

Aristocrat Systems System 7000™

REST API

3rd Party Interface Specification

Version 1.29

COMMERCIAL-IN-CONFIDENCE

###### Disclaimer

This document is the proprietary property of Aristocrat Technologies Australia Pty Limited and its related corporate bodies (Aristocrat).

This document is intended for 3rd Party vendors who have a Non-Disclosure Agreement with Aristocrat.

All efforts have been made to ensure the accuracy of this document, and whilst due care is taken to ensure that the information contained in this document is correct at the time of printing; Aristocrat does not assume responsibility for errors nor for any consequences arising from errors in this document.

This interface and the information in this document are subject to change without notice.

###### Examples And Images

Reference to companies, names, data, or coding within this publication is for illustrative purposes only.

All examples and images in this publication are fictitious unless expressly stated otherwise and are included for demonstration purposes only. Any resemblance to any person is purely coincIdental and unintentional.

###### Copyright And Trademarks Notice

© Aristocrat Technologies Australia Pty Limited 2023. All rights reserved.

This publication is protected by copyright. No part of this document may be reproduced, stored in a retrievable system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Aristocrat Technologies Australia Pty Limited.

All brand names and product names used in this document are trademarks, registered trademarks, or trade names of Aristocrat Technologies Australia Pty Limited or third party holders. The trademarks of Aristocrat Technologies Australia Pty Limited, its parents, affiliates and subsIdiaries may only be used publicly with express permission from Aristocrat Technologies Australia Pty Limited

Published by: Aristocrat Systems

Aristocrat Technologies Australia Pty Limited Building A, Pinnacle Office Park,

85 Epping Road,

North Ryde NSW 2113 Australia

###### Software Release

Software SYSTEM 7000, Release 7.4 and higher

###### Related Documents

For further information, refer to the following:

|  |  |  |
| --- | --- | --- |
| **Document Title** | **Version** | **Date** |
|  |  |  |
|  |  |  |
|  |  |  |

###### Audience

This document has been prepared for:

* Aristocrat Systems Management
* Aristocrat D&D, D&D Test and Systems Support Teams
* 3rd Party Vendors who have signed a non-disclosure agreement with Aristocrat.

###### Assumptions

This document assumes that, prior to developing a client application to interface to System 7000; a 3rd Party provIder will obtain detailed description from Aristocrat about the workings of this facility.

REST API 3rd Party Interface Specification Version 1.29

# TABLE OF CONTENTS

[PREFACE 5](#_bookmark0)

[Purpose 5](#_bookmark1)

[Scope 5](#_bookmark2)

[Revision History 5](#_bookmark3)

[Glossary 8](#_bookmark4)

[S7000 REST API GENERAL CONVENTIONS 9](#_bookmark5)

[S7000 REST API Clients 9](#_bookmark6)

[URLs 9](#_bookmark7)

[JSON 9](#_bookmark8)

[Licensing and Access Privileges 9](#_bookmark9)

[Authentication and Session management 10](#_bookmark10)

[Session Token 11](#_bookmark11)

[S7000 REST API OVERVIEW 14](#_bookmark12)

[START SESSION – LOGIN 15](#_bookmark13)

[STOP SESSION – LOGOUT 17](#_bookmark14)

[GET NEXT MEMBERSHIP NUMBER 19](#_bookmark15)

[ADD UPDATE MEMBER DETAILS 21](#_bookmark16)

[UPDATE MEMBER DETAILS 29](#_bookmark17)

[ADD UPDATE MEMBER FACIAL IMAGE 38](#_bookmark18)

[ADD UPDATE MEMBER IMAGE DIRECT 42](#_bookmark19)

[DELETE MEMBER 45](#_bookmark20)

[GET MEMBER DETAILS 48](#_bookmark21)

[GET MEMBER DETAILS BY CARD OR MEMBER 52](#_bookmark22)

[GET A MEMBER IMAGE 57](#_bookmark23)

[GET A MEMBER IMAGE DIRECT 60](#_bookmark24)

[GET MEMBER ANALYSIS DETAILS 64](#_bookmark25)

[GET MEMBER PLAYING AT EGM 67](#_bookmark26)

[CHECK MEMBER PIN 71](#_bookmark27)

[RESET MEMBER PIN 75](#_bookmark28)

[VALIDATE MEMBER AND PIN 80](#_bookmark29)

[GET CARDHOLDER DETAILS 84](#_bookmark30)

[GET MEMBER POINT BALANCE 91](#_bookmark31)

[REDEEM MEMBER POINTS 95](#_bookmark32)

[AWARD MEMBER POINTS 99](#_bookmark33)

[VERIFY POS ORDER TIMESTAMP 103](#_bookmark34)

[GET SQL RESULT 106](#_bookmark35)

[GET SQL RESULT AS A FILE 109](#_bookmark36)

REST API 3rd Party Interface Specification Version 1.29

[EXPORT MEMBER FACIAL IMAGES 112](#_bookmark37)

[GET API VERSION 115](#_bookmark38)

[APPENDIX : S7000 DATABASE VIEWS 116](#_bookmark39)

[V\_PUB\_MEMBERS 116](#_bookmark40)

[V\_PUB\_CARDS 117](#_bookmark41)

[V\_PUB\_MEMBER\_ANALYSIS 118](#_bookmark42)

[V\_PUB\_MEM\_OTHERDETAILS 119](#_bookmark43)

[V\_PUB\_MEMBER\_SECADDRESS 120](#_bookmark44)

[V\_PUB\_MEM\_POINT\_BALANCE 120](#_bookmark45)

[V\_PUB\_MEM\_PLAY\_SESSION 121](#_bookmark46)

[V\_PUB\_OCCUPATION\_LIST 122](#_bookmark47)

[CAVEAT 123](#_bookmark48)

[TECHNICAL SUPPORT 124](#_bookmark49)

# PREFACE

### Purpose

This is the reference document for the S7000 REST API, and the resources provided by the API. The S7000 REST API is a standard facility for 3rd party Vendors such as Point of Sale (POS) System, Membership System external to S7000 or any other External Systems who intend to interface to S7000 system.

The S7000 REST API can be used by individual 3rd Party Vendors building solutions for S7000 customers.

This reference document provides general conventions about the S7000 REST API and specific information about each API operation and supported resources.

### Scope

This document is intended for the development personnel of the client application that intends to make requests to S7000 REST API.

It is assumed that the reader of this document has an understanding of client- server interactions and web conventions in general.

Aristocrat reserves the right to change this interface and the information in this document without notice.

Aristocrat does not assume responsibility for errors or for any consequences arising from errors or losses in the provision or use of this interface.

This document contains intellectual property of Aristocrat and contents must not be disclosed to parties not having a non-disclosure agreement with Aristocrat.

Any coding examples shown in this document is provided as a guIde to assist in development of the interface to S7000. These examples might not be programmatically correct.

### Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **System Version** | **API**  **Version** |
| 1.0 | April-15 | Created. |  |  |
| 1.1 | April-15 | Formatting and Consistency |  |  |
| 1.3 | April-15 | 1st Draft Release of the Doc |  |  |
| 1.4 | June-15 | Various changes to HTTP Codes and Messages |  |  |
| 1.5 | July-15 | External Release of the Doc + Updated 2 HTTP codes |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **System Version** | **API**  **Version** |
| 1.6 | Oct-15 | Updated the Add Update member facial image section to notify about the mapping file not supplied option. |  |  |
| 1.7 | Feb-16 | Changed the v\_member\_cards.cardlost column values 1=card lost 0=card not lost |  |  |
| 1.8 | Jun-16 | Updated the Header of the document to reflect the correct version number and the title. |  |  |
| 1.9 | Aug-17 | 3 New APIs introduced getFullMemberDetails, resetPIN, updateMemberDetails, Updated the database view called V\_MEMBER\_ANALYSIS\_VIEW to include analysiscode column |  |  |
| 1.10 | Nov-17 | Added occupation JSON request parameter to the updateMemberDetails and addUpdateMember API Calls |  |  |
| 1.11 | Jul-18 | getMemberImageDirect, addUpdateMemberImageDirect 2 new API calls added to cater for the image data as text format to support thin client integrations |  |  |
| 1.12 | Nov-18 | Updated the V\_PUB\_MEMBER\_ANALYSIS view example and column names |  |  |
| 1.13 | Jul-19 | Updated the REDEEM REWARD POINTS and GET MEMBER POINT BALANCE sections to include the “MULTI” option for the multiple reward bucket redemption. Added a column to the Database View V\_PUB\_MEM\_POINT\_BALANCE |  |  |
| 1.14 | Oct-20 | Updated ADD UPDATE MEMBER DETAILS  and UPDATE MEMBER DETAILS sections to include the new error response 50019 when updating the locked address fields. |  |  |
| 1.15 | Jan-21 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **System Version** | **API**  **Version** |
| 1.16 | Feb-21 | * Updated ADD UPDATE MEMBER DETAILS and UPDATE MEMBER   DETAILS sections and included 6 new fields (idType, idIssueDate, idExpiryDate, idReference, membershipType, cardLayout) to them.   * Updates GET MEMBER DETAILS & GET MEMBER DETAILS BY CARD OR   MEMBER sections and included 6 new fields (idType, idIssueDate, idExpiryDate, idReference, membershipType, cardLayout) in response of these 2 APIs.   * Added 2 new sections GET CARDHOLDER DETAILS and VALIDATE MEMBER AND PIN. * Added a new CAVEAT section |  |  |
| 1.17 | Sep-21 | * Added database view V\_PUB\_OCCUPATION\_LIST to fetch the standard occupation list along with code and active status. |  |  |
| 1.18 | June-22 | * Added UPDATE KIOSK UNIT GROUP   section |  |  |
| 1.19 | Jan-22 | * Correction to UPDATE MEMBER DETAILS – Property Gender will be overridden with empty value when not supplied. |  |  |
| 1.20 | Feb-23 | * addUpdateMember API method allows optional/blank array for analysisArray attribute | 7.4.8.11 | 3.0.45 |
| 1.21 | Mar-23 | * Additional columns added to V\_PUB\_MEMBER\_ANALYSIS for Sub Club details | 7.4.8.11 | 3.0.45 |
| 1.22 | Apr-23 | * Remove Value Paid from V\_PUB\_MEMBER\_ANALYSIS for Sub Club details | 7.4.8.11 | 3.0.45 |
| 1.23 | Apr-23 | * 1.17 updated with 1.22 changes | 7.4.8.11 | 3.0.45 |
| 1.24 | Apr-23 | * 1.19 to 1.20 changes re-instated * Removed UPDATE KIOSK UNIT GROUP | 7.4.8.11 | 3.0.45 |
| 1.25 | Apr-23 | * getAPIVersion API method added to retrieve version information. | 7.4.8.11 | 3.0.49 |
| 1.26 | Sept-23 | * Add Update Member Facial section to specify the location for images. | 7.4.8.11 | 3.0.49 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Description of Changes** | **System Version** | **API**  **Version** |
| 1.27 | Dec-23 | * Updated the section add update member details to indicate the member status is required to be supplied. | 7.4.8.11 | 3.0.49 |
| 1.28 | May-24 | Added note for GET SQL Result section - The restriction to the number of rows returned, can be increased from 500 to 2000.` | 7.4.8.12 | 3.0.53 |
| 1.29 | Jun-24 | Card Number up to 20 digits and StartWebsession and StopWebsession notes. | 7.4.8.12 | 3.0.56 |

### Glossary

The following terms, acronyms and abbreviations are used in this document.

|  |  |
| --- | --- |
| **Term/Acronymn/ Abbreviation** | **Definition** |
| S7000 | Aristocrat’s System 7000 gaming management system |
| EGM | Electronic Gaming Machine, a.k.a. a slot machine, connected to System 7000. |
| API | Application Programming Interface, a means of software applications to interact with other applications |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation, an open standard for transmitting data. |
| REST | Representational State Transfer |
| URL | Uniform Resource Locator. Also known as a web address. |
| Client | 3rd Party Client Application |
| Status | HTTP Status Code of Response |
| Reqd. | Required |

# S7000 REST API GENERAL CONVENTIONS

The S7000 Web Services REST (Representational State Transfer) APIs provIde access to resources via URL paths.

