ARMOR GAMES

Armor Games REST Services

Adventures in the AGI, Version II

Document version 1.3

The second generation of the Armor Games Interface (aka AGI) offers a suite of REST based services that allow our partners to implement deeper integrations with the Armor Games Website. This document outlines the service features available and their requirements for use. If there are any questions or issues with these services please contact us at developers@armorgames.com.

Table of Contents

AGI Developer Access

Access to these services is based on a unique API Key that is given to the Game Studio by Armor Games. Each game by the Game Studio must have its own API Key. A key is a basic UUID formatted string. Example:

```
550e8400-e29b-41d4-a716-446655440000
```

You can request an API Key by sending an email to developers@armorgames.com.

AGI Error/Status Codes

Both error and success conditions are implemented as standard HTTP status codes. As a convenience, these same codes are also implemented in the body of the response using the `code` property. The following are a list of codes currently used.

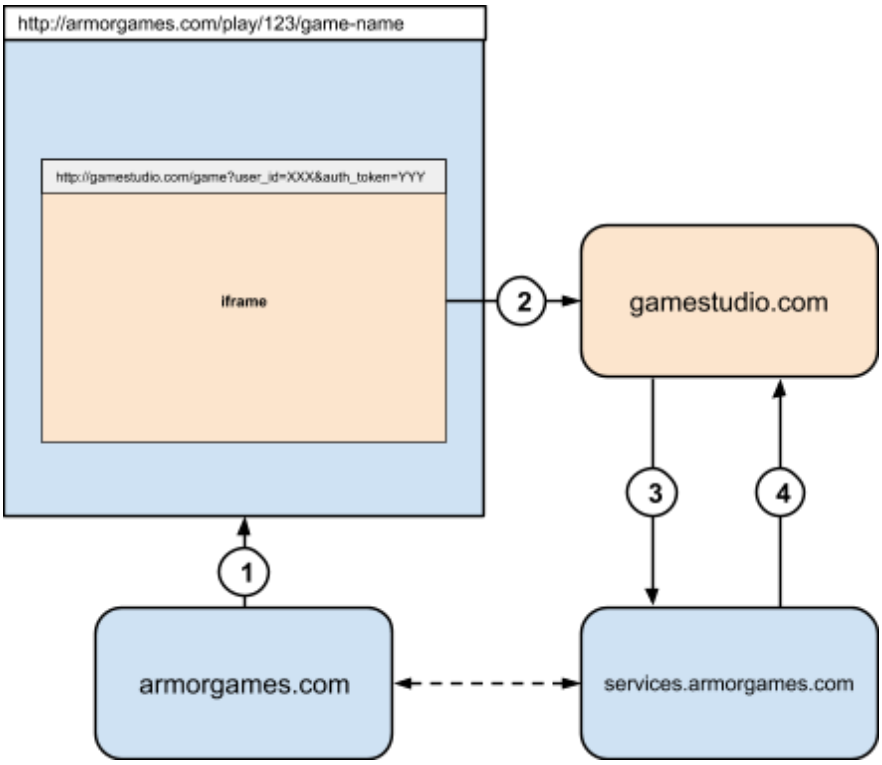
Code	Description
200	OK Standard response for a successful request. However it doesn't necessarily mean you received exactly what you were expecting (see expired token example).
400	Bad Request This occurs if a required parameter is missing or not in the expected format.
401	Unauthorized This occurs when the API Key is missing or invalid.
403	Forbidden This occurs when trying to access a secure resource insecurely (ie using HTTP when you need to use HTTPS).
404	Not Found Occurs if the URI is malformed or the target resource is not available.
405	Method Not Allowed This occurs when an unsupported method type is used to request a resource (ie

	using POST to access the Authentication Service).
415	Unsupported Media Type This occurs when a resource is requested in a format that is not supported (ie something other than JSON or XML).
500	Internal Server Error Occurs when something failed during the request and we can not provide a more specific code. In these cases see the message property for further details.
503	Service Unavailable The service is temporarily offline.
504	Gateway Timeout The service had an internal timeout from one of its upstream servers.

