



Arena API Documentation

Version 4.0.2012.2

The Arena RESTful API provides Person, Security, Groups, Promotions, Profiles, and Contributions endpoints enabling the client to view, add, and update data via given parameters.

Table of Contents

GET Methods Endpoints 2

POST Methods Endpoints 3

GET Methods..... 3

 PERSON 3

 SECURITY 7

 GROUPS..... 8

 PROMOTIONS..... 10

 PROFILES 11

 CONTRIBUTIONS 13

POST Methods..... 16

 PERSON 17

 GROUPS..... 20

 PROFILES 21

 CONTRIBUTIONS 22

GET Methods Endpoints

<u>PERSON</u> person/{id} person/list person/{id}/group/list?categoryid={categoryid} person/{id}/attribute/list person/{id}/note/list person/{id}/group/subscribedlist family/{id} attribute/list?group={group}	<u>SECURITY</u> security/template/list security/objecttype/list
<u>GROUPS</u> group/list?categoryid={categoryid} group/{id} group/{id}/member/list group/{groupid}/member/{personid} category/list category/{categoryid}/leaderlist	<u>PROMOTIONS</u> promotion/list
<u>PROFILES</u> profile/list profile/{parentprofile}/list profile/{profile} profile/{profile}/member/list profile/{profileid}/member/{personid}	<u>CONTRIBUTIONS</u> batchtype/list batch/list batch/{batchid} batch/{batchid}/contribution/list contribution/list contribution/{contributionid} fund/list fund/{fundid} project/list project/{projectid}

POST Methods Endpoints

<u>PERSON</u> person/{id}/update person/add person/{id}/picture?extension={extension} person/{id}/note	<u>GROUP</u> group/{groupid}/member/{personid}/update group/{groupid}/member/{personid}/add
<u>PROFILE</u> profile/{parentid}/profile/add profile/add profile/{id}/update profile/{id}/member/{personid}/update profile/{id}/member/{personid}/add	<u>CONTRIBUTIONS</u> contribution/bulkadd batch/{id}/contribution/add batch/add batch/{id}/update

GET Methods

GET help

Displays the service documentation.

GET version

Returns the API version. Clients should check to verify that the API version is compatible with the client.

PERSON:

GET Person

Returns the person that matches the given person ID or GUID.

Endpoint:

person/{id}

Additionally, you can pass a comma delimited list of fields to include in the result. For example, if you pass "PersonID,FirstName,LastName" then only those fields will be returned in the result.

Example

person/{id}?fields={fields}

person/542?fields=PersonID,FirstName,LastName

Returnable Fields:

ActiveMeter, **Addresses** (AddressID, AddressTypeID, AddressTypeValue, City, Country, Latitude, Longitude, State, PostalCode, Primary, Proximity, StreetLine1, StreetLine2), Age, AnniversaryDate, ArealID, AreaName, AttributesLink, BirthDate, BlobID, BlobLink, CampusID, CampusName, ContributeIndividually, DateCreated, DateModified, DisplayNotesCount, EmailStatement, **Emails** (Address), EnvelopeNumber, FamilyID, FamilyLink, FamilyMemberRoleID, FamilyMemberRoleValue, FamilyMembersCount, FamilyID, FamilyName, FirstName, ForeignKey, ForeignKey2, FullName, Gender, GivingUnitID, GraduationDate, Grade, InactiveReasonID, InactiveReasonValue, IncludeOnEnvelope, LastName, MaritalStatusID, MaritalStatusValue, MedicalInformation, MemberStatusID, MemberStatusValue, MiddleName, NickName, Notes, NotesLink, OrganizationID, PersonGUID, PersonID, PersonLink, **Phones** (Extension, Number, PhoneTypeID, PhoneTypeValue, SMSEnabled, Unlisted), PrintStatement, RecordStatusID, RecordStatusValue, RegionName, SuffixID, SuffixValue, TitleID, TitleValue

GET Person List

Returns a list of people that matches the given parameters. At least one query string parameter is required.

