupType** is not defined, they will be created as a hidden community.

RoleDescription can be used to defined who can administer a group. For example, a role can be set for a **ChapterAdministrator**, or **President**. In the Higher Logic Community Type setup, under the *Community Types Linked to the backend system* section, these roles can be designated as administrators for a community.

SecurityGroups

This section of Higher Logic’s Push API method is for **Security Groups**. Security Groups can be thought of as “Roles” and are a powerful tool for limiting viewing and access for navigation and content items on your community site. On your community site, users are broadly batched into Security Groups. Individual users can belong to multiple Security Groups.

Communities have an **IsMember** group by default which will be defined during implementation. Multiple security groups can be made a part of this group.

Organizations with multiple types of membership can represent those types by adding additional Security Groups to the community site, designating **IsMember** status as needed. The most common of these is the “Staff” Security Group for employees of the organization. Others include “Associate Member” or “Student Member.”

Remember, it is not necessary to create Security Groups around any of your Community Groups. By default, only the members of a particular community will be able to access the affiliated Discussion Group and Resource Library. For instance, it is not necessary to create a “Board of Directors” Security Group if the aim is to restrict access to a Discussion Group that is already a feature of the “Board of Directors” Community.

This API section has several elements that are repeated for each security group the constituent belongs:

Field	Description	Type & Size
SinceDate	The initial date-time when the constituent became part of this group, or join date.	Date-time
BeginDate	The begin date-time of the current membership in this group.	Date-time
EndDate	The end date-time of the current membership in this group. If an end date is not included, this group assignment will not end.	Date-time
GroupKey (required)	A unique id for this group. For example, a chapter or committee ID. This value must not change over time.	String(100)
GroupName (required)	The name of this security group.	String(200)
GroupType	Type has no effect on functionality, it is purely a cosmetic way of grouping the different types of groups.	String(100)
RoleDescription	A role given to this user in this group.	String(100)

New security groups are created automatically when users with them are pushed to Higher Logic. These groups may still need to be configured in Higher Logic for the correct use, but they will not need to be created.

Demographics

This section of Higher Logic's Push API method is for **Demographics**. If you capture additional demographics from what is passed in the **MemberDetails** section of the Push API method, those may be passed here. This is ideal for demographics your constituents would use primarily to connect with each other but do not serve an additional organizational purpose. These can be a single value, like eye color, or one-to-many, like languages spoken. They can also be of a fixed value, or free form text.

This API section has several elements that are repeated for each demographic the constituent has:

Field	Description	Type & Size
DemographicTypeKey (required)	Unique identifier for this category (pick) of demographic	String(100)
DemographicTypeValue (required)	Descriptive name for the category	String(100)
DemographicKey (required when IsFreeForm = False)	Unique identifier for the value in the category; only used when demographic is not free form text.	String(100)
DemographicValue (required)	Descriptive name for the value, or specific value if free form text.	String(100)
IsFreeForm	True if this is a free form text field or False if specific pick values are required.	Boolean

For example, if languages spoken by the constituent is needed as a demographic on profiles, the **DemographicTypeKey** could be **LANGUAGE** and the **DemographicTypeValue** could be

Languages Spoken. The **DemographicKey** could be **FRENCH** and the **DemographicValue** could be **French**. Since the demographic options are fixed values (different languages) **IsFreeForm** would be **False**. This would be repeated for other languages spoken by the constituent. Because this is not free form text, Higher Logic will allow selection of the specific demographic values when searching a directory for members with this demographic.

If the demographic is free form text, the **DemographicKey** is ignored and the **DemographicValue** would be the free form text.

JobHistory

This section is for a user’s employment history. This section is only needed if there is employment information that is managed in the CRM. If this data is not stored in the CRM, this data can live independently on the Higher Logic profile.

This API section has several elements that are repeated for each job the constituent had. None of these fields are required.

Field	Description	Type & Size
CompanyName (required)	The name of the company worked for	String(200)
Title	The job title for this employment	String(200)
City	City where the job was located	String(100)
State	State where the job was located	String(100)
Country	Country where the job was located	String(100)
StartDate	Date-time when the employment started	Date-time
EndDate	Date-time when the employment ended	Date-time
WebsiteURL	A full URL to the company website	String(1024)

For the Job History to be brought into a user’s profile, the **MembershipSyncJobHistory Configuration Parameter** that must be set by HL Staff to **True**.