Authentication Service

The AGI's REST based Authentication Service provides two primary functions. First it allows game studios who host their own content with Armor Games to securely access Armor Games user data. Secondly it provides the ability for game studios to validate the authenticity of a given user. *Validating every game request with the Authentication Service is key in accessing the latest user data and ensuring the validity of the user.*

Authentication Flow (Iframe)



1. User loads the play page from armorgames.com web server.
2. An embedded iframe on the play page makes a request to the game studios web servers to load game content. The iframe request includes two GET parameters:
    a. `user_id` The current user's ID.
    b. `auth_token` A temporary authentication token that will be used to access user data.
3. The game studio's web server makes a secure request Armor Games services to retrieve user data. This request includes 3 GET parameters:
    a. `user_id` The user ID received from the iframe request.
    b. `auth_token` The authentication token received from the iframe request.
    c. `api_key` The API key that corresponds to the current game.

4. If all parameters are valid and the authentication token has not expired, the Armor Games service returns user data in JSON or XML format.

IFRAME Request Structure

External games are loaded via an iframe within the armorgames.com site. The URI loaded is specified by the Game Studio prior to game launch. If a user is logged in then the following query string parameters will be added to the Game Studios URI request.

Name	Description
user_id	32 hexadecimal digits (md5) The unique and consistent ID of the current user
auth_token	32 hexadecimal digits (md5) A temporary authentication token that can be used to access the user's data

Authentication Flow (flash)

The authentication process for flash is nearly identical to the Iframe flow. The only difference is that the **`auth_token`** and **`user_id`** are passed into the movie via the flashvars parameter.

```
<object … >
  ...
  <param name="flashvars" value="auth_token=bbfe02aef4493fdcd72848ba3a9e73fd&
                                 user_id=59dd255c50b9d521ab1a94e9595ed489" />
  <embed … flashvars="auth_token=bbfe02aef4493fdcd72848ba3a9e73fd&
                      user_id=59dd255c50b9d521ab1a94e9595ed489" />
</object>
```

Note: There are no actual line breaks in the flashvars string. It is broken here for readability.

Accessing flashvars varies between ActionScript 2.0 and ActionScript 3.0. You can review Adobe's documentation on how to access this data in either version of ActionScript.

Service URI

[GET] https://services.armorgames.com/services/rest/v1/authenticate/user.json

Parameters

Requests to the Authentication service require the following parameters.

Name	Description
user_id	32 hexadecimal digits (md5) The unique and consistent ID of the current user
auth_token	32 hexadecimal digits (md5) A temporary authentication token that can be used to access the user's data
api_key	32 hexadecimal digits (UUID) An access key that is unique to each game

Response

The resource can be formatted as either JSON or XML depending on the extension used in the service URI ('json' for JSON and 'xml' for XML). The following is the response to a successful request to the Authentication Service.

JSON: https://services.armorgames.com/services/rest/v1/authenticate/user.**json**

```
{
 version: 1,
 code: 200,
 message: "OK",
 payload: {
   uid:
"79054025255fb1a26e4bc422aef54eb4",
   username: "JohnDoe"
 }
}
```

XML: https://services.armorgames.com/services/rest/v1/authenticate/user.**xml**

```
<agi version="1">
 <version>1</version>
 <code>200</code>
 <message>OK</message>
 <payload>

<uid>79054025255fb1a26e4bc422aef54eb4</uid>
   <username>JohnDoe</username>
 </payload>
</agi>
```

The following is a response for a token that is invalid or expired. Notice that the payload value is `NULL` (or empty in the case of XML). *This is the key to determining if the user is authenticated.* The message property will give a hint as to the current condition.

JSON

```
{
 version: 1,
 code: 200,
 message: "Either the user is not logged in or the authentication token has
expired",
 payload: null
}
```

XML

```
<agi version="1">
 <version>1</version>
 <code>200</code>
   <message>
     Either the user is not logged in or the authentication token has
expired
   </message>
 <payload/>
</agi>
```

Authentication Testing

Developers can test their API key and server code prior to game launch.

1. Request test accounts from our web team (please don't create a new account)
2. Visit http://armorgames.com or (http://stage.armorgames.com if you're working of that) and login to your account.
3. While logged in, visit the Token Generation Service: http://armorgames.com/service/user-auth-token-generator/123. Replace '123' with your armor games game ID (dont know it? contact developers@armorgames.com and we'll help you out). You will see a response that is structured like the following:

```
{
 status: true,
 game_id: "123",
```

```
 uid: "e563ebd827f5ceb25000df75cb81748d",
 token:
"6c1779b5dec8048c334dea46b1bc2de4",
 created_on: 1337817765,
 expires_on: 1337818065,
 duration: 0.032696008682251
}
```

4. Copy the value from the `uid` property and use that for the `user_id` parameter. Use the value for the `token` property as the `auth_token` parameter. With these two values and your API key you can now make a service call and fetch your own user data. Note that authentication tokens only last for a short time. However you can always request another via the Token Generator Service.