Endpoint:

person/list

Additionally, you can pass the following:

- A comma delimited list of fields to include in the results.
- A sorting field (via the sortField parameter). Valid values are FirstName, NickName, LastName, and Address/Address/Proximity.
- A sortDirection parameter (if you include a sortField). Valid values are ASC and DESC.

Optional Parameters (must supply at least one):

Address, altID, arealID, birthdate, email, giftID, loginID, firstName, lastName, personID, phone, profileID, onlyConnected, searchDistance, latitude, longitude, campusID, includeInactive, memberStatus, attributeID, attributeIntValue, attributeVarcharValue, attributeDateTimeValue, attributeDecimalValue, name

Additional query modifiers:

fields, sortField, sortDirection

Example:

person/list?personid=542

Returnable Fields:

[See the returnable fields for the Get Person method.](#)

GET Person Attribute List

Returns person attributes for the given person. Additionally, you can pass an attribute group ID to filter the list.

Endpoint:

person/{id}/attribute/list

Example:

person/542/attribute/list?group=15

Returnable Fields:

AttributeGroupID, AttributeGroupName, AttributeID, AttributeName, DisplayValue, IntValue

GET Person Group List

Returns a list of small groups to which the person matching the given ID belongs. The given ID can be either the person ID or person GUID. **Note:** The category ID is required.

Endpoint:

person/{id}/group/list?categoryid={categoryid}

Example:

person/542/group/list?categoryid=1

Returnable Fields:

Active, AreaID, AreaName, AverageAge, Description, GroupClusterID, GroupClusterTitle, GroupID, GroupTypeID, GroupTypeValue, GroupUrl, Guid, HasMeetingEndTime, HasMeetingStartTime, ImageBlobLink, LeaderID, LeaderLink, MaxMembers, MeetingDayID, MeetingDayValue, MeetingEndTime, MeetingStartTime, MemberLink, Name, Notes, PictureUrl, PrimaryAgeID, PrimaryAgeValue, PrimaryMaritalStatusID, PrimaryMaritalStatusValue, Schedule, **TargetLocation** (AddressID, AddressTypeID, AddressTypeValue, City, Country, Latitude, Longitude, State, PostalCode, Primary, Proximity, StreetLine1, StreetLine2), TargetLocationID, Title, TopicID, TopicValue

GET Person Group Subscribed List

Returns a list of small groups to which the person matching the given ID is subscribed. The given ID can be either the person ID or person GUID.

Endpoint:

person/{id}/group/subscribedlist

Example:

person/542/group/subscribedlist

Returnable Fields:

[See the returnable fields for the Get Person Group List.](#)

GET Person Notes

Returns person notes. Additionally, you can pass a zero-based start index and maximum number of notes to return. If you pass “display=true” it will only return display notes.

Endpoint:

person/{id}/note/list

Example:

person/1/note/list?start=0&max=2&display=false

Returnable Fields:

Max, **PersonNotes** (AuthorID, AuthorName, DateCreated, DateModified, Display, NoteID, Private, SecurityTemplateID, Text), Start, Total

GET Family

Returns family information. Additionally, you can pass a comma delimited list of fields to display for FamilyMembers.

Endpoint:

family/{id}

Example:

family/1?fields=PersonID,FirstName,LastName

Returnable Fields:

FamilyID, **FamilyMembers** (FirstName, FullName, LastName, NickName, PersonID, PersonLink), FamilyName

GET Attributes

Returns person attributes. Additionally, you can pass a group ID to filter the list.

Endpoint:

attribute/list

Example:

attribute/list?group=15

Returnable Fields:

[See returnable fields for the Get Person Attribute List method.](#)

GET Me

Returns person information for the user that is logged in.

Endpoint:

me

Additionally, you can pass a comma delimited list of fields to include in the result.

Example:

me/?fields=PersonID,FirstName,LastName

Returnable Fields:

[See the returnable fields for the Get Person method.](#)

SECURITY

GET Security Object Types

Returns a list of security object types.