Education

This section is for a user’s education history. This section is only needed if there is education information that is managed in the CRM. If this data is not stored in the CRM, this data can live independently on the Higher Logic profile.

This API section has several elements that are repeated for each education level the constituent has. None of these fields are required.

Field	Description	Type & Size
SchoolName (required)	Name of the Institution	String(400)
Degree	Degree obtained	String(100)
FieldOfStudy	Field of study	String(400)
Dissertation	Dissertation Title	String(1000)
Advisor	Advisor’s name	String(100)

IsHighestDegreeAttained	Is this the highest degree the person has (True or False).	Boolean
City	City where the institution is located	String(100)
State	State where the institution is located	String(8)
Country	Country where the institution is located	String(4)
FromYear	4-digit year when started	String(4)
ToYear	4-digit year when completed	String(4)
DegreeYear	4-digit year when degree obtained	String(4)

For the Education to be bought into a user’s profile, the **MembershipSyncEducation Configuration Parameter** that must be set by HL Staff to **True**.

3.0 Push API for Event Information

Event information can be passed to Higher Logic so it can be displayed within a community's calendar. This information included is not only for display purpose, but will also be included in search results. Detail and registration URLs can also be supplied such that more information and a registration page can be displayed in the community calendar.

The Push API Endpoint URL for event information can be found at

<https://data.higherlogic.com/push/v1/events>

The Higher Logic Push API integration has several elements that are passed to Higher Logic to keep community member records current.

The sections below described the data that is sent with Higher Logic's Push API method. When the method is called, the full dataset must be populated, and not only the changed values. Event information may take up to 30 minutes to show on the community calendar after being sent via the Push API.

Authentication

An API Gateway Key and Tenant Code are needed for the Push API method. The API Gateway Key is generated by Higher Logic and the Tenant Code is specific to the client's Higher Logic's instance. The API Gateway Key provided by Higher Logic should be sent in a header called "x-api-key". The TenantCode is passed with the actual request.

Field	Description	Type & Size
MeetingId (required)	A unique ID for the event from the CRM system	String(100)
Title (required)	Title of the event.	String(100)
Description	Large description of the event that can contain HTML.	String(unlimited)
Type (required)	Event Type (possible values defined by client in their Tenant).	String(50)
BeginDate (required)	The start date-time of the event	Date-time
EndDate	The end date-time of the event	Date-time
TimeZoneCode (required)	Time zone where the event takes place (based on Microsoft Time Names)	WindowsID
IsActive	Denotes an active event (True or False).	Boolean
GroupKey	If this event is limited to a particular Community Group, this is the GroupKey that was defined for that Community Group in the community member Push API method.	String(100)
Address1	Address line 1 for the event location	String(100)
Address2	Address line 2 for the event location	String(100)
Address3	Address line 3 for the event location	String(100)
City	City for the event location	String(100)
State	State for the event location (2-character abbreviation preferred)	String

PostalCode	Postal Code for the event location	String(20)
Country	Country (ISO ALPHA-2 & ALPHA-2 are acceptable values)	String
RegistrationUrl	A full URL when end users may register for the event.	String(1024)
DetailsUrl	A full URL where more information can be found on the event.	String(1024)

Frequently Asked Questions

Q: For date fields, it looks like the examples are sending date-time data. Do we need to add a time element to date fields if one does not exist?

A: Time is not required and if not set will be set to 00:00:00.000 AM

Q: For the Phone fields, it says "formatted phone number". Are we supposed to send over the phone number already formatted? Is there a format we need to follow?

A: Higher Logic does not have specific phone formats. This is usually defined by the organization in their CRM and we will accept that as a text field.

Q: Are these phone type values in Higher Logic configurable? In the CRM we have: business phone, home phone, mobile phone, etc.

A: Higher Logic does not have specific phone types defined, so you can use the types defined in the CRM. Please note that they should be consistent across all records. For example, if **Home** is being put in PhoneType1, **Home** should be PhoneType1 for all the records passed. And if **Work** is being put in PhoneType2, **Work** should be PhoneType2 for all the records passed.

