

## LDNA API

### GENERAL CONSIDERATIONS

- The Player has an established and verified *MARI GAMES* account.
- The Game has an established *MARI GAMES* relationship, establishing an api key and explicit Data Contracts.
- The LDNA API RESTful endpoint calls are encoded in UTF-8.
- Timestamps returned and supplied shall be encoded as ISO 8601 UTC time:

**YYYY-MM-DDTHH:MM:SS.sZ** (example 2013-02-22T13:45:00.0Z)

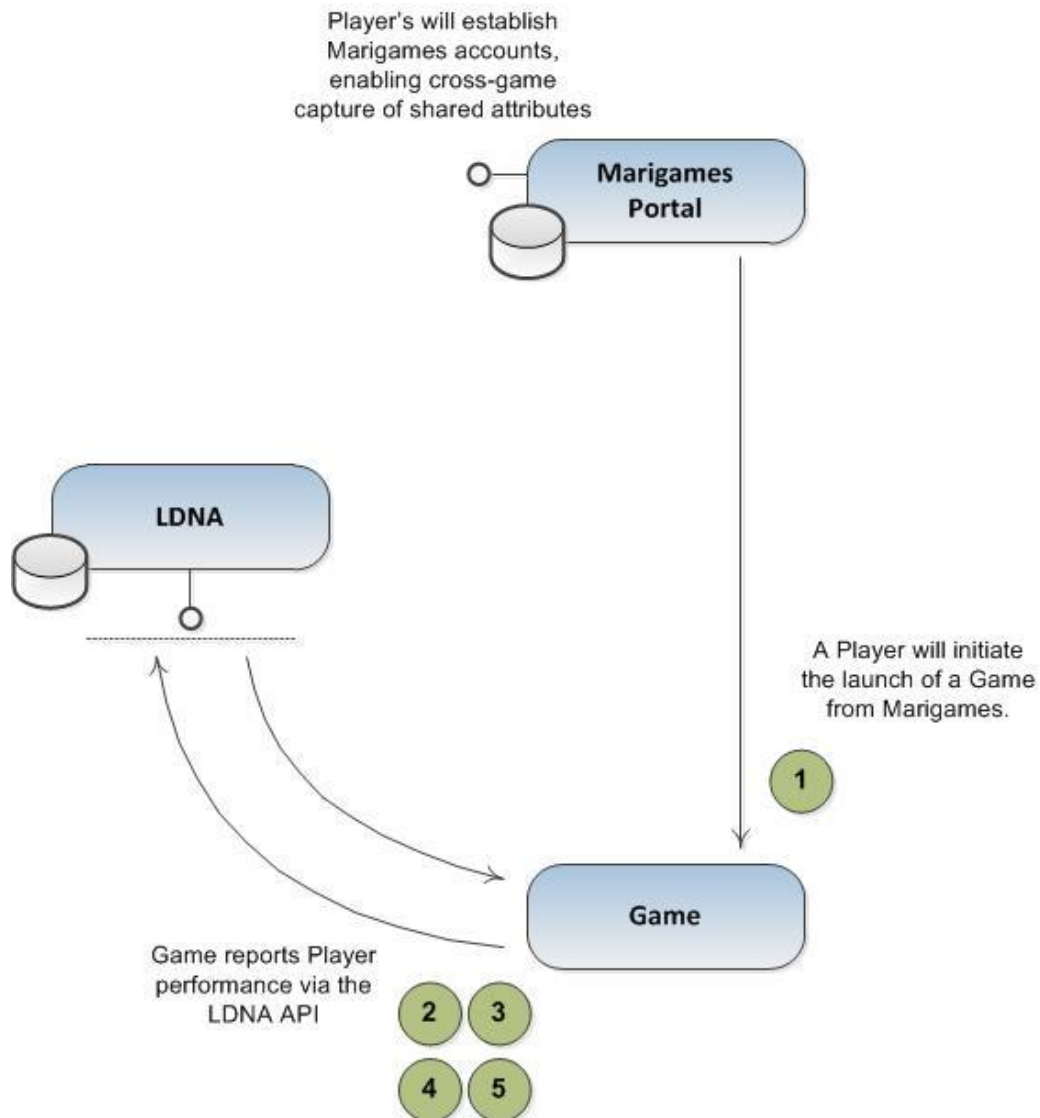
- The LDNA API endpoint will expect, use, and return data in syntactically conformant JSON (specific formats described below).
- Game launch shall be initiated via *MARI GAMES* where the Player will authenticate prior to interacting with the Game. *MARI GAMES* will provide the Player's access token to the Game at launch.
- The Game should not make any assumptions concerning the identity or any other demographic information regarding the Player.
- The Game can only supply (write) performance observations for the Player's Personal Attributes (PAs) that are defined in its Data Contract.