Endpoint:

security/objecttype/list

Returnable Fields:

ObjectType

GET Security Templates

Returns a list of security templates for the given type name. You need to obtain the list of type names from security/objecttype/list.

Endpoint:

security/template/list

Example:

security/template/list?type=Application

Returnable Fields:

Description, Name, ObjectType, SecurityTemplateID

GROUPS

GET Group

Returns the small group that matches the given ID. The ID can either be the group ID or group GUID.

Endpoint:

group/{id}

Example:

group/1

Returnable Fields:

Active, AreaID, AreaName, AverageAge, Description, GroupClusterID, GroupClusterTitle, GroupID, GroupTypeID, GroupTypeValue, GroupUrl, Guid, HasMeetingStartTime, HasMeetingEndTime, ImageBlobLink, LeaderID, LeaderLink, MaxMembers, MeetingDayID, MeetingDayValue, MeetingEndTime, MeetingStartTime, MemberLink, Name, Notes, PictureUrl, PrimaryAgeID, PrimaryAgeValue, PrimaryMaritalStatusID, PrimaryMaritalStatusValue, Private, Schedule, TargetLocationID, Title, TopicID, TopicValue, **TargetLocation** (AddressID, AddressTypeID, AddressTypeValue, City, Country, Latitude, Longitude, State, PostalCode, Primary, Proximity, StreetLine1, StreetLine2)

GET Group List

Returns a list of small groups filtered by the given information.

Note: The category ID is required.

Endpoint:

group/list?categoryid={categoryid}

Optional parameters:

XAxis, YAxis, ZAxis, GroupTopic, UseGroupType, GroupType, MeetingDay1, MeetingDay2, AgeGroup, MaritalPreference, Distance, Areaid, SearchText, LimitSearchResultsClusterID, activeOnly, spotsAvailable

Example:

group/list?categoryid=1

Returnable Fields:

[See the returnable fields for the Get Group method.](#)

GET Group Member List

Returns a list of members in the small group with the given group ID. The ID can either be group ID or group GUID.

Additionally, you can pass a comma delimited list of fields to include in the result.

Endpoint:

group/{id}/member/list

Optional parameters:

roleid

Additional query modifiers:

fields

Example:

group/1/member/list?roleid=24&fields=PersonID,FirstName,LastName

Returnable Fields:

Active, DateInactive, DateJoined, GroupID, MemberNotes, PersonID, **PersonInformation** ([See the returnable fields for the Get Person method](#)), PersonLink, RoleID, RoleValue, UniformNumber

GET Group Member

Returns small group member information for the given person ID in the given group ID. Additionally, you can pass a comma delimited list of fields to include in the result.

Endpoint:

group/{groupid}/member/{personid}

Example:

group/1/member/2?fields=PersonID,FirstName,LastName

Returnable Fields:

See the returnable fields for the [Get Person](#) and [Get Group](#) methods.

GET Category List

Returns the list of categories.

Endpoint:

category/list

Returnable Fields:

AgeGroupCaption, AllowRegistrations, CategoryID, CategoryName, CreditAsSmallGroup, DescriptionCaption, GroupLink, GroupPageID, LeaderCaption, LocationTargetCaption, MaritalPreferenceCaption, MaximumMembersCaption, MeetingDayCaption, MeetingEndTimeCaption, MeetingStartTimeCaption, NameCaption, NotesCaption, ParentCaption, PictureCaption, ScheduleCaption, TopicCaption, TypeCaption, UrlCaption

GET Category Leader List

Returns group and person information for leaders in a given category ID. Additionally, you can pass a comma delimited list of person fields to include in the result.

Endpoint:

category/{categoryid}/leaderlist

Example:

category/1/leaderlist

Returnable Fields:

See the returnable fields for the [Get Person](#) and [Get Group](#) methods.

PROMOTIONS

GET Promotions

Returns a list of promotions that matches the given criteria.

Endpoint:

promotion/list

Additional query modifiers:

topicAreasList, areaFilter, campusId, maxItems, eventsOnly, documentTypeId

Note: TopicAreasList is required and must be an integer. The areaFilter must be either “primary”, “secondary” or “both.”