Q: What if we don't have states/countries stored as abbreviations/codes?

A: You can pass the values you have and the Push API will attempt to convert them to the corresponding abbreviations/codes.

Appendix A- Examples

Example Community Member Post Request

```
{
  "TenantCode" : "HLTENANT",
  "Items" : [{
    "CommunityGroups" : [{
      "SinceDate" : "",
      "BeginDate" : "9/25/2014 12:00:00 PM",
      "EndDate" : "1/1/0001 12:00:00 AM",
      "GroupKey" : "RepDiscussion",
      "ParentCommunityKey" : "",
      "GroupName" : "National Discussion",
      "GroupType" : "HLTENANT ",
      "GroupSubType" : "",
      "RoleDescription" : "Member"
    }
  ],
  "SecurityGroups" : [{
    "SinceDate" : "1/1/1992 12:00:00 PM",
    "BeginDate" : "1/1/1992 12:00:00 PM",
    "EndDate" : "1/1/0001 12:00:00 AM",
    "GroupKey" : "active",
    "ParentCommunityKey" : "",
    "GroupName" : "Active",
    "GroupType" : "HLTENANT",
    "GroupSubType" : "",
    "RoleDescription" : null
  }, {
    "SinceDate" : "1/1/0001 12:00:00 AM",
    "BeginDate" : "1/1/0001 12:00:00 AM",
    "EndDate" : "12/31/9999 11:59:59 PM",
    "GroupKey" : "",
    "ParentCommunityKey" : "",
    "GroupName" : "Authenticated",
    "GroupType" : "Security",
    "GroupSubType" : "",
    "RoleDescription" : null
  }, {
    "SinceDate" : "1/1/0001 12:00:00 AM",
    "BeginDate" : "1/1/0001 12:00:00 AM",
    "EndDate" : "12/31/9999 11:59:59 PM",
    "GroupKey" : "",
    "ParentCommunityKey" : "",
    "GroupName" : "Public",
    "GroupType" : "Security",
    "GroupSubType" : "",
    "RoleDescription" : null
  }
  ],
  "Demographics" : [{
    "DemographicKey" : "Administrative",
    "DemographicValue" : "Administrative",
    "DemographicTypeKey" : "membertype",
    "DemographicTypeValue" : "MemberType",
    "IsFreeForm" : "False"
  }, {
    "DemographicKey" : "No",
    "DemographicValue" : "No",
```

```

        "DemographicTypeKey" : "cpgmember",
        "DemographicTypeValue" : "CPG Member",
        "IsFreeForm" : "False"
    }, {
        "DemographicKey" : "No",
        "DemographicValue" : "No",
        "DemographicTypeKey" : "issupervisor",
        "DemographicTypeValue" : "Is Supervisor",
        "IsFreeForm" : "False"
    }, {
        "DemographicKey" : "",
        "DemographicValue" : "04/07/2014",
        "DemographicTypeKey" : "datejoinedcambridge",
        "DemographicTypeValue" : "Date Joined Cambridge",
        "IsFreeForm" : "True"
    }, {
        "DemographicKey" : "",
        "DemographicValue" : "07, 24, 63",
        "DemographicTypeKey" : "industrylicenses",
        "DemographicTypeValue" : "Industry Licenses",
        "IsFreeForm" : "True"
    }
},
"MemberDetails" : {
    "LegacyContactKey" : "123456789",
    "Age" : 55,
    "Birthday" : "4/14/1961 12:00:00 AM",
    "MemberSince" : "1/1/0001 12:00:00 AM",
    "MemberExpiresOn" : "1/1/0001 12:00:00 AM",
    "Designation" : "",
    "EmailAddress" : "",
    "PrefixCode" : "",
    "FirstName" : "ROBERT",
    "MiddleName" : "D",
    "LastName" : "SMITH",
    "InformalName" : "BOB",
    "Gender" : "M",
    "Ethnicity" : "",
    "SuffixCode" : "",
    "CompanyName" : "",
    "Title" : "",
    "ParentMemberKey" : "",
    "PrimaryContactLegacyContactKey" : "",
    "ExcludeFromDirectory" : "False",
    "IsActive" : "True",
    "IsOrganization" : "False",
    "MemberId" : "",
    "Bio" : "",
    "WebsiteUrl" : "",
    "ProfileImageUrl" :
    "Phone1" : "",
    "Phone1Type" : "OFFICE",
    "Phone2" : "",
    "Phone2Type" : "MOBILE",
    "Phone3" : "",
    "Phone3Type" : "FAX",
    "Phone4" : "",
    "Phone4Type" : "",
    "Address1" : "1600 Wilson Blvd Ste 400",
    "Address2" : "",