The S7000 REST API uses [JSON](http://en.wikipedia.org/wiki/JSON) as its communication format, and the standard HTTP methods like GET, PUT, POST and DELETE.

The S7000 REST API follows specifications, conventions, and best practices for HTTP and the web in general. This includes things such as case sensitivity of URLs, character encodings, HTTP methods, and so forth.

##### S7000 REST API Clients

A Client application is required to send and receive REST-based JSON-formatted messages to the S7000 Web Services. The Client will make an HTTP request and parse the response and must be equipped to handle SUCCESS as well as FAILURE response messages.

Because the REST API is based on open standards, the 3rd Party Vendor can use any web development language to access the S7000 REST API resources.

##### URLs

The S7000 REST API provides access to resources via URL paths. All S7000 REST API URLs are case-sensitive.

The REST API calls are POSTs and GETs, with the following URL syntax.

|  |
| --- |
| **URL** |
| http://WebServerIPAddress:Port/api/class/methodName |

Here's a Sample URL:

http://10.10.10.103:8080/api/membership/addUpdateMember

#### JSON

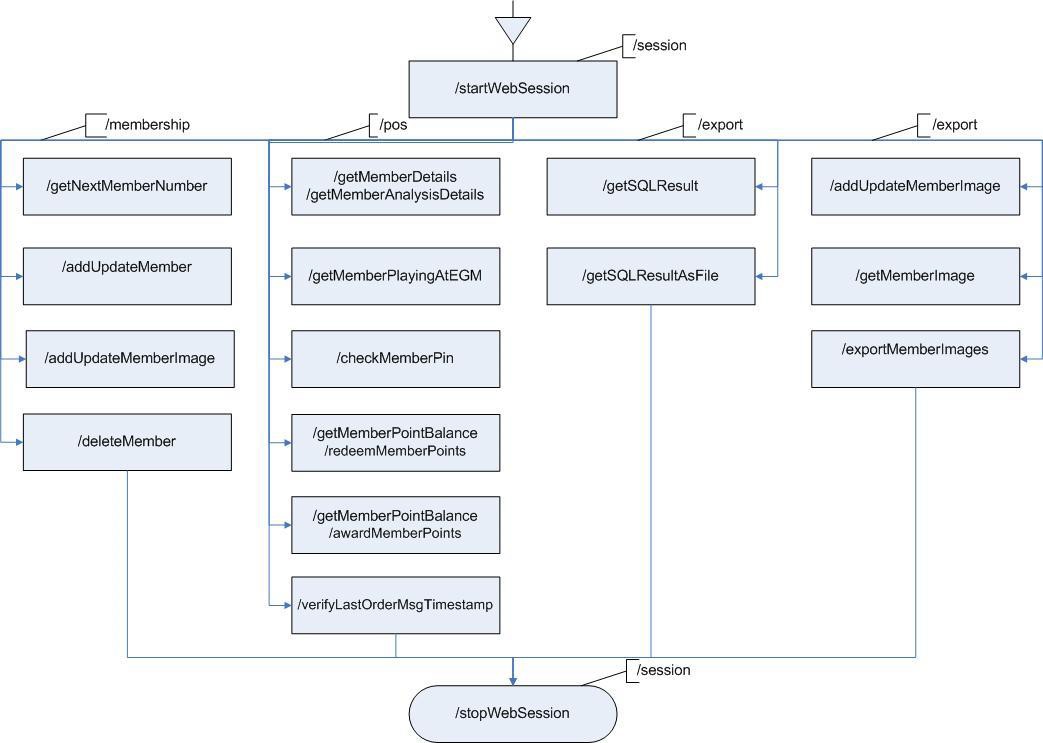
The S7000 REST API follows the specifications and conventions of the JSON data format and any related JavaScript syntax specifications.

##### Licensing and Access Privileges

The 3rd Party vendor is required to register for a unique Registration Code, Username and Password from Aristocrat in order to access the S7000 REST API.

The S7000 REST API resources are categorised as POS, Membership and Export classes. Hence the licensing accommodates various types of application and deployment scenarios in which different 3rd party vendors operate at Aristocrat Customer venues.

The vendors can choose the license options that cover the class of the S7000 REST API requests they will be using at each venue.



In the events where the:

* Client misses to provide the Registration Code, Username and Password, the S7000 sends a response with HTTP status 401 (Unauthorized).
* Client makes the API requests that have access restricted, the S7000 sends a response with HTTP status 403 (Forbidden).
* Client registration has expired and the client makes an API request the S7000 sends a response with HTTP status 402 (Payment Required). In these instances the header and the body of the JSON response indicates the licence expiry status and the expiry date.

##### Authentication and Session management

The S7000 REST API has the concept of a client session. Login requests return a Session token that represents a successful authentication, and which can be used for additional calls without authenticating again.

The 3rd Party client authentication and session management process flows as follows:

1. 3rd Party client sends the login request startWebSession with required properties.
2. If authentication is successful, the S7000 API sends back a Session token. This is the token that represents a session established by the login process, and must be sent along with all subsequent requests to the API.
3. The Session token to be provided in the HTTP header.

##### Session Token

The login request, startWebSession returns a Session Token that represents a successful authentication, and which is to be used for additional calls without authenticating again.

The Session token is a unique value for a request / response exchange until the token expires. The Session token has a timestamp and expiration.

The Session token must be provided as a HTTP header in the following format:

|  |
| --- |
| **HTTP Header Format** |
| Authorization: Bearer<session-token> |

Sample HTTP Header

Authorization: Bearer2a4e975f-6d30-4f0f-a6f1-6ac9a2db8588

By default, the Session token expires after a set period. No requests will be processed with an expired session token until the login process is repeated to get the new session token.

If the 3rd Party client does not make any requests after obtaining the Session token, the session token is inactivated. In this inactive state, the Session token is not valid for requests; the Session token must be re-obtained using the startWebSession login request.

If a 3rd Party client makes a request and does not include a Session token, the server sends a response with HTTP status 401 (Unauthorized).

If a client request contains an invalid session token, the response status is HTTP 401 (Unauthorized).

##### Error Handling

If the S7000 REST API encounters a problem while request/response exchange, the client may receive an error message in one of the following three ways.

* 1. Major errors, such as a web service is down or an authorisation error, cause the API to send an HTTP error page.
  2. Errors within an API, such as a syntax problem or bad property value, usually just return an Error object in the JSON format, rather than the expected response.
  3. And in some cases an indivIdual API may include important information in the data response itself. This is documented in the respective individual API sections.

The best convention is to check every response for HTTP errors and then API errors.

##### HTTP Status Codes

S7000 REST API uses standard HTTP status codes as below: 2XX - Success

4XX - Error occurred in client’s part 5XX - Error occurred in server’s part

The following table shows the HTTP status codes that may be returned, and the accompanying messages.

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 400 | Bad Request |
| 401 | Authentication failure |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |

Note that any of the above single status code may be associated with more than one message. In such cases, the status message correlates with a specific method request.

##### Next Steps

To use the S7000 REST APIs a 3rd Party vendor will need to:

* Develop a web service client or other utility that can send and receive messages via HTTP
* Use the S7000 Sandbox environment for testing purposes, then move to the live environment at the S7000 customer venue for production processing.
* When testing, generate a Session token with the test Registration Code, Username and Password to make calls to the S7000 Sandbox URLs. These credentials may be obtained from Aristocrat.
* When set to go live, use the live Registration Code, Username and Password credentials assigned by Aristocrat to generate new session tokens to be used with the live URLs at an S7000 venue.
* The live Registration code, Username and Password may need periodic renewals depending on the licensing options chosen.

REST API 3rd Party Interface Specification Version 1.29

# S7000 REST API OVERVIEW

The S7000 REST API can be used by the individual 3rd Party vendors such as Point of Sale (POS) System, Membership System external to S7000 and any other external systems to accomplish the following:

For Point of Sale (POS) systems the S7000 REST API offers the request / response exchange that facilitates:

* Enquiry for member details when an S7000 member swipes a card at any POS terminal.
* Member Pin validation check
* Enquiry for member reward point balances
* Redemption and Award of reward points for any S7000 member
* Enquiry for a member currently playing at an EGM
* Enquiry for the timestamp of the last POS order captured by the S7000 system

For Membership Systems which are external to S7000, the REST API may be used to make requests to:

* Register a new member into the S7000 system
* Update details of an existing S7000 member
* Issue or Delete cards for an S7000 member
* Attach a facial image for an S7000 member
* Delete an S7000 member

Additionally, the S7000 REST API allows access to the resources which facilitate the following distinctive features:

* Fetching the results of a SQL query supplied for the S7000 database views
* Export the results of a SQL query as a text file
* Export of the member facial images from the S7000 system
* Import of the member facial images into the S7000 system

REST API 3rd Party Interface Specification Version 1.29

# START SESSION – LOGIN

Request from client to authenticate the supplied login credentials and issue a new session token to be used for subsequent requests.

Before any of the other web method requests it is required that startWebSession method be called with the Registration Code, Username and Password supplied by Aristocrat.

On successful authentication this web request sends back the Session token and expiration timestamp. This Session token is to be used in the header of every subsequent web request to the API.

**Note: The client is required to call startWebSession to re-obtain a new session token whenever the previous session token expires.**

**Request**

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/session/startWebSession |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| registrationCode | String (16) | Y | Registration Code supplied by Aristocrat along with the Username/Password. |
| userId | String (8) | Y | Username supplied by Aristocrat |
| webPassword | String (16) | Y | Password supplied by Aristocrat for the above Username |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |
| sessionToken | String | The unique Identifier of the current session. This value is used in the HTTP header of any subsequent the web requests. |
| sessionExpireTime | String | The time that the current session will timeout. Once the session has timed out, no more requests can be made unless a new startWebSession request is sent. |

**Sample Request**

{ "registrationCode" : "6F7C759AA4D77EAC", "userId" : "TESTBIGT",

"webPassword": "3ACBFA42CFD8C395" }

**Sample Response**

REST API 3rd Party Interface Specification Version 1.29

{ "code": 0, "message": "Success",

"sessionToken": "6b649d8d-3aca-44c0-abd3-37e864a390cb", "sessionExpireTime": "2015-04-14 11:15:00" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 201  OK | Created |
| 400  Bad Request | One or more properties of the request are not valid. Could be a result of incorrect format or the values for the properties.  {"code": 40000,  "message": "Invalid input data"} |
| 401  Unauthorized | The userId request property has incorrect username  {"code": 40100,  "message": "The userId was invalid”} |
| The webPassword request property has incorrect password  {"code": 40100,  "message": “The password was invalid”} |
| 403  Forbidden | The registrationCode request property has incorrect Registration Code  {"code": 40300,  "message": "Registration code: 'X' was not found"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# STOP SESSION – LOGOUT

Request from the client to logout or close the current S7000 web session.