### CONNECTING TO THE LDNA API

For the 2013 *MARI GAMES* competition, only the game play initiated via the *MARI GAMES* portal will be counted for purposes of judging. Figure 1 depicts the scenario where the Player has an established *MARI GAMES* account, has logged into *MARI GAMES*, and has directly selected Game to play.

1. Launch of the Game from the *MARI GAMES* portal is user-directed. In this situation, the Player's intention is clear and *MARI GAMES* will pass a valid `access token` directly to the launched Game. The Game does not require any further authentication in order to submit the Player's attributes (PAs).
2. The Game initiates a new LDNA API session on behalf of the authenticated Player. Establishing an explicit session with the LDNA API provides *MARI GAMES* an opportunity to confirm game launch counts and the Game's conformance to its Data Contract.
3. Retrieve the Player's attributes defined by of the Game's "consume" Data Contracts via LDNA API GET requests. **Note: Retrieving Player attributes is only enabled during the development and testing of the Game. During the judging period, no "consume" Data Contracts will be available.**

4. Submit the Player's attributes defined by the Game's "supply" Data Contracts via LDNA API PUT requests.
5. Explicitly communicate to LDNA that the Game will no longer require or provide any LDNA attributes, effectively ending the LDNA Session and invalidating the current session token.



**Direct Launch of a Game from *MARI GAMES***

## IMPLEMENTATION REFERENCE

### API Endpoint and Version

The LDNA API endpoint shall be: *http://ldna.marigames.org*

- Every string passed to/from the LDNA API endpoint shall be UTF-8 encoded
- All LDNA API endpoint URLs shall include the API major version – formatted “vX/”

### API Error Codes

The LDNA API returns error codes between 11000 and 11999; all errors originating within the LDNA API processes shall be integers of the form: 11xxx (exactly 5 digits). Each error code shall have a short corresponding human-readable message (for testing/debug) – all short error messages shall have the format:

```
"{SUCCESS|ERROR|MISC}: Short Message"
```

The LDNA API partitions error codes by type, source, effect, etc.... The third digit of the returned error code is reserved for this purpose:

```
111xx - LDNA API Authentication/Access Errors
112xx - App and Data Contract Errors
113xx - Retrieve Data Errors
114xx - Submit Data Errors
115xx - Internal Processing Errors
```

### Error Code 11000

The first error code within the API set is reserved to report default successful operations.

```
[11000, "SUCCESS: Operation Completed"]
```

### Standard HTTP Status Codes

```
HTTP_STATUS_OK = 200
HTTP_STATUS_ACCEPTED = 202
HTTP_STATUS_NO_CONTENT = 204
HTTP_STATUS_BAD_REQUEST = 400
HTTP_STATUS_UNAUTHORIZED = 401
HTTP_STATUS_FORBIDDEN = 403
HTTP_STATUS_NOT_FOUND = 404
HTTP_STATUS_CONFLICT = 409
HTTP_STATUS_INTERNAL_SERVER_ERROR = 500
HTTP_STATUS_NOT_IMPLEMENTED = 501
```

### Standard JSON Attribute PUT Request Format

When reporting an attribute, the following data fields shall apply:

Field	Description	Type	Mandatory
attributeID	The attribute being reported	String – must be one of the available attributes for the Data Contract	Yes
value	Indicates the observation of attributes	String representing an decimal number between 0.0 and 1.0	Yes
confidence	Indicates the degree confidence in the observation	String representing a decimal number between 0.0 and 1.0	Yes
timeOfObservation	Indicates the time the attribute was observed	String representing an ISO 8601 datetime.	No - defaults to the system time at the when the transaction was committed
observation	Indicates the method of observation	String, one of: <ul style="list-style-type: none"> <li>• direct</li> <li>• Indirect</li> <li>• none</li> </ul>	No – defaults to “direct”
localContext	Provides additional Game-specific information describing the factors and context related to the observation	String	No – no default

