

Affinitomics API methods

Sample Code

Example code utilizing the Affinitomics API can be found at these locations:

https://github.com/Affinitomics/affinitomics_js_lib

https://github.com/Affinitomics/affinitomics_dungeon_example

General Arguments

All api calls will need to be provided the following arguments for your request, in addition to arguments necessary for a specific api method:

| | |
|----------------------------|--|
| user_key (required) | the api key provided by Prefrent |
| ctype (required) | the type of client making the request |
| cversion (required) | the version of the client making the request |
| nonce (required) | a unique, random string between 8-64 characters in length. |

All api calls will return the following response variables for your request, in addition to response variables for a specific api method:

| | |
|---------------|---|
| status | Indicates the result of the api call. Can be 'success' or 'error' |
| errors | a list containing error codes and matching error messages |

Available Methods

RequestDomains

Returns a list of domains associated with the user key

Arguments

This method does not require any additional arguments

Response variables (dictionary)

| | |
|----------------|---|
| domains | list of domain ids associated with the user_key. Domains can be any unique string and are not just limited to URLs |
|----------------|---|

Example Response

```
{
  "domains":
  [
    "83cf93f0-31c5-4098-ac93-42d8930c3b73",
    "subuPref.com",
    "cowsAndCars",
    "0cd95b90-37be-4dc4-ae73-ad9d7d077395"
  ],
  "status": "success",
  "errors": []
}
```

RequestArchetypes

Returns a list of archetypes associated with the user key

Arguments

This method does not require any additional arguments

Response variables (dictionary)

| | |
|-------------------|---|
| archetypes | list of archetype ids that are associated with the given user_key |
|-------------------|---|

Example Response

```
{
  "archetypes":
  [
    {
      "archetype_id": "http://subuPref.com/infinit/2015/05/24/satellite-wagon/",
      "domain": "subuPref.com"
    }
  ]
}
```

```

    },
    {
      "archetype_id": "3302773c-f678-4e52-9ecf-dee20ca4ca52",
      "domain": "83cf93f0-31c5-4098-ac93-42d8930c3b73"
    },
    {
      "archetype_id": "viper",
      "domain": "cowsAndCars"
    }
  ],
  "status": "success",
  "errors": []
}

```

RequestArchetypesByDomain

Returns a list of archetypes associated with the user key and the given domain

Arguments

domain the domain id from which to the archetypes belong

Response variables (dictionary)

archetypes list of archetype ids that are associated with the given user_key

domain the domain id from which to the archetypes belong

Example Response

```

{
  "archetypes": [
    "http://subuPref.com/infinity/a-chevy-wagon/",
    "http://subuPref.com/infinity/2015/09/07/crazy-hippos/",
    "viper",
    "http://subuPref.com/infinity/snakes-on-a-plate/",
    "motorcycle",
    "http://subuPref.com/infinity/2015/09/04/vipers-are-awesome/",
    "http://subuPref.com/infinity/activity/viper-page/",
    "http://subuPref.com/infinity/2015/05/24/satellite-wagon/",
    "horse",
    "cuda",
    "bike"
  ],
  "domain": "subuPref.com",
  "status": "success",
  "errors": []
}

```

}

SyncArchetype

Syncs an archetype to the Affinitomics server. If the archetype does not yet exist, the server will create an archetype with the given id. If the domain does not yet exist for the given user_key, it will also be created. Note that although the arguments ***descriptors***, ***draws***, and ***distances*** are all optional, at least one of the arguments must be given in order for the archetype to be able to be matched against other archetypes.

Arguments

| | |
|-------------------------------|--|
| domain | the domain id to which this archetype does/will belong. If the domain does not yet exist, it will be created. Domains can be any unique string and are not just limited to URLs |
| archetype_id | id of the archetype to use to sync data. If the id does not yet exist, it will be created |
| sets (optional) | list of set ids that the given archetype_id is a member of |
| descriptors (optional) | array of descriptors for the archetype. A cars archetype might include the descriptors "cars, automobiles" |
| draws (optional) | <p>an array of dictionaries that each contain a draw term and magnitude for the archetype.</p> <p>term, a draw to be added to the archetype. A cars archetype might include the draw "horsepower"</p> <p>magnitude, a number from 1-5 that is the magnitude of the draw, with 1 being a low draw and 5 being the highest. The higher the draw, the more likely that this archetype will match archetypes that have this draw term in their descriptors</p> |
| distances (optional) | an array of dictionaries that each contain a distance term and magnitude for the archetype. |