This is the final request from the client when the client intends to close request / response exchange with S7000. It is always recommended to send this request so the web service performs a graceful connection closure out of S7000 system.

This web request requires the session token that was sent out by the startWebSession request.

**Note: The client is required to call the stopWebSession when the token is no longer required.**

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/session/stopWebSession |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| sessionToken | String | Y | The currentSession token sent back by the startWebSession request. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "sessionToken" : "6b649d8d-3aca-44c0-abd3-37e864a390cb" }

**Sample Response**

{ "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200 | The web method call was successful, returns the below code and message. |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| OK | "code": 0, "message": "Success" |
| 400  Bad Request | The sessionToken property of the request may not have been supplied or Could be a result of incorrect format or the value of the property.  {"code": 40000,  "message": "Invalid input data"} |
| 404  Not Found | The value supplied for the sessionToken property is not valid.  {"code": 40400,  "message": "Session token 'XXXXXXXX-XXXX-XXXX-XXXX- XXXXXXXXXXXX' was not found"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET NEXT MEMBERSHIP NUMBER

Request from the client to fetch the Next Membership Number from the S7000.

The next membership number received by the client may be used when requesting S7000 to register/add a new member into the S7000 system.

The request for next membership number may be used by the Membership System in the instances where the S7000 system is required to maintain and generate membership number sequencing.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/membership/getNextMemberNumber |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The next membership number sent back by the S7000. |

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "101884" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getNextMembershipNumber request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 0, "message": "Success"} |
| The getNextMembershipNumber request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden for the current registration  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| 404  Not Found | Could not fetch the next membership number  {"code": 40400,  "message": "Could not fetch next member number"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# ADD UPDATE MEMBER DETAILS

Request from the client to:

1. Add a new member into S7000.
2. Update or replace an existing member’s details.
3. Allocate new active card number or activate an existing card for a member.
4. Delete cards associated with a member.
5. Replace existing member’s analysis details

The S7000 system will create/add a new member if the member does not already exist. And will update the details if the member already exists within the S7000 system.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/membership/addUpdateMember |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| firstName | String(25) | Y | First name of the member. |
| lastName | String(25) | Y | Surname of the member. |
| membershipId | Number(12) | Y | Unique Identifier for the member i.e. the membership number.  All information regarding the member is associated with this Number. |
| Title | String(25) | N | Title of member, E.g. Mr., etc. |
| dateOfBirth | DATE | Y | Date of birth of member. |
| dateOfJoining | DATE | N | Date when the member first joined or registered with the venue.  If not supplied, this will be defaulted to the current system date. |
| Gender | String(1) | N | ‘M’ or ‘F’ for the member’s gender. ‘M’ – for Male; ‘F’ – for Female  If not supplied the gender will be set as Unknown in S7000. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| Telephone | String(20) | N | Home or primary telephone number to go to the main number |
| Mobile | String(20) | N | Mobile phone number. |
| address1 | String(60) | N | Primary address of member (line 1). |
| address2 | String(60) | N | Primary address of member (line 2). |
| address3 | String(60) | N | Primary address of member (line 3) |
| address4 | String(60) | N | Primary address of member (line 4) |
| Postcode | String(12) | N | Postal code for member. |
| barCode | String(37) | N | Mail Barcode for the above address. |
| Email | String(64) | N | Email address. |
| cardNumber | String(20) | N | Support up to 20 digits membership card number.  If the card number is not supplied, the member will be added without a card number; however, such a member will not be recognised on an EGM. |
| cardType | Number(4) | N | The type of the card that is associated with the cardNumber.  If supplied, the cardType of the above cardnumber is updated to this type.  If not supplied, the existing detail for the member will remain unchanged. |
| dateExpiry | DATE | N | Expiry date of the supplied card.  If the card expiry date is not supplied, the expiry date will be defaulted to the expiry date based on the setting of S7000 system. |
| Suspended | Number(1) | Y | Indicates the current suspension status of the member. 0 = member is not suspended, else suspended. |
| memberStatus | Number(1) | Y | The number which represents the member’s current status of membership with the club. The values supported are:   1. – for **Pending** status 2. – for **Active** or **Financial** status 3. – for **Un-financial** status 4. – for **Retired** status |
| occupation | String(30) | N | Occupation of the member. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| analysisArray | Two- dimension al user defined string array | N | Array of Analysis Type and value as member analysis.  The analysis type and value should be supplied as a pair.  This property accepts the numeric Analysis Type Code and the Analysis Value Code as configured in the S7000 system.  E.g.:  "analysisArray" : [["02","35"],  ["03","25"]]  If this property is supplied, the existing member analysis details will be replaced with the supplied details.  "analysisArray" : []  If this property is supplied with blank array, the existing details for the member will be removed.  If this property is not supplied, the existing member analysis details will be deleted in the S7000. Hence the client is required to supply complete member details with every request. |
| idType | Number | N | The number which represents the member’s ID documentType. The values supported are:  5 - DRIVERS LICENCE  10 - PASSPORT  15 - PROOF OF AGE CARD  20 - BIRTHCARD |
| idIssueDate | DATE | N | Issue Date of member ID document |
| idExpiryDate | DATE | N | Expiry Date of member ID document |
| idReference | String (60) | N | An alphanumeric value of member’s ID document. |
| membershipType | Number | N | MembersipType of the patron, should have the same value as of cardType. |
| cardLayout | String | N | Text value of the Patron’s cardType. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "firstName" : "Jack",

"lastName": "Jones", "membershipId" : "12917", "title" : "MR", "dateOfBirth" : "1986-11-11",

"dateOfJoining" : "2014-10-12", "gender" : "M",

"telephone" : "0299994444",

"mobile" : "0413777888",

"address1" : "",

"address2" : "21 West Street", "address3" : "PARRAMATTA",

"address4" : "NSW",

"postCode" : "1122",

"barCode" : "1301012011121020010211300303023313013",

"email" : "[jj@gmail.com](mailto:jj@gmail.com)", "cardNumber" : "1239999997",

"dateExpiry" : "2020-10-12",

"suspended" : "0",

"memberStatus" : "0", "occupation" : "technician",

"analysisArray" : [["02","35"], ["03","25"]],

"idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"cardType": "55",

"membershipType":"55", "cardLayout":"PERPETUAL"

}

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| One or more property values of the request are not valid. Could be a result of incorrect format or the values for the properties.  {"code": 40000,  "message”: “Validation failed for membership profile", "errorMap":  {<field name> : <error message>}} E.g.:  “membershipId": "Invalid Value"  "dateOfBirth": "Invalid date format. Format must be yyyy-MM-dd and must be a valid date" "memberStatus": "memberStatus cannot be blank." |
| {"code": 40001,  "message": “Validation failed for membership profile",  "errorMap":  {"membershipId“: “The supplied Member Number is not valid"}} |
| {"code": 40002,  "message": “Validation failed for membership profile",  "errorMap":  {"firstName" : "Member First Name is missing"}} |
| {"code": 40003,  "message": “Validation failed for membership profile",  "errorMap":  {"lastName": "Member Surname is missing"}} |
| {"code": 40004,  "message": “Validation failed for membership profile",  "errorMap":  {"dateOfBirth" : "Member Date of Birth is Missing"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| 402  Payment Required | The addUpdateMemberDetails request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | The addUpdateMemberDetails request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 412  Precondition Failed | {"code": 41205,  "message": “The value supplied for gender is not valid"} |
| {"code": 41206,  "message": “The supplied card number is not valid"} |
| {"code": 41207,  "message": “Member Suspension Status is not  supplied/valid"} |
| {"code": 41217,  "message": “The value supplied for member status is not  valid"} |
| {"code": 41210,  "message": “The supplied member card number belongs to a  different member number"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |
| Exception when adding member. Contains the exception generated.  {"code": 50008,  "message": “Exception when inserting member:  XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member. Contains the exception generated.  {"code": 50009,  "message": “Exception when Updating member: XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when adding member card. Contains the exception generated.  {"code": 50011,  "message": “Exception when adding card, Member No  :XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member card. Contains the exception generated.  {"code": 50012, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": “Exception when updating cards, Member no:XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating other information for member. Contains the exception generated.  {"code": 50013,  "message": “Exception when updating cards, Member no:XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50014,  "message": “Exception when updating the member status. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50015,  "message": “Error when updating member analysis. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50016,  "message": “Exception when attempting to update member analysis. Error: NNNN, <Message>”} |
| Exception when updating any of the locked address field(s). Venue can lock/ unlock address fields from S7000 Workstation.  {  "code": 50019,  "message": " Address3 Address4 are locked fields and cannot accept value"  } |
|  | Exception when supplied cardType doesn’t match with membershipType.  {  "code": 50022,  "message": "Validation failed for membership profile",  } |
|  | Exception when supplied cardType doesn’t match with the id of cardLayout.  {  "code": 50022,  "message": "The value supplied Card Layout does not match with card type."  } |

**Add a new Member**

When adding a new member the client request should send all the property values each time.

The S7000 system will update the member details if the member exists; else a new member will be added using the supplied membership number and the other property values.

A membership card is allocated to the member if the cardNumber property value is supplied.

The add request cannot be processed successfully if a card with the same Card Number already exists in S7000 and if it is allocated to a different member.

##### Update or replace existing member’s details.

The client request is required to send all the property values every time when updating a member. Corresponding member values in S7000 will be overwritten with the update request.

The S7000 system will update the member details if the member exists; else a new member will be added using the supplied membership number and the other property values.

A new card is allocated to the supplied card number is not present in S7000; and all other pre-allocated cards of the member will be deleted.

If the cardNumber property value is supplied, all pre-allocated cards of the member will be deleted.

The update request cannot be processed successfully if a card with the same Card Number already exists in S7000 and if it is allocated to a different member.

##### Allocate new active card number or activate an existing card for a member.

The client request is required to send all the property values every time when intending to allocate a new card or to activate an existing card for a member.

In order to allocate a new card or activate existing card, the client request has to send cardNumber and cardExpiryDate property values along with the other properties.

Only one card will be allocated for member at any time within the S7000 system. This will be the card number supplied by in the client request.

If the supplied card number is not present within the S7000, the card is allocated, and all other cards will be deleted. If cardNumber property value is not supplied, all cards pre-allocated cards of that member are deleted within the S7000 system.

##### Delete cards associated with a member.

In order to delete cards associated with a member, the Membership system is required to send all the above details except the card number and card expiry date. If cardNumber property value is not supplied, all pre-allocated cards of that member are deleted within the S7000 system.

# UPDATE MEMBER DETAILS

Request from the client to:

1. Update or replace an existing member’s details.
2. Allocate new active card number or activate an existing card for a member.
3. Delete cards associated with a member.
4. Replace existing member’s analysis details

The S7000 system will update the details if the member already exists within the S7000 system.

The client request is required to supply the membershipid and only the property values that are intended to be updated within the S7000 system for the supplied membership Id i.e. it is not mandatory to supply every property value. The existing details for this member will be remain untouched if they are supplied for updating.