*Example:*

```
{
  "attributeID": "EXAMPLE-ATTRIBUTE-2",
  "timeOfObservation": "2013-01-01T00:00:00.0Z",
  "value": "0.007",
  "observation": "indirect",
  "confidence": "0.50",
  "localContext": "User directly supplied measure."
}
```

## Usage Notes:

- A set of attributes can be reported as a batch by using a JSON list containing multiple entries of the above structure (one entry in the list for each attribute).
- If more than one attribute are reported, each should have a unique `attributeID`. If two or more reported attributes share the same `attributeID`, only the first instance will be committed and the API will report that duplicate IDs were provided.
- The `observation` field is optional; if it is not provided, it will default to "direct" when the transaction is committed.
- The `timeOfObservation` field is optional; if it is not provided, it will default to the local (server) system time when the transaction is committed.
- The `localContext` field is optional; if it is not provided, it will default to empty string ("") when the transaction is committed.
- If the Game wants to explicitly indicate that a given attribute was not observed, the Game should report a `value` of empty string ("") and `observation` of `none`.

## Standard JSON Response Format

```
{
  "status": "200"
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data":
    {
      "attributeID": "EXAMPLE-ATTRIBUTE-1",
      "timeOfObservation": "2013-01-01T00:00:00.0Z",
      "value": "1.0",
      "observation": "direct",
      "confidence": "0.5"
    }
}
```

- All non-HTML 5xx LDNA API responses shall provide all of the fields in the standard JSON response.
- By default, all attributes within a Data Contract will be reported together. If one or more attributes do not have data recorded for an individual, then those attribute's will be reported where `value` will be the empty string ("") and `observation` will be `none` – note this is the default state for any attribute that has not been recorded for an individual.
- The `data` field may include more than one JSON document as a list.

**Example Response: (from retrieving all attributes for a given contract)**

```
{
  "status": "200",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11300",
  "message": "MISC: Requested Data is Incomplete",
  "data": [
    {
      "attributeID": "EXAMPLE-ATTRIBUTE-1",
      "timeOfObservation": "2013-01-01T00:00:00.0Z",
      "value": "1.0",
      "observation": "direct",
      "confidence": "0.5"
    },
    {
      "attributeID": "EXAMPLE-ATTRIBUTE-2",
      "timeOfObservation": "2013-01-01T00:00:00.0Z",
      "value": "0.007",
      "observation": "indirect",
      "confidence": "0.5"
    },
    {
      "attributeID": "EXAMPLE-ATTRIBUTE-3",
      "value": "",
      "observation": "none"
    }
  ]
}
```

**Example Response: (from submitting a single attribute)**

```
{
  "status": "202",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data": ""
}
```

**Example Response: (from submitting an improperly signed Request)**

```
{
  "status": "403",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11001",
  "message": "ERROR: Invalid Signature",
  "data": ""
}
```

## LDNA API METHODS

## /init

**DESCRIPTION** The Game initiates a new session with LDNA for the authenticated Player.

**URL STRUCTURE** `http://ldna.MARI GAMES.org/v1/init`

**VERSIONS** 1

**METHOD** **POST**

`access_token`: (*required – POST Parameter*)

Use to inform LDNA about the authenticated Player; this will be provided by *MARI GAMES* to the Game at launch.

*Note: Prior to the start of the judging period, a Game may provide this value directly to simulate a MARI GAMES initiated launch. A Game may provide the same `access_token` via multiple `init` calls to simulate the same Player.*

**PARAMETERS** `userID`: (*optional – POST Parameter*)

Use to inform LDNA the Game's local User ID.

*Note: LDNA will ignore this value once the judging period beings. During judging, the authenticated user's MARI GAMES ID will be used instead.*

`api_key`: (*required – URL Parameter*)

Use to inform LDNA about source Game.

A session dependent `sessionToken` the Game must use in subsequent data access requests.

**RETURNS**

A session specific `sessionID`, that can be used to reference the collection of attributes reported during that instance of game play.

**Sample JSON Response:**

```
{
  "status": "200",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data": {
    "sessionToken": "117c-0ae3-480c-d52a-197d-7afc-c7c6-c044"
    "sessionID": "221f-fe9a-92d0-00e1-6890-4b32-7afc-c7c6"
  }
}
```

- ERRORS**
- **11001** : ERROR: Invalid Signature
  - **11100** : ERROR: Missing Access Token
  - **11101** : ERROR: App Not Recognized
  - **11102** : ERROR: User Not recognized
  - **11103** : ERROR: Session Already Started

**USAGE NOTES** None

## / {contractID}

**DESCRIPTION** Get a list of all attribute values included in the Data Contract.

**URL STRUCTURE** http://ldna.MARI GAMES.org/v1/<contractID>

**VERSION** 1

**METHOD** **GET**

**PARAMETERS** sessionToken: (*required – URL Parameter*)  
Use to inform LDNA about the authenticated session

**RETURNS** A list of values for all attributes defined the Data Contract for the Player. Some attributes may have an “unknown” status.