Returnable Fields:

CampusID, CampusName, ContactEmail, ContactName, ContactPhone, CreatedBy, DateCreated, DateModified, DetailLink, EventID, GUID, ModifiedBy, Priority, PromotionRequestID, Title, TopicAreaId, TopicAreaValue, WebApprovedBy, WebApprovedDate, WebApprovalRequired, WebDisplayOnHome, WebEndDate, WebExternalLink, WebImageBlobID, WebImageBlobLink, WebPromote, WebStartDate, WebSummary, WebSummaryImageBlobID, WebSummaryImageBlobLink, WebText

PROFILES

GET Profile

Returns a profile with the given ID. If the profile type is Serving, the **ServingInformation** section will be included in the response.

Endpoint:

profile/{profileid}

Example:

profile/541

Returnable Fields:

Active, ActiveBranch, ActiveMembers, BlobLink, CampusID, CampusName, CategoryLevel, ChildProfilesLink, CreatedBy, CriticalMembers, DateCreated, DateModified, DormantMembers, EndTime, Guid, HasEndTime, HasStartTime, HierarchyLevel, InReviewMembers, MembersLink, ModifiedBy, Name, NoContactMembers, Notes, OrganizationID, OwnerID, OwnerLink, OwnerRelationshipStrength, ParentProfileID, ParentProfileLink, PendingMembers, PeerRelationshipStrength, ProfileActiveMemberCount, ProfileID, ProfileMemberCount, ProfileTypeID, ProfileTypeValue, QualifierID, QualifierValue, **Serving Information** (ContactEmail, ContactInfo, CriticalNeed, DefaultHoursPerWeek, Details, DisplayPublic, ExperienceSkills, ScheduleNotes, ServingBlobID, ServingBlobLink, VideoLink, VolunteersNeeded), StartTime, Summary, Title, TotalMembers

GET Profile List

Returns a list of profiles of the given profile type. If the profile type is Serving, the **ServingInformation** section will be included in the response. Additionally, you can pass a profile type ID to filter the list. **Note:** You must supply the profile type ID number NOT the profile type name.

Endpoint:

profile/list

Example:

profile/list?profiletype=2

Returnable Fields:

[See the returnable fields for the Get Profile method.](#)

GET Parent Profile List

Returns a list of profiles with the parent of the given profile ID.

Endpoint:

profile/{parentprofile}/list

Example:

profile/2283/list

Returnable Fields:

[See the returnable fields for the Get Profile method.](#)

GET Profile Member

Returns Profile member information for the given profile and person IDs. Additionally, you can pass a comma delimited list of fields to include in the result.

Endpoint:

profile/{profileid}/member/{personid}

Example:

profile/541/member/1?fields=PersonID,FirstName,LastName

Returnable Fields:

AttendanceCount, DateActive, DateDormant, DateInReview, DatePending, MemberNotes, **Person Information** ([see the returnable fields for the Get Person method](#)), ProfileID, ProfileLink, SourceID, SourceValue, StatusID, StatusReason, StatusValue

GET Profile Member List

Returns a list of members for a Profile with the given profile ID. Additionally, you can pass a comma delimited list of fields to include in the result.

Endpoint:

profile/{profileid}/member/list

Example:

profile/1/member/list?fields=PersonID,FirstName,LastName

Returnable Fields:

[See the returnable fields for the Get Profile Member method.](#)

CONTRIBUTIONS

GET Batch

Returns the batch that matches the given batch ID.

Endpoint:

batch/{batchid}

Example:

batch/9

Returnable Fields:

BatchDate, BatchDateEnd, BatchId, BatchName, BatchTypeName, Finalized, Owner, VerifyAmount

GET Batch List

Returns a list of batches. Additionally, you can pass query string parameters to filter the list.

Endpoint:

batch/list

Optional Parameters:

Finalized, StartDate, EndDate, BatchName, Owner

Example:

batch/list?finalized=1&startdate=2012-11-04

Returnable Fields:

[See the returnable fields for the Get Batch method.](#)

GET Batch Type List

Returns a list of batch types.