'if "" value or any other value is passed in API then data will be updated in the db as null for the below parameters:

* mem\_phone
* mem\_add1
* mem\_add2
* mem\_add3
* mem\_add4
* mem\_pcode
* mem\_barcode
* mem\_idtype(this will throw an error if not an existing value in IDTYPE)
* mem\_date\_idiss
* mem\_date\_idexp
* mem\_idref
* mco\_telephone1
* mco\_email
* mco\_occupation

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/membership/updateMemberDetails |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| firstName | String(25) | Y | First name of the member.  If not supplied, the existing detail for the member will remain unchanged. |
| lastName | String(25) | Y | Surname of the member.  If not supplied, the existing detail for the member will remain unchanged. |
| membershipId | Number(12) | Y | Unique Identifier for the member i.e. the membership number.  All information regarding the member is associated with this Number.  It is mandatory to supply this property for the updateMemberDetails request. |
| Title | String(25) | N | Title of member, E.g. Mr., etc.  If not supplied, the existing title for the member will remain unchanged. |
| dateOfBirth | DATE | Y | Date of birth of member.  If not supplied, the existing date for the member will remain unchanged. |
| dateOfJoining | DATE | N | Date when the member first joined or registered with the venue.  If not supplied, the existing join date for the member will remain unchanged. |
| Gender | String(1) | N | ‘M’ or ‘F’ for the member’s gender. ‘M’ – for Male; ‘F’ – for Female  If not supplied, the existing detail for the member will remain unchanged. |
| Telephone | String(20) | N | Home or primary telephone number to go to the main number  If not supplied, the existing detail for the member will remain unchanged. |
| Mobile | String(20) | N | Mobile phone number.  If not supplied, the existing detail for the member will remain unchanged. |
| address1 | String(60) | N | Primary address of member (line 1).  If not supplied, the existing detail for the member will remain unchanged. |
| address2 | String(60) | N | Primary address of member (line 2).  If not supplied, the existing detail for the member will remain unchanged. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| address3 | String(60) | N | Primary address of member (line 3)  If not supplied, the existing detail for the member will remain unchanged. |
| address4 | String(60) | N | Primary address of member (line 4)  If not supplied, the existing detail for the member will remain unchanged. |
| postcode | String(12) | N | Postal code for member.  If not supplied, the existing detail for the member will remain unchanged. |
| barCode | String(37) | N | Mail Barcode for the above address.  If not supplied, the existing detail for the member will remain unchanged. |
| Email | String(64) | N | Email address.  If not supplied, the existing detail for the member will remain unchanged. |
| cardNumber | String(20) | N | Support up to 20 digits membership card number.  If the card number is not the existing card/s for the member will remain unchanged.  If supplied as a 0, the existing card/s for the member will be deleted. |
| cardType | Number(4) | N | The type of the card that is associated with the cardNumber.  If supplied, the cardType of the above cardnumber is updated to this type.  If not supplied, the existing detail for the member will remain unchanged. |
| dateExpiry | DATE | N | Expiry date of the supplied card.  If not supplied, the existing detail for the member will remain unchanged. |
| Suspended | Number(1) | Y | Indicates the current suspension status of the member. 0 = member is not suspended, else suspended.  If not supplied, the existing suspension status of the member will remain unchanged. |
| memberStatus | Number(1) | N | The number which represents the member’s current status of membership with the club. The values supported are:   1. – for **Pending** status 2. – for **Active** or **Financial** status |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
|  |  |  | 1. – for **Un-financial** status 2. – for **Retired** status   If not supplied, the existing detail for the member will remain unchanged. |
| occupation | String(30) | N | Occupation of the member. |
| analysisArray | Two- dimension al user defined string array | N | Array of Analysis Type and value as member analysis.  The analysis type and value should be supplied as a pair.  This property accepts the numeric Analysis Type Code and the Analysis Value Code as configured in the S7000 system.  E.g.:  "analysisArray" : [["02","35"],  ["03","25"]]  If this property is supplied, the existing member analysis details will be replaced with the supplied details.  If not supplied, the existing detail for the member will remain unchanged. |
| idType | Number | N | The number which represents the member’s ID documentType. The values supported are:  DRIVERS LICENCE PASSPORT  PROOF OF AGE CARD BIRTHCARD  And ID can be fetched from Database against the values. |
| idIssueDate | DATE | N | Issue Date of member ID document |
| idExpiryDate | DATE | N | Expiry Date of member ID document |
| idReference | String (60) | N | An alphanumeric value of member’s ID document. |
| membershipType | Number | N | MembersipType of the patron, should have the same value as of cardType. |
| cardLayout | String | N | Text value of the Patron’s cardType. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "firstName" : "Jack",

"lastName": "Jones", "membershipId" : "12917", "title" : "MR", "dateOfBirth" : "1986-11-11",

"dateOfJoining" : "2014-10-12", "gender" : "M",

"telephone" : "0299994444",

"mobile" : "0413777888",

"address1" : "",

"address2" : "21 West Street", "address3" : "PARRAMATTA",

"address4" : "NSW",

"postCode" : "1122",

"barCode" : "1301012011121020010211300303023313013",

"email" : "[jj@gmail.com](mailto:jj@gmail.com)", "cardNumber" : "1239999997",

"dateExpiry" : "2020-10-12",

"suspended" : "0",

"memberStatus" : "0", "occupation" : "technician",

"analysisArray" : [["02","35"], ["03","25"]],

"idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"cardType": "55", "membershipType":"55", "cardLayout":"PERPETUAL"

}

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0,  "message": "Success" |
| 400  Bad Request | One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}}  E.g.: “memberId": "Invalid Value" |
| One or more property values of the request are not valid. Could be a result of incorrect format or the values for the properties.  {"code": 40000,  "message”: “Validation failed for membership profile", "errorMap":  {<field name> : <error message>}} E.g.:  “membershipId": "Invalid Value"  "dateOfBirth": "Invalid date format. Format must be yyyy-MM-dd and must be a valid date"  "memberStatus": "memberStatus cannot be blank." |
| {"code": 40001,  "message": “Validation failed for membership profile",  "errorMap":  {"membershipId“: “The supplied Member Number is not  valid"}} |
| {"code": 40002,  "message": “Validation failed for membership profile",  "errorMap":  {"firstName" : "Member First Name is missing"}} |
| {"code": 40003,  "message": “Validation failed for membership profile",  "errorMap":  {"lastName": "Member Surname is missing"}} |
| {"code": 40004,  "message": “Validation failed for membership profile",  "errorMap":  {"dateOfBirth" : "Member Date of Birth is Missing"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| 402  Payment Required | The addUpdateMemberDetails request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "Success"} |
| The updateMemberDetails request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has  expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for  this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 412  Precondition Failed | {"code": 41205,  "message": “The value supplied for gender is not valid"} |
| {"code": 41206,  "message": “The supplied card number is not valid"} |
| {"code": 41207,  "message": “Member Suspension Status is not  supplied/valid"} |
| {"code": 41217,  "message": “The value supplied for member status is not  valid"} |
| {"code": 41210,  "message": “The supplied member card number belongs to a  different member number"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |
| Exception when adding member. Contains the exception generated.  {"code": 50008,  "message": “Exception when inserting member:  XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member. Contains the exception generated.  {"code": 50009,  "message": “Exception when Updating member:  XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when adding member card. Contains the exception generated.  {"code": 50011,  "message": “Exception when adding card, Member No  :XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member card. Contains the exception generated. |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | {"code": 50012,  "message": “Exception when updating cards, Member no:XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating other information for member. Contains the exception generated.  {"code": 50013,  "message": “Exception when updating cards, Member no:XXXXXXXXXXXX. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50014,  "message": “Exception when updating the member status. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50015,  "message": “Error when updating member analysis. Error: NNNN, <Message>”} |
| Exception when updating member status. Contains the exception generated.  {"code": 50016,  "message": “Exception when attempting to update member analysis. Error: NNNN, <Message>”} |
| Exception when updating any of the locked address field(s). Venue can lock/ unlock address fields from S7000 Workstation.  {  "code": 50019,  "message": " Address3 Address4 are locked fields and cannot accept value"  } |
|  | Exception when supplied cardType doesn’t match with membershipType.  {  "code": 50022,  "message": "Validation failed for membership profile",  } |
|  | Exception when supplied cardType doesn’t match with the id of cardLayout.  {  "code": 50022,  "message": "The value supplied Card Layout does not match with card type."  } |

**Update or replace existing member’s details.**

The client request is required to send only the property values that are intended to be updated in the S7000 system with the supplied values for any given member.

The S7000 system will update the member details if the member exists; else an appropriate HTTP response is code and message will be returned by the S7000.

A new card is allocated to the supplied card number if the card is not present in S7000; and all other pre-allocated cards of the member will be deleted.

If the cardNumber property value is supplied, all pre-allocated cards of the member will be deleted.

The update request cannot be processed successfully if a card with the same Card Number already exists in S7000 and if it is allocated to a different member.

##### Allocate new active card number or activate an existing card for a member.

The client request is required to send all the property values every time when intending to allocate a new card or to activate an existing card for a member.

In order to allocate a new card or activate existing card, the client request has to send the cardNumber, cardType and the cardExpiryDate property.

Only one card will be allocated for a member at any time within the S7000 system. This will be the card number supplied by in the client request.

If the supplied card number is not present within the S7000, the card is allocated, and all other cards will be deleted. If cardNumber property value is not supplied, no changes will be done to the existing cards of the member within the S7000 system.

##### Delete cards associated with a member.

In order to delete cards associated with a member, the Membership system is required to send a zero (0) as the card number.

# ADD UPDATE MEMBER FACIAL IMAGE

Request sent from the client to add or update a member’s facial image within the S7000 system.

The client will need to provide a mapping text and the member image jpeg files.

The first line of the mapping text file is the header line that will contain the label text “MemberId, ImageFileName”.

Every line after the header line should contain the data i.e. member Id, followed by the image file name, separated by a comma.

In the instances where a mapping text file is not supplied the S7000 system will verify if the jpeg image file name itself is the MemberId. In such instances if the jpeg image file name is numeric and matches an existing MemberId within the S7000 system, the image will be imported and linked to the respective Member.

##### Request

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/export/addUpdateMemberImage api/membership/addUpdateMemberImage |

The client will need to place the image mapping file and the jpeg image files at the directory location in webserver i.e. /u/datain.

Access permissions to this folder by third parties is between Operations and the Venue (and the Venue and the Third Party).

**Sample Mapping and Image Files**



**Sample JPEG image file that has the MemberId as the file name**



**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The addUpdateMemberImage request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The addUpdateMemberImage request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The SQL execution resulted in no rows to be returned.  {"code": 40400,  "message": “Member XXXXX does not exist.”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"}  Returns NULL if successful else an error message.  In the event of the operation not being successful this field will be |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | populated with a message providing details of the error.  Returns the first error found. There may be several errors if the client is trying to add in multiple images. These errors will be logged into a log file which can be requested from the Aristocrat if required. |

REST API 3rd Party Interface Specification Version 1.29

# ADD UPDATE MEMBER IMAGE DIRECT

Request sent from the client to add or update a member’s facial image within the S7000 system.