Note: For all attribute values are returned, the values returned will be the most recently reported to LDNA (transaction timestamp) for the Player.

### Sample JSON Response:

```
{
  "status": "200",
  "timestamp": "2012-10-01:00:00:00.0Z",
  "code": "11300",
  "message": "MISC: Requested Data is Incomplete",
  "data": [
    {
      "attributeID": "14",
      "timeOfObservation": "2013-01-01T00:00:00.0Z",
      "value": "1.0",
      "observation": "direct",
      "confidence": "0.5"
    },
    {
      "attributeID": "17",
      "timeOfObservation": "2013-01-01T00:00:00.0Z",
      "value": "0.007",
      "observation": "direct",
      "confidence": "0.5"
    }
  ]
}
```



```
        "attributeID": "31",  
        "value": "",  
        "observation": "none"  
    }  
]  
}
```

- ERRORS**
- **11001** : ERROR: Invalid Signature
  - **11110** : ERROR: Unknown Session Token
  - **11111** : ERROR: Session Token has Expired
  - **11200** : ERROR: Unknown Contract
  - **11201** : ERROR: Wrong Contract Type
  - **11203** : ERROR: Contract not Associated with App
  - **11300** : MISC: Requested Data is Incomplete

**USAGE NOTES**

- The contract identified in the URL must be associated with the Game.
- The contract identified in the URL must be a “consume” type Data Contract.

## / {contractID} / {attributeID}

**DESCRIPTION** Get a single attribute value included in the Data Contract.

**URL STRUCTURE** http://ldna.MARI GAMES.org/v1/<contractID>/<attributeID>

**VERSION** 1

**METHOD** GET

**PARAMETERS** sessionToken: *(required – URL Parameter)*  
Use to inform LDNA about the authenticated session

**RETURNS** A single specified attribute from the Data Contract associated with the Player; the attribute may have an 'unknown' or 'unobserved' status.

Note: If a value is returned, it is the value that has been most recently PA value reported to LDNA (transaction timestamp) for the Player.

### Sample JSON Response:

```
{
  "status": "200",
  "timestamp": "2012-10-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data": {
    "attributeID": "2",
    "timeOfObservation": "2013-01-01:00:00:00.0Z",
    "value": "1.0",
    "observation": "direct",
    "confidence": "0.5"
  }
}
```

**ERRORS**

- **11001** : ERROR: Invalid Signature
- **11110** : ERROR: Unknown Session Token
- **11111** : ERROR: Session Token has Expired
- **11200** : ERROR: Unknown Contract
- **11201** : ERROR: Wrong Contract Type
- **11202** : ERROR: Contract does not include the Attribute
- **11203** : ERROR: Contract not Associated with App

- The contract identified in the URL must be associated with the Game.

**USAGE NOTES** • The contract identified in the URL must be a “consume” type Data Contract.

- The attribute identified in the URL must be part of the Data Contract. -

## / {contractID}

**DESCRIPTION** Set a list of attributes values included in the Data Contract.

**URL STRUCTURE** http://ldna.MARI GAMES.org/v1/<contractID>

**VERSION** 1

**METHOD** PUT

**PARAMETERS** sessionToken: *(required – URL Parameter)*  
Use to inform LDNA about the authenticated session

*(required – REQUEST Body)* A list of attribute observations to be associated with the Player. The list does not have to include every attribute defined for the Data Contract.