Users Service

The Users service returns user data for the given user ID (aka uid). The uid is passed in as a URL path element while the api key is passed as a GET parameter.

Service URI

[GET] https://services.armorgames.com/services/rest/v1/users/[uid].[json|xml]

Parameters

Requests to the Users service require the following parameters.

Name	Description
api_key (optional)	32 hexadecimal digits (UUID) An access key that is unique to each game Note: When an api_key is not provided or is invalid, only a subset of the user data will be returned which includes: uid, username and avatar. Also, if the user has played the game before, a new field “plays_game” will show up on the request, with a value of 0 or 1. The game must be a launchable game, such as an mmo, for the plays_game value to work.
username (optional)	If set to 1 (or true), you may supply a username instead of a uid as part of the service call, and it searches for and retrieves data based on username instead of uid.

Response

The resource can be formatted as either JSON or XML depending on the extension used in the service URI (‘json’ for JSON and ‘xml’ for XML). The following is the response to a successful request to the Users Service.

JSON: https://services.armorgames.com/services/rest/v1/users/c3323c76c063636c91d2e8cb8f4dfcaf.**json**


```
{
  version: 1,
  code: 200,
  message: "OK",
  payload: {
    uid: "c3323c76c063636c91d2e8cb8f4dfcaf",
    username: "test_user",
    avatar: "http://armatars.armorgames.com/armatar_149_50.50_c.jpg",
    birthday: "0000-00-00",
    gender:"Male",
```

```
    created_on: "1335466318"
  }
}
```

XML: https://services.armorgames.com/services/rest/v1/users/c3323c76c063636c91d2e8cb8f4dfcaf.**xml**

```
<agi version="1">
  <version>1</version>
  <code>200</code>
  <message>OK</message>
  <payload>
    <uid>c3323c76c063636c91d2e8cb8f4dfcaf</uid>
    <username>test_user</username>
    <avatar>http://armatars.armorgames.com/armatar_149_50.50_c.jpg</avatar>
    <birthday>0000-00-00</birthday>
    <gender>Male</gender>
    <created_on>1335466318</created_on>
  </payload>
</agi>
```

Friends Service

The Friends service returns user data for all the friends of the given user ID (aka uid). The uid is passed in as a URL path element while the api key is passed as a GET parameter.

Service URI

[GET] https://services.armorgames.com/services/rest/v1/friends/[uid].[json|xml]

Parameters

Requests to the Users service require the following parameters.

Name	Description
api_key	32 hexadecimal digits (UUID) An access key that is unique to each game.
limit	Limit how many friends are returned. Default to 500, Max is 500
offset	Offset for current friends list. If a user has 1000 friends, you would do one request with limit=500,offset=0 and another request limit=500,offset=500

page	Instead of offset you may use this page param as a paginator.

Response

The resource can be formatted as either JSON or XML depending on the extension used in the service URI (‘json’ for JSON and ‘xml’ for XML). The following is the response to a successful request to the Users Service.

JSON: https://services.armorgames.com/services/rest/v1/friends/c3323c76c063636c91d2e8cb8f4dfcaf.**json**

```
{
  version: 1,
  code: 200,
  message: "OK",
  payload: [
    {
      uid:"2d1a888522ac3b1753a05667927870c3",
      username: "user_one",
      avatar: "http://armatars.armorgames.com/armatar_119_50.50_c.jpg",
      birthday: "1987-04-24",
      gender: "Female",
      created_on: "1198082452",
      plays_game: "0"
    },
    {
      uid: "c2b0ce53de54593bc19456e542637e34",
      username: "user_two",
      avatar: "http://armatars.armorgames.com/armatar_150_50.50_c.jpg",
      birthday: "1983-09-09",
      gender: "Female",
      created_on: "1200423194",
      plays_game: "0"
    }
  ]
}
```