The client will need to provide membership number and the image data as part of the request. The image data is supplied in the text format as shown in the sample request.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/membership/addUpdateMemberImageDirect |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| membershipId | Number | The membership number of the member the image is for. |
| image\_type | String | Indicates the image type e.g. jpg or png |
| Image\_data | String | The image in a text or string format. Refer to the Sample Request. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Response**



© Aristocrat Technologies Australia Pty Limited 2023. *All rights reserved.*

COMMERCIAL-IN-CONFIDENCE

{ "membershipId" : "347", "image\_type" : "image/jpg", "image\_data" :

"/9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofH

h0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjI yMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAB4A HgDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwI EAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJ

icoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJW Wl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09 fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQ AAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpK

jU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJm aoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/ 9oADAMBAAIRAxEAPwC9nNLTc0ZpEC0UmaQmgANIaCaQmgYGqkh+0T7B9xOT7mpLiXy48D7zcAU QRiNAP4jyTQGyJgMcUhozSZpCCikNJTAM0GkzRnigA/GikzRQAouHT/WRnHqORUqSo4+VgaSo2 hRjnGD6ii4XLGaTNVcTR/dbcPRqX7UAcSKUPr2oGTmkLbRk00OGGQQar3LlyIU6t1PoKAEjBnm Mx+6OFq1TEUIoUdBTqBbi0maM0hNAWCjpSUlABmkooNACUUZooAmzRTc0ZoGKTTWAIwRS5pCaA KlxGkSF1JVu2DTLcyR5kdC27vnmlfNzcAfwJVsDAwKQbaCJOj9G59DT81G8SP1UfWovLkT/AFb

/AINzQFyzmkPNV/tBU4lQj3HIqVZFcZVgaYD6SkooAKSjNIaQgNFJRQBLmgmow1LupjH5qvcyl E2r95uBUhcBST0FVoMzSmY9Oi0DLEMYijA79TUmabRnFIQ6m0maM0ABqJoEJyMqfVeKkozQKxD mePuJB+RpVuUJw2VPvUhIpjIrDDAEUBck3A9DSE1WMO05jYr/ACppuJI/9YuR6igC0SAMmiqTX KzsI0bAPU0UDJxKD0p4fNZzx3dv/rIiV9U5qI36qhO7n0qgL08xkcQqev3vpVlMIoUdBWXbSY+ djlmq4Jc96TQMtBuKC1QeZSh6LATBqN4HWolYBuTxSzN8uetICTcD0ozVZGIYips0wHZpM+9Jm kzSAU1RuXMz+SvTv71PcS7VCj7x6U23i2Lk/ePU0C2BbWMKAQM+tFTdqKRJfJB61WnsLa5/1kS k+vQ1PRVXHYypdIZOYJSPZqqSR3Vuf3kZI9V5FdBmkOCMEUXHqjAS7B71MtwD3rQnsbecfNGM+ o4NZ0ukyJkwS5Ho/wDjTuguSeaDTxLxjNZri4tz+9iYD+8ORQl0DxuoKNRWA5zTw9Zy3AqZZs9 6VhF3dkU15VjUk1X84AZJ4qJX8995+6OgPekBPEpZjLJ1PQelTZqNXp26gkfmio91FIDRozSGk 70xi5pCaDSUDFzTeKM0lAgKg1Un063nJLIA3qvBq1mkzQKxjyaRLGcwy7h6NVV1uID+8iYD1HI ros470hw3BFO4XZzQufNbaD8o61aSYY64FX59OtpjkxgN6rwaoS6TKnMMoI/ut/jRcLkyzU8S5 GQay2Nxb/62Jl9+opyXIPeiwzSeYqhI5xRVMTgjGaKVgudJRmiikAhptFFAxKQnIoopiE7UZoo oATNJmiigAzSUUUCGsoYYIqlPplvLkhNrHuvFFFIllGXTJo+Yn3D0PFFFFHMwUmf/2Q=="

}

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired. |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The addUpdateMemberImage request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The addUpdateMemberImage request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The SQL execution resulted in no rows to be returned.  {"code": 40400,  "message": “Member XXXXX does not exist.”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"}  Returns NULL if successful else an error message.  In the event of the operation not being successful this field will be populated with a message providing details of the error.  Returns the first error found. There may be several errors if the client is trying to add in multiple images. These errors will be logged into a log file which can be requested from the Aristocrat if required. |

REST API 3rd Party Interface Specification Version 1.29

# DELETE MEMBER

Request from the client to delete a member from the S7000 system.

Request to delete a member in S7000 will delete the member history, including cards, reward points and all the activity associated with that member.

Note: The client system has to ensure that the member details associated with a Member Number are not changed to another member without first deleting the original Member Number. Not doing so may result in the new person inheriting any remaining client points and play history of the original person.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/membership/deleteMember |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member who has to be deleted from the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "membershipId" : "12917" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message..  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.:  “memberId": "Invalid Value"  "membershipId": "membershipId cannot be blank." "membershipId": "membershipId must be greater than zero" |
| The membershipId property value is not valid.  {"code": 40000,  "message”: “Failed to Validate field membershipId", "errorMap":  {<field name> : <error message>}}  E.g.: "membershipId": "membershipId must be numeric" |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The deleteMember request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "Success"} |
| The deleteMember request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET MEMBER DETAILS

Request sent from the client to validate the supplied membership card number when a member swipes the card at a POS location. The request sends back complete member details on a successful card validation.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/getMemberDetails |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| cardNumber | Number(20) | Y | As existing S7000 membership card number |
| locationNumber | Number(10) | Y | The POS till or location number where the member has swiped the card at. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The membership number of the member these details belong to. |
| firstName | String | The member’s firstname or preferred name. |
| lastName | String | The member’s surname. |
| gender | String | This determines the gender of the member. Returns ‘MALE’, FEMALE’ or ‘UNKNOWN’ |
| title | String | The member’s title. |
| cardType | String | Type of the membership i.e. membership card type. |
| cardNumber | Number | Card number of the member |
| realName | String | The member’s associated name. |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| birthday | String | This will indicate whether the current date is the member’s birthday.  Returns ‘true’ if the current day is the member’s birthday or ‘false.’ |
| idType | Number | The number which represents the member’s ID documentType. The values supported are:  5 - DRIVERS LICENCE  10 - PASSPORT  15 - PROOF OF AGE CARD  20 - BIRTHCARD |
| idIssueDate | Date | Issue Date of member ID document |
| idExpiryDate | Date | Expiry Date of member ID document |
| idReference | String | An alphanumeric value of member’s ID document. |
| membershipType | Number | Member’s Membership Type |
| cardLayout | String | Member’s Card Layout |

**Sample Request**

{ "cardNumber" : "2509010012", "locationNumber" : "9" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "10012", "firstName": "JOHN VICTOR", "lastName": "MARTIN", "gender": "MALE",

"title": "MR", "cardType": "ZAP MEMBER", "cardNumber": 2509010012, "realName": "JOHN VICTOR",

"birthday": "false", "idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"membershipType": 30,

"cardLayout": "PMFF5"

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “cardNum”: "Invalid Value" |
| The cardNumber or the locationNumber request property value is not valid.  {"code": 40000,  "message”: “Validation failed for get member failed", "errorMap":  {<field name> : <error message>}} E.g.:  “locationNumber” : “locationNumber cannot be blank” "cardNumber": "cardNumber must be numeric" "locationNumber": "locationNumber must be numeric" "cardNumber": "cardNumber cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberDetails request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "Success"} |
| The getMemberDetails request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | Could not fetch the member details as the card number is not registered or invalid in the S7000 system.  {"code": 40400,  "message": "No details found for card number: XXXXXXXXX"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET MEMBER DETAILS BY CARD OR MEMBER

Request sent from the client to validate the supplied membership or the card number and to fetch the member details. The request sends back the complete member details on a successful validation.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/getFullMemberDetails |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| cardNumber | Number(20) | Y | As existing S7000 membership card number |
| membershipId | Number(12) | Y | As existing S7000 membership number |
| locationNumber | Number(10) | Y | The POS till or location number where the member has swiped the card at. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| cardNumber | Number | Card number of the member |
| membershipId | Number | The membership number of the member these details belong to. |
| firstName | String | The member’s firstname or preferred name. |
| lastName | String | The member’s surname. |
| displayName | String | The member’s display name. |
| title | String | The member’s title. |
| gender | String | This determines the gender of the member. Returns ‘MALE’, FEMALE’ or ‘UNKNOWN’ |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| dateOfBirth | String | Date of birth |
| telephone | String | Home or primary telephone number of the member as recorded in S7000. |
| mobile | String | Mobile phone number of the member if recorded in S7000. |
| badgeNumber | String | Member’s badge number if recorded in S7000. |
| cardExpiryDate | Date | Expiry date of the membership card that is associated with the member. |
| memAnalysisList | Two- dimensional user defined string array | analysisId: Unique Identifier of the Analysis Attribute Type  analysisDesc: Name of the Analysis Attribute type.  E.g.: Tiering  analysisClassCode: Unique Identifier of the Analysis Attribute selected for the member.  analysisClassDesc: The value of the Analysis Attribute selected for the member. E.g.: GOLD MEMBER |
| idType | Number | The number which represents the member’s ID documentType. The values supported are:  5 - DRIVERS LICENCE  10 - PASSPORT  15 - PROOF OF AGE CARD  20 - BIRTHCARD |
| idIssueDate | Date | Issue Date of member ID document |
| idExpiryDate | Date | Expiry Date of member ID document |
| idReference | String | An alphanumeric value of member’s ID document. |
| membershipType | Number | Member’s Membership Type |
| cardLayout | String | Member’s Card Layout |

**Sample Request**

{ "cardNumber" : "2509010012", "membershipId" : "12917", "locationNumber" : "9" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId" : "12917",

"cardNumber": 2509010012, "firstName": "JOHN VICTOR", "lastName": "MARTIN", "displayName": "JOHN VICTOR", "title": "MR",

"gender": "MALE", "dateOfBirth": "1970-01-01",

"telephone": "0400001000",

"mobilePhone": "0400001000",

"badgeNumber": "1000",

"cardExpiryDate": "2090-06-01", "memAnalysisList":[

{

"analysisId": 0,

"analysisDesc": "MEMBERSHIP TYPE", "analysisClassCode": 20,

"analysisClassDesc": "5 YEAR"

},

{

"analysisId": 1, "analysisDesc": "TIER", "analysisClassCode": 50, "analysisClassDesc": "GOLD"

}

],

"idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"membershipType": 30, "cardLayout": "PMFF5"

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “cardNum”: "Invalid Value" |
| The cardNumber or the locationNumber request property value is not valid.  {"code": 40000,  "message”: “Validation failed for get member failed", "errorMap":  {<field name> : <error message>}} E.g.:  “locationNumber” : “locationNumber cannot be blank” "cardNumber": "cardNumber must be numeric" "locationNumber": "locationNumber must be numeric" "cardNumber": "cardNumber cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getFullMemberDetails request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getFullMemberDetails request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404 | Could not fetch the member details as the card number is not |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| Not Found | registered or invalid in the S7000 system.  {"code": 40400,  "message": "No details found for card number: XXXXXXXXX"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET A MEMBER IMAGE

Request sent from the client to send back a member facial image from the S7000 system.

The member facial image is sent in the response itself when the member image was successfully fetched from the S7000.

Also two additional files will be generated by this request that is a mapping text and the member image jpeg file. These will be placed at the directory location designated by Aristocrat.

The first line of the mapping text file is the header line that will contain the label text “MemberId, ImageFileName”. And the line after is the data i.e. member Id, followed by the image file name, separated by a comma.

##### Request

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/export/getMemberImage |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Re qd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member to retrieve the image for. MembershipId should be an existing member membership number, cannot be NULL or 0. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| code\* | Number | Refer to HTTP Response Codes and Messages |
| message\* | String | Refer to HTTP Response Codes and Messages |

\*These properties may not be returned if the response body contains the member image fetched from the S7000.