Endpoint:

batchtype/list

Returnable Fields:

Name, TypeID

GET Fund

Returns the fund that matches the given fund ID.

Endpoint:

fund/{fundid}

Example:

fund/1

Returnable Fields:

Active, CanPledge, EndDate, FundDescription, FundId, FundName, FundOrder, OnlineName, StartDate, TaxDeductible

GET Fund List

Returns a list of funds.

Endpoint:

fund/list

Returnable Fields:

[See the returnable fields for the GET Fund method.](#)

GET Project

Returns the project that matches the given project ID.

Endpoint:

project/{projectid}

Example:

project/1

Returnable Fields:

Code, Description, Name, ProjectId

GET Project List

Returns a list of projects.

Endpoint:

project/list

Returnable Fields:

[See the returnable fields for the GET Project method.](#)

GET Contribution

Returns contribution information for a given contribution ID.

Endpoint:

contribution/{contributionid}

Returnable Fields:

BatchId, BatchLink, ContributionDate, **ContributionFunds** (Amount, ContributionFundId, ContributionId, **Fund** ([see the returnable fields for the GET Fund method](#)), FundId, NonCash, **Project** ([see the returnable fields for the GET Project method](#)), ProjectId), CurrencyTypeId, CurrencyTypeValue, CurrencyAmount, ImageBackLink, ImageEnvelopeLink, ImageLink, Memo, **PersonInformation** ([see the returnable fields for the GET Person method](#)), PersonId, PersonLink, RefundContributionId, RefundContributionLink, TransactionNumber

GET Contribution List

Returns a list of contributions. Additionally, you can pass query string parameters to filter the list.

Endpoint:

contribution/list

Optional Parameters:

FirstName, LastName, FromDate, ToDate, TypeFilter, FundFilter, ProjectFilter, Transaction, SortExpression

Example:

contribution/list?firstName=John&lastName=Cook&fromDate=2012-03-21&toDate=2012-12-12

Returnable Fields:

[See the returnable fields for the GET Contribution method.](#)

GET Contribution Batch List

Returns a list of batch types.

Endpoint:

batchtype/list

Returnable Fields:

Name, TypeID

POST Methods

ModifyResult Response

All POST methods have a response of ModifyResult.

Response Fields:

- **ErrorMessage** – If an error occurs during modification, this contains the message of the error.
- **Link** – The link to the modified entity.
- **Successful** – True or False if the modification was successful.
- **ValidationResults** – List of validation errors, if any.

Example Response:

```
<ModifyResult>

  <ErrorMessage/>

  <Link>group/1/member/448</Link>

  <Successful>True</Successful>

  <ValidationResults>

    <Key/>

    <Message/>

  </ValidationResults>

</ModifyResult>
```

General Instructions:

In most cases, in order to utilize the Post methods you will need to pass in the XML of the object you are adding/updating. The easiest way to accomplish this is to use the Get method for the corresponding Post method and then copy and paste the XML that is returned. You can then change values to reflect what needs to be added or updated.

PERSON

POST Update/Add Person

Updates an existing person or creates a new one.

Endpoint:

person/{id}/update
person/add

Example Request:

```
<Person>

<Addresses>

<Address>

<AddressID>2</AddressID>

<City>Cordova</City>

<Country>US</Country>

<Latitude>35.162344</Latitude>

<Longitude>-89.714422</Longitude>

<PostalCode>38018</PostalCode>

<Primary>true</Primary>

<State>TN</State>

<StreetLine1>7345 Goodlett Farms Pkwy</StreetLine1>

</Address>

</Addresses>

<Age>20</Age>

<AreaName/>

<BirthDate>1990-10-05T00:00:00</BirthDate>

<BlobLink>
```