term, a distance to be added to the archetype. A cars archetype might include the distance “boat”

magnitude, a number from 1-5 that is the magnitude of the distance, with 1 being a low distance and 5 being the highest. The higher the distance, the more likely that this archetype will not match archetypes that have this distance term in their descriptors

title (optional) a title to be given to the archetype, if desired

Response variables (dictionary)

archetype_id archetype id that was synced/created
domain the domain used/created in the request

Example Response

```
{
  "domain": "subuPref.com",
  "archetype_id": "bike",
  "status": "success",
  "errors": []
}
```

RemoveArchetype

Completely removes an archetype from the domain and any association with the user_key

Arguments

domain the domain id from which to the archetypes belong
archetype_id id of the archetype to be deleted

Response variables (dictionary)

archetype_id archetype id used in the request
domain the domain used in the request

Example Response

```
{
  "domain": "subuPref.com",
  "archetype_id": "bike",
  "status": "success",
  "errors": []
}
```

RequestRelatedArchetypes

Finds and returns matching archetypes in the given domain to the given archetype id.

Arguments

| | |
|---------------------|--|
| domain | the domain id from which to request the related archetypes |
| archetype_id | id of the archetype to use to find related archetypes |

Response variables (dictionary)

| | |
|---------------------|--|
| archetype_id | archetype id used in the request to find related archetypes |
| domain | the domain used in the request |
| best_matches | <p>a list of dictionaries containing the related results. Each dictionary will have:</p> <ul style="list-style-type: none"> score, a value indicating the strength of the match. The higher the number, the greater the match, with the highest score being 100 archetype_id, the id of the matching archetype |

Example Response

```
{
  "matches": [
    {
      "score": 100,
      "archetype_id": "bike"
    },
    {

```

```

        "score":100,
        "archetype_id":"cuda"
      },
      {
        "score":80,
        "archetype_id":"motorcycle"
      }
    ],
    "domain":"subuPref.com",
    "archetype_id":"viper",
    "status":"success",
    "errors":[]
  }
}

```

RequestArchetypeInfo

Returns all info related to the given archetype id in the domain

Arguments

| | |
|---------------------|--|
| domain | the domain id from which to request the archetype info |
| archetype_id | the id of the archetype for the requested info |

Response variables (dictionary)

| | |
|-----------------------|---|
| archetype_id | archetype id for which the info was requested |
| domain | the domain of the requested archetype info |
| archetype_data | a dictionary of info related to the archetype |

Example Response

```

{
  "archetype_id":"viper",
  "archetype_data":
  {
    "descriptors":["car"],
    "draws":
    [
      {
        "term":"bicycle",
        "magnitude":1
      },
      {

```

```

        "term": "moped",
        "magnitude": 1
      },
      {
        "term": "skateboard",
        "magnitude": 1
      }
    ],
    "distances": [
      {
        "term": "horses",
        "magnitude": -1
      },
      {
        "term": "cows",
        "magnitude": -1
      }
    ]
  },
  "domain": "localhost",
  "status": "success",
  "errors": []
}

```

RankAgainstArchetypes

Matches a list of given archetypes in the given domain to the given archetype id.

Arguments

| | |
|---------------------------|--|
| domain | the domain id from which to request the related archetypes |
| archetype_id | id of the archetype to use to find related archetypes |
| rank_archetype_ids | an array of archetype ids, for specific pre-existing archetypes, to be used to rank against the given archetype_id |

Response variables (dictionary)

| | |
|---------------------|---|
| archetype_id | archetype id used in the request to find related archetypes |
| domain | the domain used in the request |

best_matches

a list of dictionaries containing the related results.
Each dictionary will have:

score, a value indicating the strength of the match. The higher the number, the greater the match, with the highest score being 100

archetype_id, the id of the matching archetype

Example Response

```
{
  "matches":
  [
    {
      "score":100,
      "archetype_id":"bike"
    },
    {
      "score":100,
      "archetype_id":"cuda"
    },
    {
      "score":80,
      "archetype_id":"motorcycle"
    }
  ],
  "domain":"subuPref.com",
  "archetype_id":"viper",
  "status":"success",
  "errors":[]
}
```

AddSetToDomain

Adds a set to the given domain. The set id will be unique within the domain.