**Sample Request**

{ "membershipId" : "1" }

**Sample Response**

{ "code": 40400,

"message": "No record found for membershipId XXXXXXXX" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the image for the requested membershipId. |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| The request property name membershipId is supplied incorrectly.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}}  E.g.: “memberId": "Invalid Value" |
| The membershipId property value is not valid.  {"code": 40000,  "message”: “Validation failed for get member image", "errorMap":  {<field name> : <error message>}} E.g.:  “membershipId": "Invalid Value"  "membershipId": "membershipId must be numeric" "membershipId": "membershipId cannot be blank." "membershipId": "membershipId must be greater than zero" |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberImage request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getMemberImage request was NOT successful as the client |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The member does not have a facial image captured within the S7000 system.  {"code": 40400,  "message": No record found for membershipId xxxxxxxxxx”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# GET A MEMBER IMAGE DIRECT

Request sent from the client to send back a member facial image from the S7000 system.

The member facial image text is sent in the response itself when the member image was successfully fetched from the S7000.

Also two additional files will be generated by this request that is a mapping text and the member image jpeg file. These will be placed at the directory location designated by Aristocrat.

The first line of the mapping text file is the header line that will contain the label text “MemberId, ImageFileName”. And the line after is the data i.e. member Id, followed by the image file name, separated by a comma.

##### Request

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/membership/getMemberImageDirect |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Re qd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member to retrieve the image for. MembershipId should be an existing member membership number, cannot be NULL or 0. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| code\* | Number | Refer to HTTP Response Codes and Messages |
| message\* | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The membership number of the member the image is for. |
| image\_type | String | Indicates the image type e.g. jpg or png |
| Image\_data | String | The image in a text or string format |

\*These properties may not be returned if the response body contains the member image fetched from the S7000.

**Sample Request**

{ "membershipId" : "1" }

**Sample Response**

{ "code": 40400,

"message": "No record found for membershipId XXXXXXXX" }

**Sample Response**

{ "code": 0, "message": null,

"membershipId" : "347", "image\_type" : "image/jpg", "image\_data" :

"/9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofH

h0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjI yMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjL/wAARCAB4A HgDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwI EAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJ

icoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJW Wl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09 fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQ AAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpK

jU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJm aoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/ 9oADAMBAAIRAxEAPwC9nNLTc0ZpEC0UmaQmgANIaCaQmgYGqkh+0T7B9xOT7mpLiXy48D7zcAU QRiNAP4jyTQGyJgMcUhozSZpCCikNJTAM0GkzRnigA/GikzRQAouHT/WRnHqORUqSo4+VgaSo2 hRjnGD6ii4XLGaTNVcTR/dbcPRqX7UAcSKUPr2oGTmkLbRk00OGGQQar3LlyIU6t1PoKAEjBnm Mx+6OFq1TEUIoUdBTqBbi0maM0hNAWCjpSUlABmkooNACUUZooAmzRTc0ZoGKTTWAIwRS5pCaA KlxGkSF1JVu2DTLcyR5kdC27vnmlfNzcAfwJVsDAwKQbaCJOj9G59DT81G8SP1UfWovLkT/AFb

/AINzQFyzmkPNV/tBU4lQj3HIqVZFcZVgaYD6SkooAKSjNIaQgNFJRQBLmgmow1LupjH5qvcyl E2r95uBUhcBST0FVoMzSmY9Oi0DLEMYijA79TUmabRnFIQ6m0maM0ABqJoEJyMqfVeKkozQKxD mePuJB+RpVuUJw2VPvUhIpjIrDDAEUBck3A9DSE1WMO05jYr/ACppuJI/9YuR6igC0SAMmiqTX KzsI0bAPU0UDJxKD0p4fNZzx3dv/rIiV9U5qI36qhO7n0qgL08xkcQqev3vpVlMIoUdBWXbSY+ djlmq4Jc96TQMtBuKC1QeZSh6LATBqN4HWolYBuTxSzN8uetICTcD0ozVZGIYips0wHZpM+9Jm kzSAU1RuXMz+SvTv71PcS7VCj7x6U23i2Lk/ePU0C2BbWMKAQM+tFTdqKRJfJB61WnsLa5/1kS k+vQ1PRVXHYypdIZOYJSPZqqSR3Vuf3kZI9V5FdBmkOCMEUXHqjAS7B71MtwD3rQnsbecfNGM+ o4NZ0ukyJkwS5Ho/wDjTuguSeaDTxLxjNZri4tz+9iYD+8ORQl0DxuoKNRWA5zTw9Zy3AqZZs9 6VhF3dkU15VjUk1X84AZJ4qJX8995+6OgPekBPEpZjLJ1PQelTZqNXp26gkfmio91FIDRozSGk 70xi5pCaDSUDFzTeKM0lAgKg1Un063nJLIA3qvBq1mkzQKxjyaRLGcwy7h6NVV1uID+8iYD1HI ros470hw3BFO4XZzQufNbaD8o61aSYY64FX59OtpjkxgN6rwaoS6TKnMMoI/ut/jRcLkyzU8S5 GQay2Nxb/62Jl9+opyXIPeiwzSeYqhI5xRVMTgjGaKVgudJRmiikAhptFFAxKQnIoopiE7UZoo oATNJmiigAzSUUUCGsoYYIqlPplvLkhNrHuvFFFIllGXTJo+Yn3D0PFFFFHMwUmf/2Q=="

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the image for the requested membershipId. |
| 400 | The HTTP Header may be in an incorrect format.  {"code": 40000, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| Bad Request | "message": "Invalid input data"} |
| The request property name membershipId is supplied incorrectly.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}}  E.g.: “memberId": "Invalid Value" |
| The membershipId property value is not valid.  {"code": 40000,  "message”: “Validation failed for get member image", "errorMap":  {<field name> : <error message>}} E.g.:  “membershipId": "Invalid Value"  "membershipId": "membershipId must be numeric" "membershipId": "membershipId cannot be blank." "membershipId": "membershipId must be greater than zero" |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberImage request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getMemberImage request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "The table or view is not accessible"} |
| 404  Not Found | The member does not have a facial image captured within the S7000 system.  {"code": 40400,  "message": No record found for membershipId xxxxxxxxxx”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET MEMBER ANALYSIS DETAILS

Request sent from the client to send back member analysis details for the supplied membership number. Member Analysis details are additional details captured for any member within the S7000 system. These details may contain information such as member’s tier level and other member preferences.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | **api/pos/getMemberAnalysisDetails** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member whose details are being requested for. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The membership number of the member the details belong to. |
| analysis1 | String (20) | The member analysis detail 1. |
| analysis2 | String (20) | The member analysis detail 2. |
| analysis3 | String (20) | The member analysis detail 3. |
| analysis4 | String (20) | The member analysis detail 4. |
| analysis5 | String (20) | The member analysis detail 5. |
| analysis6 | String (20) | The member analysis detail 6. |
| analysis7 | String (20) | The member analysis detail 7. |
| analysis8 | String (20) | The member analysis detail 8. |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| analysis9 | String (20) | The member analysis detail 9. |
| analysis10 | String (20) | The member analysis detail 10. |

**Sample Request**

{ "membershipId" : "1" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "1", "analysis1": null, "analysis2": null, "analysis3": "GOLD", "analysis4": "KARDIS", "analysis5": null, "analysis6": null, "analysis7": null, "analysis8": null, "analysis9": null, "analysis10": "Post" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| The membershipId property name of the request is incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| The membershipId property value is not valid.  {"code": 40000,  "message”: “Failed to validate field membershipId", "errorMap": |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" "membershipId": "membershipId cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberAnalysisDetails request was ssuccessful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getMemberAnalysisDetails request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | Could not fetch the details as the membershipId is not registered or invalid in the S7000 system.  {"code": 40400,  "message": "No details found for membership Id: XXXXX"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET MEMBER PLAYING AT EGM

Request sent from the client to send back the details of the member who is currently playing on the supplied gaming machine number.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/getMemberAtEgm |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| egmId | String (64) | Y | The EGM Id to be supplied for enquiry about the member currently playing on.  EGM Id should be an existing S7000 machine name. This property value is case sensitive and should match the name of the EGM as configured within the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The membership number of the member currently playing on the EGM being enquired for. Returns a value if successful else NULL.  NOTE: The presence of a member, as returned from the S7000 system may not be the current situation at the EGM. |
| firstName | String | The member’s first name or preferred name. |
| lastName | String | The member’s surname. |
| displayName | String | The member’s display name if recorded in S7000. |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| title | String | The member’s title. |
| gender | String | This determines the gender of the member. Returns ‘MALE’, FEMALE’ or ‘UNKNOWN’ |
| dateOfBirth | Date | The member’s date of birth. |
| telephone | String | Home or primary telephone number of the member as recorded in S7000. |
| mobilePhone | String | Mobile phone number of the member if recorded in S7000. |
| badgeNumber | String | Member’s badge number if recorded in S7000. |
| memberStatus | String | The current membership status of the member. Returns one of the following if successful else NULL:   * PENDING * FINANCIAL * UNFINANCIAL * RETIRED/DECEASED |
| analysis1 | String | The member analysis detail 1. |
| analysis2 | String | The member analysis detail 2. |
| analysis3 | String | The member analysis detail 3. |
| analysis4 | String | The member analysis detail 4. |
| analysis5 | String | The member analysis detail 5. |
| analysis6 | String | The member analysis detail 6. |
| analysis7 | String | The member analysis detail 7. |
| analysis8 | String | The member analysis detail 8. |
| analysis9 | String | The member analysis detail 9. |
| analysis10 | String | The member analysis detail 10. |

**Sample Request**

{ "egmId" : "2901" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "10012", "firstName": "JOHN VICTOR", "lastName": "MARTIN", “displayName”: “JOHN”, "title": "MR",

"gender": "MALE", "dateOfBirth" : "1986-11-11",

"telephone": "988998899",

"mobilePhone": "041212121",

"badgeNumber": "54321", "memberStatus": "PENDING", "analysis1": null, "analysis2": null, "analysis3": "GOLD", "analysis4": "KARDIS", "analysis5": null, "analysis6": null, "analysis7": null, "analysis8": null, "analysis9": null, "analysis10": "Post" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| The egmId property name of the request is incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “egId": "Invalid Value" |
| The egmId request property value is not valid.  {"code": 40000,  "message”: “Failed to validate field egmId", "errorMap":  {"egmId": "egmId cannot be blank."}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberAtEgm request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getMemberAtEgm request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | No member is currently playing on the supplied EGM.  {"code": 40400,  "message": "No Member is currently playing on the egm"} |
| The supplied egmId could not be validated in the S7000 system.  {"code": 40400,  "message": "The supplied egm is not valid."} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# CHECK MEMBER PIN

Request sent from the client to validate a member’s PIN.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/checkMemberPin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member that the PIN is to be validated for.  Can be blank if cardNumber is being supplied. |
| cardNumber | Number(20) | Y | The card number of the member that the PIN is to be validated for.  Can be blank if membershipId is being supplied. |
| pin | String | Y | The pin to be validated against this member. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "membershipId" : "", "cardNumber" : "2509010012", "pin" : "1234" }