XML: https://services.armorgames.com/services/rest/v1/friends/c3323c76c063636c91d2e8cb8f4dfcaf.**xml**

```
<agi version="1">
  <version>1</version>
  <code>200</code>
  <message>OK</message>
  <payload>
    <item>
      <uid>2d1a888522ac3b1753a05667927870c3</uid>
      <username>user_one</username>
      <avatar>http://armatars.armorgames.com/armatar_119_50.50_c.jpg</avatar>
      <birthday>1987-04-24</birthday>
```

```
      <gender>Female</gender>
      <created_on>1198082452</created_on>
      <plays_game>0</plays_game>
    </item>
    <item>
      <uid>c2b0ce53de54593bc19456e542637e34</uid>
      <username>user_two</username>
      <avatar>http://armatars.armorgames.com/armatar_150_50.50_c.jpg</avatar>
      <birthday>1983-09-09</birthday>
      <gender>Female</gender>
      <created_on>1200423194</created_on>
      <plays_game>0</plays_game>
    </item>

  </payload>
</agi>
```

Achievements/Quests Service

The achievements service is split up into two sections. There are Game Achievements and User Achievements. Game Achievements are the actual achievement tied to a game, and User Achievements are individual achievements that a user has been awarded (or are currently working on).

Game Achievements/Quests

You are able to query what Achievements there are and some basic info about them.

To request Achievements for a particular game, you will need to supply the parent_id which is the game_id.

To request information about a specific achievement you will need to supply either the Achievement-ID (aid) or developer_id+type+parent_id.

parent_id depends on the type of achievement, for a type of 'Game' it would be the game_id.

Service URI

[GET] https://services.armorgames.com/services/rest/v1/achievements/[parent_id].[json|xml]

Parameters

Requests to the achievements service have the following parameters:

Method	Name	Description
ALL	api_key	32 hexadecimal digits (UUID) An access key that is unique to each game
ALL	developer_id	For game achievements -- a unique key to identify an achievement tied to a particular game.
ALL	type	The type of achievement. Defaults to 'Game'
ALL	aid (optional)	Achievement ID. If you know the actual database id of the achievement, you can reference it this way.

GET Request Example

JSON: https://services.armorgames.com/services/rest/v1/achievements/7446.json


```
{
  "version":1,
  "code":200,
  "message":"OK",
  "payload":[
     {
        "id":121,
        "developer_id":"GOT_EYES",
        "parent_id":"7446",
        "type":"Game",
        "name":"I can see again!",
        "description":"Recover your optics unit.",
        "admin_description":"",
        "icon":"e3100427987c36090cff1d1903a1c76f.png?v=1352245035",
        "start_value":0,
        "target_value":1,
        "score":0,
        "state":"Active",
        "completion_count":12863,
        "created_date":1358988582,
        "updated_date":1358988582,
        "tags":{
           "1":"Easy"
        }
     },
     {
        "id":122,
        "developer_id":"ENEMY_KILLED",
        "parent_id":"7446",
        "type":"Game",
        "name":"Vermin Spring Cleaning",
        "description":"Clean out enough of those filthy vermin to please mother.",
        "admin_description":"",
        "icon":"a9e7214dbf5e67f6cf19f798f404032d.png?v=1352245934",
        "start_value":0,
        "target_value":1,
        "score":0,
        "state":"Active",
        "completion_count":6644,
        "created_date":1358988582,
        "updated_date":1358988582,
        "tags":{
           "2":"Medium"
        }
     },
```

```
      ...snip other achievements...
    ]
}
```

User Achievements/Quests

You may increment the progress of a User's achievement, set a progress percentage, or simply requests user achievement data. All requests require a valid api_key.

In order to increment or update progress of an achievement, you need to set up a signature, using a specific formula. This formula is calculated for you in our AGI, and currently is not public/part of this document. An Admin or Partner may ignore the signature signing requirement.

When forming your request, you should provide the developer_id+type+pid.

Service URI

[GET/PUT/DELETE] https://services.armorgames.com/services/rest/v1/achievements/**user**/[uid].[json|xml]

Parameters

Requests to the achievements service have the following parameters:
NOTE: Parameters are case sensitive

Method	Name	Description
ALL	api_key	32 hexadecimal digits (UUID) An access key that is unique to each game
ALL	method (optional)	If your browser cannot do a DELETE or PUT you can use this parameter.
ALL	pid	Parent ID. Either a game_id or a website_id, depending on the 'type'
ALL	type	Game or Website, currently.
ALL	developer_id	The ID the achievement users in-game.
PUT	increment	How much you wish to increment the progress of the achievement by.