<http://example.com/CachedBlob.aspx?guid=84D2F742-59C4-4910-B626-CA19C8E8B2D7>

```
</BlobLink>

<BusinessPhone/>

<CampusID>1</CampusID>

<CampusName>Campus 1</CampusName>

<CellPhone/>

<FamilyID>2</FamilyID>

<FamilyMemberRoleID>29</FamilyMemberRoleID>

<FamilyMemberRoleValue>Adult</FamilyMemberRoleValue>

<FirstActiveEmail>email@example.com</FirstActiveEmail>

<FirstName>William</FirstName>

<Gender>Male</Gender>

<HomePhone>(901) 555-5555</HomePhone>

<LastName>Smith</LastName>

<MemberStatusID>958</MemberStatusID>

<MemberStatusValue>Member</MemberStatusValue>

<NickName>"Bill"</NickName>

<PersonGUID>a41061d9-6419-4502-a0c6-3df2b877eaff</PersonGUID>

<PersonID>3</PersonID>

<PersonLink>person/3</PersonLink>

<RecordStatusValue>Active</RecordStatusValue>

</Person>
```

POST Update Person Picture

Uploads a new picture for the given person.

Endpoint:

person/{id}/picture?extension={extension}

Note:

- Extension is required.
- The extension parameter should be a valid image extension: jpg, gif, png, etc.
- The request should send a byte array of the file contents.

POST Add Person Note

Creates a new note for the given person and returns XML for the PersonNote created.

Endpoint:

person/{id}/note

Note:

- Private must be set to “true” or “false.”
- SecurityTemplateID is obtained from the security/template/list endpoint.
- Pass in a SecurityTemplateID of 0 for All Users access.
- All of the XML Elements need to be in alphabetical order or the values will not be saved properly.

Example:

person/542/note/add

Example Request:

```
<PersonNote>

  <Private>true</Private>

  <SecurityTemplateID>2</SecurityTemplateID>

  <Text>Test note</Text>

</PersonNote>
```

GROUPS

POST Add Group Member

Adds an existing member with the given person ID to the given group ID. If the person is already in the group an error will be returned.

Endpoint:

group/{groupid}/member/{personid}

Example:

group/1/member/4016/add

Example Request:

```
<GroupMember>
<Active>true</Active>
<MemberNotes/>
<RoleID>24</RoleID>
<UniformNumber>-1</UniformNumber>
</GroupMember>
```

POST Update Group Member

Updates the existing member with the given person ID in the given group ID. If the person is not in the group, an error will be returned.

Endpoint:

group/{groupid}/member/{personid}/update

PROFILES

POST Add Profile

Adds a profile.

Endpoint:

profile/add

POST Update Profile

Updates the profile with the given profile ID.

Endpoint:

profile/{id}/update

POST Add Profile Member

Adds an existing member with the given person ID to the given profile ID. If the person is already in the profile an error will be returned.

Endpoint:

profile/{id}/member/{personid}/add

POST Update Profile Member

Updates the existing member with the given person ID in the given profile ID. If the person is not in the group, an error will be returned.

Endpoint:

profile/{id}/member/{personid}/update

POST Add Profile to Parent

Adds a profile to the parent of the given parent ID.

Endpoint:

profile/{parentid}/profile/add

CONTRIBUTIONS

POST Add Batch

Adds a batch.

Endpoint:

batch/add

POST Update Batch

Updates the batch with the given batch ID.

Endpoint:

batch/{batchid}/update

POST Add Contribution to Batch

Adds a contribution to the batch with the given batch ID.

Endpoint:

batch/{batchid}/contribution/add

Note:

- Pass the batch ID in the URL.
- The contribution details (obtained from contribution/{contributionid} endpoint) should be passed in the XML request.

POST Add Contributions by Batch (ShelbyTELLER XML format)

Adds contributions by batch.

Endpoint:

contribution/bulkadd

This has extra options that can be added to the XML:

- BatchID of an existing batch to add the contributions to.
- BatchType to set the type.
- ContinueOnError should be true if you want contributions to be added even if some of the contributions fail validation (for example, they don't match a person). This defaults to false, therefore if one contribution fails nothing is added.