**Sample Response**



© Aristocrat Technologies Australia Pty Limited 2023. *All rights reserved.*

COMMERCIAL-IN-CONFIDENCE

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| The membershipId and/or pin property value is not valid.  {"code": 40000,  "message”: “Validation failed for check PIN", "errorMap":  {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" |
| The supplied PIN is incorrect for the membershipId  {"code": 40003,  "message": “Validation failed for check PIN", "errorMap":  {"pin" : "Wrong PIN"}} |
| The basic request properties haven’t been supplied  {"code": 40010,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"Please Supply Membership or Card Number"}} |
| The PIN for validation is not supplied  {"code": 40011,  "message": “Validation failed for check PIN", "errorMap":  {"pin" : "Please Supply the Pin to Validate"}} |
| The PIN for validation is not supplied  {"code": 40012,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"XXXXXXXXXX supplied card number is not valid"}} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | HTTP Header is missing Authorization  {"code": 40013,  "message": “Validation failed for check PIN", "errorMap":  {"membershipId" : "XXXXXXXXXX supplied member number is not valid"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The checkMemberPIN request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The checkMemberPIN request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied member number does not have a PIN defined.  {"code": 404xx,  "message": <error message>} |
| {"code": 40402,  "message": "No Pin associated with that member"} |
| {"code": 40405,  "message": “No PIN associated with that member. Use  last 4 digits of mem number."} |
| {"code": 40406,  "message": “No PIN associated with that member. Use the |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | PIN 'xxxx'"} |
| 412  Precondition Failed | {"code": 41204,  "message": “PIN retries exceeded"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# RESET MEMBER PIN

Request sent from the client to reset a member’s PIN. The PIN will be reset to the supplied pin. If the pin is not supplied the member pin will be reset to the default pin configured within the S7000 system.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/resetMemberPin |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member that the PIN is to be validated for.  Can be blank if cardNumber is being supplied. |
| cardNumber | Number(20) | Y | The card number of the member that the PIN is to be validated for.  Can be blank if membershipId is being supplied. |
| pin | String | Y | The new pin that that the member’s pin needs to be reset to.  If no value is supplied for this pin property the member pin will be reset to the default pin that is configured within the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "membershipId" : "12917", "cardNumber" : "",

"pin" : "1234" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| The membershipId and/or pin property value is not valid.  {"code": 40000,  "message”: “Validation failed for reset PIN", "errorMap":  {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" |
| The supplied PIN is incorrect for the membershipId  {"code": 40003,  "message": “Validation failed for rest PIN", "errorMap":  {"pin" : "Wrong PIN"}} |
| The basic request properties haven’t been supplied  {"code": 40010,  "message": “Validation failed for reset PIN", "errorMap":  {"cardNumber" :"Please Supply Membership or Card Number"}} |
| The PIN for validation is not supplied  {"code": 40011,  "message": “Validation failed for reset PIN", "errorMap":  {"pin" : "Please Supply the Pin to reset"}} |
| The PIN for validation is not supplied  {"code": 40012,  "message": “Validation failed for reset PIN", |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "errorMap":  {"cardNumber" :"XXXXXXXXXX supplied card number is not valid"}} |
| HTTP Header is missing Authorization  {"code": 40013,  "message": “Validation failed for check PIN", "errorMap":  {"membershipId" : "XXXXXXXXXX supplied member number is not valid"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The resetMemberPIN request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The resetMemberPIN request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied member number does not have a PIN defined.  {"code": 404xx,  "message": <error message>} |
| {"code": 40402,  "message": "No Pin associated with that member"} |
| {"code": 40405,  "message": “No PIN associated with that member. Use |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | last 4 digits of mem number."} |
| {"code": 40406,  "message": “No PIN associated with that member. Use the  PIN 'xxxx'"} |
| 412  Precondition Failed | {"code": 41204,  "message": “PIN retries exceeded"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# VALIDATE MEMBER AND PIN

Request sent from the client to validate a member’s PIN.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/validateMemberandPIN |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| cardNumber | Number(20) | Y | The card number of the member that the PIN is to be validated for.  Can be blank if membershipId is being supplied. |
| locationNumber | Number(10) | Y | The POS till or location number where the member has swiped the card at. |
| memberNumber | Number(12) | Y | The membership number of the member that the PIN is to be validated for.  Can be blank if cardNumber is being supplied. |
| pin | String | Y | The pin to be validated against this member. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{

"cardNumber" : "2000074666",

"locationNumber" : "99", "memberNumber":250, "pin":"0000"

}

**Sample Response**

{

"licenseStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "memberNumber": "250",

"cardNumber": "2000074666", "firstName": "WKPPKKO", "surName": "AXCCMJ", "displayName": "WKPPKKO", "title": "MR",

"gender": "MALE", "dateOfBirth": "1936-02-27",

"telephone": "1850505899", "mobilePhone": null, "badgeNumber": "250",

"cardExpiryDate": "2022-06-30", "memAnalysisList":[

{"analysisId": 0, "analysisDesc": "MEMBER TYPE", "analysisClassCode":

30, "analysisClassDesc": "PMFF5"…},

{"analysisId": 1, "analysisDesc": "TIERING", "analysisClassCode": 35,

"analysisClassDesc": null…},

{"analysisId": 2, "analysisDesc": "Analysis 3", "analysisClassCode": 25,

"analysisClassDesc": "OTHER ASIAN"…},

{"analysisId": 3, "analysisDesc": "Analysis 4", "analysisClassCode": 30,

"analysisClassDesc": "COCKTAILS"…}], "promotionsList": null, "nonSwipePromotionsList": null, "promoAlreadyQualList": null, "dateFirstRegistered": null, "personId": 0,

"idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"membershipType": 30, "cardLayout": "PMFF5"

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| The membershipId and/or pin property value is not valid.  {"code": 40000,  "message”: “Validation failed for check PIN", "errorMap":  {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" |
| The supplied PIN is incorrect for the membershipId  {"code": 40003,  "message": “Validation failed for check PIN", "errorMap":  {"pin" : "Wrong PIN"}} |
| The basic request properties haven’t been supplied  {"code": 40010,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"Please Supply Membership or Card Number"}} |
| The PIN for validation is not supplied  {"code": 40011,  "message": “Validation failed for check PIN", "errorMap":  {"pin" : "Please Supply the Pin to Validate"}} |
| The PIN for validation is not supplied  {"code": 40012,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"XXXXXXXXXX supplied card number is not valid"}} |
| HTTP Header is missing Authorization  {"code": 40013,  "message": “Validation failed for check PIN", "errorMap":  {"membershipId" : "XXXXXXXXXX supplied member number is not valid"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The checkMemberPIN request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The checkMemberPIN request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied member number does not have a PIN defined.  {"code": 404xx,  "message": <error message>} |
| {"code": 40402,  "message": "No Pin associated with that member"} |
| {"code": 40405,  "message": “No PIN associated with that member. Use  last 4 digits of mem number."} |
| {"code": 40406,  "message": “No PIN associated with that member. Use the  PIN 'xxxx'"} |
| 412  Precondition Failed | {"code": 41204,  "message": “PIN retries exceeded"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET CARDHOLDER DETAILS

Request sent from the client to validate a member’s PIN.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/getcardholderdetails |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| cardNumber | Number(20) | Y | The card number of the member that the PIN is to be validated for.  Can be blank if membershipId is being supplied. |
| locationNumber | Number(10) | Y | The POS till or location number where the member has swiped the card at. |
| memberNumber | Number(12) | Y | The membership number of the member that the PIN is to be validated for.  Can be blank if cardNumber is being supplied. |
| pin | String | Y | The pin to be validated against this member. |
| groupId | Number(2) | N |  |
| hasPrinter | Number(1) | N |  |
| includeNonSwipeP romo | Number(1) | N |  |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| cardNumber | Number | Card number of the member |
| memberNumber | Number | The membership number of the member these details belong to. |
| firstName | String | The member’s firstname or preferred name. |

|  |  |  |
| --- | --- | --- |
| surName | String | The member’s surname. |
| displayName | String | The member’s display name. |
| title | String | The member’s title. |
| gender | String | This determines the gender of the member.  Returns ‘MALE’, FEMALE’ or ‘UNKNOWN’ |
| dateOfBirth | String | Date of birth |
| telephone | String | Home or primary telephone number of the member as recorded in S7000. |
| mobilePhone | String | Mobile phone number of the member if recorded in S7000. |
| badgeNumber | String | Member’s badge number if recorded in S7000. |
| cardExpiryDate | Date | Expiry date of the membership card that is associated with the member. |
| memAnalysisList | Two- dimensional user defined string array | analysisId: Unique Identifier of the Analysis Attribute Type  analysisDesc: Name of the Analysis Attribute type.  E.g.: Tiering  analysisClassCode: Unique Identifier of the Analysis Attribute selected for the member.  analysisClassDesc: The value of the Analysis Attribute selected for the member. E.g.: GOLD MEMBER |
| promotionsList | Two- dimensional user defined string array | List of swipe promotions awarded to the member on card swipe. |
| nonSwipePromotionsLi st | Two- dimensional user defined string array | List of non-swipe promotions awarded to the member |
| promoAlreadyQualList | Two- dimensional user defined string array | List of promotion already awarded to the member |
| dateFirstRegistered | Date |  |
| personId | Number |  |
| idType | Number | The number which represents the member’s ID documentType. The values supported are:  5 - DRIVERS LICENCE  10 - PASSPORT |

|  |  |  |
| --- | --- | --- |
|  |  | 15 - PROOF OF AGE CARD  20 - BIRTHCARD |
| idIssueDate | Date | Issue Date of member ID document |
| idExpiryDate | Date | Expiry Date of member ID document |
| idReference | String | An alphanumeric value of member’s ID document. |
| membershipType | Number | Member’s Membership Type |
| cardLayout | String | Member’s Card Layout |

**Sample Request**

{

"locationNumber" : "99", "groupId":2, "memberNumber":250, "cardNumber" : "2000074666",

"includeNonSwipePromo" : "0", "hasPrinter":1}

**Sample Response**

{

"licenseStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "memberNumber": "250",

"cardNumber": "2000074666", "firstName": "WKPPKKO", "surName": "AXCCMJ", "displayName": "WKPPKKO", "title": "MR",

"gender": "MALE", "dateOfBirth": "1936-02-27",

"telephone": "1850505899", "mobilePhone": null, "badgeNumber": "250",

"cardExpiryDate": "2022-06-30", "memAnalysisList":[

{"analysisId": 0, "analysisDesc": "MEMBER TYPE", "analysisClassCode":

30, "analysisClassDesc": "PMFF5"…},

{"analysisId": 1, "analysisDesc": "TIERING", "analysisClassCode": 35,

"analysisClassDesc": null…},

{"analysisId": 2, "analysisDesc": "Analysis 3", "analysisClassCode": 25,

"analysisClassDesc": "OTHER ASIAN"…},

{"analysisId": 3, "analysisDesc": "Analysis 4", "analysisClassCode": 30,

"analysisClassDesc": "COCKTAILS"…}],

"promotionsList": null,

"nonSwipePromotionsList": null, "promoAlreadyQualList": null, "dateFirstRegistered": null, "personId": 0,

"idType": 15,

"idIssueDate": "2017-06-26",

"idExpiryDate": "2022-06-25",

"idReference": "140001256",

"membershipType": 30, "cardLayout": "PMFF5"

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| The membershipId and/or pin property value is not valid.  {"code": 40000,  "message”: “Validation failed for check PIN", "errorMap":  {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" |
| The supplied PIN is incorrect for the membershipId  {"code": 40003,  "message": “Validation failed for check PIN", "errorMap":  {"pin" : "Wrong PIN"}} |
| The basic request properties haven’t been supplied  {"code": 40010,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"Please Supply Membership or Card Number"}} |
| The PIN for validation is not supplied  {"code": 40011,  "message": “Validation failed for check PIN", |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "errorMap":  {"pin" : "Please Supply the Pin to Validate"}} |
| The PIN for validation is not supplied  {"code": 40012,  "message": “Validation failed for check PIN", "errorMap":  {"cardNumber" :"XXXXXXXXXX supplied card number is not valid"}} |
| HTTP Header is missing Authorization  {"code": 40013,  "message": “Validation failed for check PIN", "errorMap":  {"membershipId" : "XXXXXXXXXX supplied member number is not valid"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The checkMemberPIN request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The checkMemberPIN request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied member number does not have a PIN defined.  {"code": 404xx,  "message": <error message>} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | {"code": 40402,  "message": "No Pin associated with that member"} |
| {"code": 40405,  "message": “No PIN associated with that member. Use  last 4 digits of mem number."} |
| {"code": 40406,  "message": “No PIN associated with that member. Use the  PIN 'xxxx'"} |
| 412  Precondition Failed | {"code": 41204,  "message": “PIN retries exceeded"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET MEMBER POINT BALANCE

Request sent from the client to send back the reward point balance for the supplied member number and the reward point type properties.