Data in the REQUEST body must be conformant JSON and the REQUEST must include the following headers:

- Content-Type: application/json
- Charset: UTF-8

### Example Request Body:

#### REQUEST BODY

```
[
  {
    "attributeID": "11",
    "value": "0.75",
    "confidence": "0.30",
    "observation": "indirect"
  },
  {
    "attributeID": "12",
    "value": "0.75",
    "timeOfObservation": "2013-01-01T00:00:00.0Z",
    "confidence": "0.8"
  },
  {
    "attributeID": "13",
    "value": "",
    "localContext": "Player choose path that prevented
observation."
  }
]
```

**RETURNS** Commit status of the attribute transactions.

## Sample JSON Response:

```
{
  "status": "200",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11400",
  "message": "MISC: Submitted Data was Incomplete",
  "data": ""
}
```

- ERRORS**
- **11001** : ERROR: Invalid Signature
  - **11110** : ERROR: Unknown Session Token
  - **11111** : ERROR: Session Token has Expired
  - **11200** : ERROR: Unknown Contract
  - **11201** : ERROR: Wrong Contract Type
  - **11203** : ERROR: Contract not Associated with App
  - **11400** : MISC: Submitted Data was Incomplete
  - **11401** : ERROR: Syntax Error
  - **11403** : MISC: Extra Data Reported
  - **11404** : ERROR: No Data
  - **11405** : MISC: Attribute Submitted Twice

- The contract identified in the URL must be associated with the Game.
- The contract identified in the URL must be a “supply” type Data Contract.

## USAGE NOTES

- If an attribute is provided more than once, only the data from the first occurrence will be committed.
- The “localContext” field may be used to provide optional game-specific context information related to the observation. The value provided for this field is strictly informative and has an unconstrained format.

## / {contractID} / {attributeID}

**DESCRIPTION** Set a specific attribute value included in the Data Contract.

**URL STRUCTURE** http://ldna.MARI GAMES.org/v1/<contractID>/<attributeID>

**VERSION** 1

**METHOD** PUT

**PARAMETERS** sessionToken: *(required – URL Parameter)*  
Use to inform LDNA about the authenticated session

*(required – REQUEST Body)* A single attribute observation from the Data Contract to be associated with the Player

Data in the REQUEST body must be conformant JSON and the REQUEST must include the following headers:

- Content-Type: application/json
- Charset: UTF-8

**REQUEST BODY** Example Request Body:

```
{
  "attributeID": "29",
  "value": "0.15",
  "timeOfObservation": "2013-01-01T00:00:00.0Z",
  "confidence": "0.95",
  "observation": "direct"
  "localContext":
    "Multiple attempts from incorrect selection"
}
```

**PARAMETERS** sessionToken: *(required)* Use to inform LDNA about the authenticated session

**RETURNS** Commit status of the attribute transaction.

## Sample JSON Response:

```
{
  "status": "202",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data": ""
}
```

- ERRORS**
- **11001** : ERROR: Invalid Signature
  - **11110** : ERROR: Unknown Session Token
  - **11111** : ERROR: Session Token has Expired
  - **11200** : ERROR: Unknown Contract
  - **11201** : ERROR: Wrong Contract Type
  - **11202** : ERROR: Contract does not include the Attribute
  - **11203** : ERROR: Contract not Associated with App
  - **11401** : ERROR: Syntax Error
  - **11402** : ERROR: Attribute Mismatch
  - **11404** : ERROR: No Data

- The contract identified in the URL must be associated with the Game.
- The contract identified in the URL must be a “supply” type Data Contract.
- The attribute identified in the URL must be part of the Data Contract identified in the URL.

## USAGE NOTES

- The `attributeID` provided within the Request Body must match the `attributeID` provided in the URL.
- The `localContext` field may be used to provide optional game-specific context information related to the observation. The value provided for this field is strictly informative and has an unconstrained format.

## /stop

**DESCRIPTION** The Game indicates that no further communication to LDNA will occur for the given LDNA Session.

**URL STRUCTURE** `http://ldna.MARI GAMES.org/v1/stop`

**VERSION** 1

**METHOD** **POST**

**PARAMETERS** `sessionToken` : *(required – POST Parameter)*  
Use to inform LDNA about the authenticated User

**RETURNS** Confirmation that the LDNA session has stopped.

### Sample JSON Response:

```
{
  "status": "200",
  "timestamp": "2013-01-01T00:00:00.0Z",
  "code": "11000",
  "message": "SUCCESS: Operation Completed",
  "data": ""
}
```

**ERRORS**

- **11001** : ERROR: Invalid Signature
- **11110** : ERROR: Unknown Session Token
- **11112** : ERROR: Session Already Stopped

- The `sessionToken` becomes invalid after this call is made; any further attempts to communicate data using that token will be denied.

### USAGE NOTES

- If the Game requires further communication, call `init` first. However, be aware that subsequent calls to `init` will result in a different `sessionToken`.