```

```

        "Address3" : "",
        "City" : "Arlington",
        "State" : "VA",
        "PostalCode" : "22209",
        "Country" : "",
        "YouTubeURL" : "",
        "FacebookURL" : "",
        "TwitterURL" : "",
        "GooglePlusURL" : "",
        "LinkedInURL" : "",
        "BloggerURL" : "",
        "WordPressURL" : "",
        "OtherBlogURL" : ""
    }
}, {
    "CommunityGroups" : [],
    "SecurityGroups" : [{
        "SinceDate" : "1/1/0001 12:00:00 AM",
        "BeginDate" : "1/1/0001 12:00:00 AM",
        "EndDate" : "12/31/9999 11:59:59 PM",
        "GroupKey" : "",
        "ParentCommunityKey" : "",
        "GroupName" : "Authenticated",
        "GroupType" : "Security",
        "GroupSubType" : "",
        "RoleDescription" : null
    }, {
        "SinceDate" : "1/1/0001 12:00:00 AM",
        "BeginDate" : "1/1/0001 12:00:00 AM",
        "EndDate" : "12/31/9999 11:59:59 PM",
        "GroupKey" : "",
        "ParentCommunityKey" : "",
        "GroupName" : "Public",
        "GroupType" : "Security",
        "GroupSubType" : "",
        "RoleDescription" : null
    }
    ],
    "Demographics" : [{
        "DemographicKey" : "HARRISONMKENNARD",
        "DemographicValue" : "HARRISON M KENNARD",
        "DemographicTypeKey" : "branchmanager",
        "DemographicTypeValue" : "Branch Manager",
        "IsFreeForm" : "False"
    }, {
        "DemographicKey" : "No",
        "DemographicValue" : "No",
        "DemographicTypeKey" : "issupervisor",
        "DemographicTypeValue" : "Is Supervisor",
        "IsFreeForm" : "False"
    }, {
        "DemographicKey" : "HomeOffice",
        "DemographicValue" : "Home Office",
        "DemographicTypeKey" : "membertype",
        "DemographicTypeValue" : "MemberType",
        "IsFreeForm" : "False"
    }, {
        "DemographicKey" : "No",
        "DemographicValue" : "No",
        "DemographicTypeKey" : "cpgmember",
    }
    ]
}

```

```

        "DemographicTypeValue" : "CPG Member",
        "IsFreeForm" : "False"
    }
},
"Education": [{
    "SchoolName": "University of Phoenix",
    "State": "",
    "Country": "",
    "Degree": "",
    "FieldOfStudy": "",
    "FromYear": "2009",
    "ToYear": "2012",
    "IsHighestDegreeAttained": "False",
    "City": "",
    "DegreeYear": "0",
    "Advisor": "",
    "Dissertation": ""
},
{
    "SchoolName": "ASU, Carey School of Business",
    "State": "AZ",
    "Country": "US",
    "Degree": "",
    "FieldOfStudy": "",
    "FromYear": "2008",
    "ToYear": "2009",
    "IsHighestDegreeAttained": "False",
    "City": "",
    "DegreeYear": "0",
    "Advisor": "",
    "Dissertation": ""
},
{
    "SchoolName": "Indiana University",
    "State": "IN",
    "Country": "US",
    "Degree": "",
    "FieldOfStudy": "",
    "FromYear": "1986",
    "ToYear": "1990",
    "IsHighestDegreeAttained": "False",
    "City": "Bloomington",
    "DegreeYear": "0",
    "Advisor": "",
    "Dissertation": ""
}],
"JobHistory": [{
    "CompanyName": "",
    "Title": "VP, Financial Advisor",
    "City": "Scottsdale",
    "State": "Arizona",
    "Country": "",
    "StartDate": "5/1/2007 12:00:00 AM",
    "EndDate": "4/1/2009 12:00:00 AM",
    "WebsiteUrl": ""
},
{
    "CompanyName": "",
    "Title": "VP, Trust Division",
    "City": "Chicago",