*Arguments***domain**

the domain id to which the set will be added

set_id

the set id to be added to the domain. Set will be created, and added to the domain, if it does not already exist

Response variables (dictionary)

| | |
|---------------|--|
| domain | the domain id to which the set was added |
| set_id | the set id that was added/created in the request |

RemoveArchetypeFromSet

Removes an archetype from a given set in the domain.

Arguments

| | |
|---------------------|--|
| domain | the domain id from which the set and archetype belongs |
| set_id | the set id to remove the archetype from. |
| archetype_id | the id of the archetype to remove from the given set |

Response variables (dictionary)

| | |
|---------------------|--|
| domain | the domain used in the request |
| set_id | the set id that the archetype was removed from |
| archetype_id | the id of the archetype removed from the set |

AddArchetypeToSet

Adds an archetype to a given set in the domain. If the set id does not already exist, it will be created.

Arguments

| | |
|---------------------|--|
| domain | the domain id from which the set belongs |
| set_id | the set id to add the archetype to. Set will be created, and added to the domain, if it does not already exist |
| archetype_id | the id of the archetype to add to the given set |

Response variables (dictionary)

| | |
|---------------------|--|
| domain | the domain used in the request |
| set_id | the set id that the archetype was added to |
| archetype_id | the id of the archetype added to the set |

GetArchetypesInSet

Returns a list of archetypes that are in a given set in the domain.

Arguments

| | |
|---------------|---|
| domain | the domain id from which the set belongs |
| set_id | the set id from which to retrieve archetype ids |

Response variables (dictionary)

| | |
|-------------------|---|
| domain | the domain used in the request |
| set_id | the set id that contains the archetype ids |
| archetypes | a list of archetype ids that are in the given set |

RemoveSetFromDomain

Removes a set from the given domain.

Arguments

| | |
|---------------|--|
| domain | the domain id from which the set belongs |
| set_id | the set id to be removed from the domain |

Response variables (dictionary)

| | |
|---------------|--|
| domain | the domain used in the request |
| set_id | the set id that was removed in the request |

RelateArchetypeWithSet

Finds and returns matching archetypes to the given archetype id in the given set in the domain.

Arguments

| | |
|---------------------|---|
| domain | the domain id from which the archetype belongs |
| set_id | the set id to find related archetypes to the given archetype_id |
| archetype_id | id of the archetype to be related to the set |

Response variables (dictionary)

| | |
|---------------------|--|
| domain | the domain used in the request |
| set_id | the set id that contains the archetype ids |
| archetype_id | archetype id used in the request to find related archetypes in the set |
| best_matches | <p>a list of dictionaries containing the related results. Each dictionary will have:</p> <ul style="list-style-type: none"> score, a value indicating the strength of the match. The higher the number, the greater the match, with the highest score being 100 archetype_id, the id of the matching archetype |

Example Response

```
{
  "matches":
  [
    {
      "score":100,
      "archetype_id":"bike"
    },
    {
      "score":100,
      "archetype_id":"cuda"
    },
    {
      "score":80,
      "archetype_id":"motorcycle"
    }
  ],
  "domain":"subuPref.com",
  "archetype_id":"viper",
  "set_id":"fastCars",
  "status":"success",
  "errors":[]
}
```

Errors List

| Error Code | Error Message |
|------------|---|
| 121 | Over the monthly transaction limit for this plan. Upgrade your plan to continue |
| 160 | Warning: you are over the archetype limit for this plan. Please upgrade your plan |
| 161 | Over the archetype limit for this plan. Upgrade your plan to continue |
| 202 | api key is not valid |
| 252 | invalid domain |
| 262 | archetype does not exist |
| 302 | required parameter is not present: {missing parameter} |
| 432 | archetype invalid |
| 819 | unable to delete archetype |
| 869 | unable to create the set |
| 981 | Invalid nonce length |
| 999 | Invalid repeated request |