PUT	progress	A float percentage from 0.0 to 1.0 of the total progress for the achievement.
PUT	reset (optional)	If you supply reset=1 as part of the parameters, it will reset the achievement back to its starting value.
GET	limit (optional)	Limit the amount of achievements returned.
GET	offset (optional)	The offset count before the limit comes into play.
GET	order (optional)	Order results by a specific field: started_date, completed_date, state, parent_id, percent_complete, name
GET	sort (optional)	Sort direction of the order: asc desc
GET	filter (optional)	Filtering results, options: parent (with type game this is the game name), name, percent_complete, state
GET	match (optional)	Used with filter.

Response

GET Request JSON:
https://services.armorgames.com/services/rest/v1/achievements/user/bc361c39c528e399fbb7a9941c50784d.json?filter=parent_id&limit=5&offset=0&match=13662&method=get&api_key=...


```
{
  "version":1,
  "code":200,
  "message":"OK",
  "payload":[
     {
        "id":1233384,
        "uid":"bc361c39c528e399fbb7a9941c50784d",
        "achievement_id":218,
        "parent_id":"13662",
        "current_value":1,
        "percent_complete":100,
        "state":"Completed",
        "started_date":"2013-01-16 20:47:02",
        "completed_date":"2013-01-16 20:47:02",
        "updated_date":"2013-01-16 20:47:02",
        "developer_id":"unluckyDeath",
        "type":"Game",
        "name":"Unlucky 13",
        "target_value":1
     },
     {
        "id":1233363,
        "uid":"bc361c39c528e399fbb7a9941c50784d",
        "achievement_id":166,
        "parent_id":"13662",
        "current_value":1,
```

```
        "percent_complete":100,
        "state":"Completed",
        "started_date":"2013-01-16 20:45:37",
        "completed_date":"2013-01-16 20:45:37",
        "updated_date":"2013-01-16 20:45:37",
        "developer_id":"lowRoad",
        "type":"Game",
        "name":"Take the Low Road",
        "target_value":1
     },
    .. snipped ..
  ]
}
```

PUT Request JSON:
https://services.armorgames.com/services/rest/v1/achievements/user/f6f8a828a87767560d9f5e52aa38b5cb.json?method=put&progress=1.0&dev_id=beacons-x3&pid=10317&type=Game&api_key=...


```
{
   "version":1,
   "code":200,
   "message":"OK",
   "payload":{
      "state":"Completed",
      "current_value":1,
      "percent_complete":100,
      "developer_id":"beacons-x3"
   }
}
```

Products / Premium Content / Purchases

This service is for giving, taking away, or listing products that a user has purchased (or has been given) for a particular game or for the website.

For a PUT request a signature is required as part of the request.

Service URI

https://services.armorgames.com/services/rest/v1/products/user/[uid].[json|xml]

Parameters

Method	Name	Description
ALL	api_key	Your API Key.
ALL	sku	The product SKU identifier.
ALL	parent_id	The ID of the game or website service.
ALL (optional)	type	Either 'Game' or 'Website', defaults to 'Game'
PUT/POST	qty	Amount of incrementing or decrementing that you are attempting
PUT	increment	Set to true if incrementing
PUT	decrement	Set to true if decrementing

Examples

GET Request Example

JSON:

https://services.armorgames.com/services/rest/v1/products/user/b2463b9f5e2335a11acb37d41efcdf7f.json?parent_id=1000042&api_key=...


```
{
   "version":1,
   "code":200,
   "message":"OK",
   "payload":[
      {
         "purchase":{
            "success":false,
            "data":false,
```

```
            "message":"Getting from
product.user\/uat-test_unlockable\/b2463b9f5e2335a11acb37d41efcdf7f\/product"
         },
         "product":{
            "id":114,
            "sku":"uat-test_unlockable",
            "name":"test_unlockable",
            "description":"Unlockable test product",
            "type":"unlockable",
            "price":"1.00",
            "display_order":0,
            "data":[

            ],
            "media":{
               "image":"product-images\/pr3csxmxR7u5Ky52RXUz_star.png?v=1434404834"
            },
            "status":"active",
            "parent_type":"game",
            "sold":0,
            "created_on":"2015-06-15 17:47:14",
            "updated_on":"2015-06-15 17:47:14"
         }
      },
...
```