```

```

        "State": "Illinois",
        "Country": "",
        "StartDate": "4/1/2009 12:00:00 AM",
        "EndDate": "4/1/2010 12:00:00 AM",
        "WebsiteUrl": ""
    },
    {
        "CompanyName": "",
        "Title": "Senior Director, Former Player Services",
        "City": "Washington DC",
        "State": "District Of Columbia",
        "Country": "",
        "StartDate": "8/1/2010 12:00:00 AM",
        "EndDate": "1/1/1753 12:00:00 AM",
        "WebsiteUrl": ""
    }
}],
"MemberDetails": {
    "LegacyContactKey": "246813579",
    "Age": 26,
    "Birthday": "4/17/1989 12:00:00 AM",
    "MemberSince": "1/1/0001 12:00:00 AM",
    "MemberExpiresOn": "1/1/0001 12:00:00 AM",
    "Designation": "",
    "EmailAddress": "",
    "PrefixCode": "",
    "FirstName": "John",
    "MiddleName": "M.",
    "LastName": "Public",
    "InformalName": "",
    "Gender": "M",
    "Ethnicity": "",
    "SuffixCode": "",
    "CompanyName": "",
    "Title": "",
    "ParentMemberKey": "",
    "PrimaryContactLegacyContactKey": "",
    "ExcludeFromDirectory": "False",
    "IsActive": "False",
    "IsOrganization": "False",
    "MemberId": "",
    "Bio": "",
    "WebsiteUrl": "",
    "ProfileImageUrl": "https://cloudfront.net/Profile50.png",
    "Phone1": "(202) 360-4402",
    "Phone1Type": "OFFICE",
    "Phone2": "(202) 555-1212",
    "Phone2Type": "MOBILE",
    "Phone3": "(202) 123-4563",
    "Phone3Type": "FAX",
    "Phone4": "",
    "Phone4Type": "",
    "Address1": "200 Pauline St",
    "Address2": "Ste 15a",
    "Address3": "",
    "City": "",
    "State": "",
    "PostalCode": "",
    "Country": "",
    "YouTubeURL": "",
    "FacebookURL": ""
}

```



```

        "TwitterURL" : "",
        "GooglePlusURL" : "",
        "LinkedInURL" : "",
        "BloggerURL" : "",
        "WordPressURL" : "",
        "OtherBlogURL" : ""
    }
}
}}

```

Example Event Post Request

```

{
  "TenantCode" : "MARCOTESTMSS",
  "Items" : [{
    "MeetingId" : "78542",
    "Title" : "2016 Annual Convention: The Big Show in Vegas",
    "ShortTitle" : "2016 Annual Convention",
    "Description" : "Annual Convention for all practitioners 2016",
    "Type" : "ANNUAL",
    "BeginDate" : "2016-07-03T09:00:00",
    "EndDate" : "2016-07-08T17:00:00",
    "TimeZoneCode" : "US Eastern Standard Time",
    "IsActive" : "1",
    "Address1" : "The Casino",
    "Address2" : "100 Main St.",
    "Address3" : "Ballroom 3",
    "City" : "Las Vegas",
    "State" : "NV",
    "PostalCode" : "12456",
    "Country" : "USA",
    "RegistrationUrl" : "http://www.association.com/register.aspx?id=78542",
    "DetailsUrl" : "http://www.association.com/calendar/78542"
  }, {
    "MeetingId" : "36519",
    "Title" : "October Board Meeting",
    "ShortTitle" : "October Board Meeting",
    "Type" : "ONLINE",
    "BeginDate" : "2016-10-01T10:00:00",
    "EndDate" : "2016-10-01T18:00:00",
    "TimeZoneCode" : "Central Standard Time",
    "IsActive" : "0",
    "GroupKey" : "formscommunity"
  }
]
}
}}

```