This request is to be sent by the client as a pre-redemption balance check every time the client system intends to redeem member reward points from the S7000 system. This ensures that the member reward point balances do not go negative after redemption.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/getMemberPointBalance |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member for reward point balance enquiry.  This is the membership number to use when awarding and redeeming reward points. |
| pointType | String (6) | Y | The Reward Point type i.e. S7000 Reward Type of the reward points to be enquired.  E.g. BONUS.  If this property value is supplied as an S7000 Reward Point Type like in the example above, the S7000 system will return the points balance of the supplied Reward point type.  This property value can be supplied as “MULTI” if the S7000 is required to return the total balance of the multiple reward points available for redemption as configured in the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| membershipId | Number | The membership number of the member. |
| pointType | String | The S7000 reward point type for which the balance is enquired for. E.g. BONUS.  This will be returned as MULTI if the request property was supplied as “MULTI” |
| quantity | Number | Current point balance of this reward point type.  This property value will be the total balance of the multiple reward points available for redemption as configured in the S7000 system if “MULTI” if was supplied in the request for the pointType property. |
| description | String | The description of the reward point type as defined in S7000 system. |
| balance | Number | This value could be same as the points property value. Caters for the future implementation where the available points are different to the point balance. |

**Sample Request**

{ "membershipId" : "1", "pointType" : "BONUS"}

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "1", "pointType": "BONUS", "quantity": 2013.6, "description": "Bonus Points", "balance": 2013.6 }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400 | The HTTP Header may be in an incorrect format.  {"code": 40000, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| Bad Request | "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| Some of the request properties values are not valid.  {"code": 40000,  "message”: “Validation failed for member point balance", "errorMap":  {<field name> : <error message>}} E.g.:  "membershipId": "membershipId must be numeric" "membershipId": "membershipId cannot be blank." "pointType": "pointType cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getMemberPointBalance request was successful; however, the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getMemberPointBalance request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404 | The supplied membershipId and pointType property values not found |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| Not Found | in the S7000 system.  {"code": 40400,  "message": <error message>} |
| {"code": 40400,  "message": " Member Number not found in database "} |
| {"code": 40400,  "message”: “Invalid Reward Type"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# REDEEM MEMBER POINTS

Request sent from the client to redeem reward points of an existing member from the S7000 system.

The pre-requisite for this request is the getMemberPointBalance. The getMemberPointBalance request is to be sent by the client as a pre-redemption check to verify if the member has enough reward points balance for redemption. This is to ensure that the member reward point balance does not go negative.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/redeemMemberPoints |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member for reward point redemption. |
| pointType | String (6) | Y | The Reward point type i.e. S7000 Reward point Type of the reward points to be inquired.  E.g. BONUS.  If this property value is supplied as an S7000 Reward Point Type like in the example above, the S7000 system will redeem the points from the supplied Reward point type balance. In the instances where the supplied Reward point type does not have enough balance the response to this request will be sent as an http error code and message.  This property value can be supplied as “MULTI” if the redemption order of the rewards is to be determined by the S7000. In the instances where “MULTI” is supplied the S7000 system will redeem the points from the multiple reward point types if the requested point redemption quantity is not available in a single reward bucket. |
| points | Number(24,4) | Y | The quantity of reward points to be redeemed. |
| sourceId | String (10) | Y | A string to supply the userId used in the startWebSession E.g. “TESTBIGT” |
| terminalId | Number(10) | Y | The terminal Id of the POS location where the |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | redemption of reward points is initiated from. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| membershipId | Number | The membership number of the member whose points have been redeemed. |
| pointsRedeemed | Number | The number of the reward points redeemed. |
| transactionId | Number | Transaction number of redemption transaction. |

**Sample Request**

{ "membershipId" : "1", "pointType" : "BONUS", "points" : "12", "sourceId" : "TESTBIGT", "terminalId" : "1234" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "1",

"pointsRedeemed": 12.0,

"transactionId": 5447527 }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| Some of the request property names are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “SourcId": "Invalid Value" |
| The “points” request property value is invalid.  {"code": 40000,  "message":"Invalid Reward points"} |
| Some of the required property values are missing or invalid.  {"code": 40000,  "message”: “Validation failed for redeem points", "errorMap":  {<field name> : <error message>}} E.g.:  "pointType": "pointType cannot be blank." "membershipId": "membershipId must be numeric" "points": "points cannot be a negative value" "terminalId": "terminalId must be numeric" "terminalId": "terminalId must be greater than zero" "membershipId": "membershipId cannot be blank." "points": "points cannot be blank."  "terminalId": "terminalId cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The redeemMemberPoints request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The redeemMemberPoints request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31", |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied membershipId and pointType property values not found in the S7000 system.  {"code": 40400,  "message": <error message>} |
| {"code": 40400,  "message": "Member Number not found in database"} |
| {"code": 40400,  "message":"Invalid Reward Type"} |
| 412  Precondition Failed | {"code": 41200,  "message":"Not enough Points for redemption "} |
| {"code": 41200,  "message":"Not enough values"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# AWARD MEMBER POINTS

Request sent from the client to award reward points to an S7000 member.

Once the reward points are awarded, these reward points are available to be redeemed by the S7000 system or by the client application. It is the responsibility of the client system to determine the number of the reward points to be awarded to each member.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/awardMemberPoints |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| membershipId | Number(12) | Y | The membership number of the member for awarding reward points. |
| pointType | String (6) | Y | The Reward point type i.e. S7000 Reward point Type of the reward points to be inquired.  E.g. BONUS. |
| points | Number(24,4) | Y | The quantity of points to be awarded to the member. |
| terminalId | Number(10) | Y | The terminal Id of the POS location where the award of points is initiated from. |
| device | String (10) | Y | A string to indicate the system or device doing the award. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| membershipId | String | The membership number of the member who has |

|  |  |  |
| --- | --- | --- |
|  |  | been awarded the points. |
| pointsAwarded | Number | The number of reward points awarded. |
| totalPoints | Number | The resultant balance of the member’s reward points. |

**Sample Request**

{ "membershipId" : "1", "pointType" : "BONUS", "points" : "50",

"terminalId" : "12345678", "device" : "TESTBIGT" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "membershipId": "1",

"pointsAwarded": 50.0,

"totalPoints": 2013.6 }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “memberId": "Invalid Value" |
| Some of the request property values are invalid.  {"code": 40000,  "message”: “Validation failed for award member points", "errorMap":  {<field name> : <error message>}} E.g.:  “terminalId": "terminalId must be numeric" |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "membershipId": "membershipId must be numeric" "points": "points must be numeric"  "points": "points cannot be a negative value" "membershipId": "membershipId cannot be blank." "pointType": "pointType cannot be blank." "terminalId": "terminalId cannot be blank." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The awardMemberPoints request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The awardMemberPoints request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | {"code": 40400,  "message": "Member Number not found in database"} |
| {"code": 40400,  "message”: “Invalid Reward Type"} |
| {"code": 40400,  "message”: “Invalid Reward points"} |
| 412  Precondition Failed | Some of the request properties are invalid.  {"code": 41200,  "message":" Not enough values"} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# VERIFY POS ORDER TIMESTAMP

Request sent from the client to enquire for the latest timestamp of the POS order captured within the S7000 system.

This is the POS order message recorded and forwarded by the S7000 system when a member has placed a POS order from the EGM console.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/pos/verifyOrderMsgTimestamp |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| lastTime | String formatted YYYY- MM-DD HH:mm:S S | Y | Date/time of the last player service order/request received from S7000 |
| egmId | String (64) | Y | The EGM Id to be supplied for enquiry about the member currently playing on. EGM Id should be an existing S7000 machine name, cannot be NULL or 0. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| Code | Number | Refer to HTTP Response Codes and Messages |
| Message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "lastTime" : "2015-10-01 10:12:00",

"egmId": "2901" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “egId": "Invalid Value" |
| Some of the request property values are invalid.  {"code": 40000,  "message”: "Validation failed for verify order message timestamp",  "errorMap":  {<field name> : <error message>}} E.g.:  “lastTime": "Invalid timestamp format. Format must be yyyy-MM-dd HH:mm:ss and must be a valid date” |
| The name of the EGM has not been supplied.  {"code": 40002,  "message”: “Validation failed for verify order message timestamp”,  "errorMap":  {“egmId”: “The name of the EGM has not been supplied"}} |
| Date/time of the last player service order has not been supplied.  {"code": 40003,  "message”: “Validation failed for verify order message timestamp”,  "errorMap":  {“lastTime": “Date/time of the last player service order has not been supplied"}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The verifyPosOrderTimestamp request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The verifyPosOrderTimestamp request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The supplied property values could not be matched or validated in the S7000 system.  {"code": 40400,  "message": “The timestamp 'XXXX-XX-XX XX:XX:XX' cannot be  validated for EGM 'YYY’"} |
| {"code": 40401,  "message": “The external system might not have received the last request processed by S7000"} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# GET SQL RESULT

Request sent from the client for executing and fetching of results of a SQL query submitted to the S7000.

Please note that the number of data rows sent back in the response may be restricted by the S7000 system.

**Note:** The restriction to the number of rows returned, can be increased from 500 to 2000.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/export/getSQLResult |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| sqlString | String | Y | The SQL query string that the client intends to be executed on the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |
| resultSet | String | SQl result set as fetched from the S7000 system. |

**Sample Request**

{ "sqlString" : "select membernumber, firstname || ' ' || surname, membershipstatus from V\_PUB\_MEMBERS where membernumber < 10" }

**Sample Response**



© Aristocrat Technologies Australia Pty Limited 2023. *All rights reserved.*

COMMERCIAL-IN-CONFIDENCE

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "resultSet":

[ ["MEMBERNUMBER","FIRSTNAME||''||SURNAME","MEMBERSHIPSTATUS"], [" 100626","FNAME2 SNAME2","PENDING"],

[" 100627","FNAME4 SNAME4","PENDING"],

[" 100628","FNAME5 SNAME5","PENDING"], [" 100629","TESTAGAIN TEST","PENDING"], ["1","COOKER SALLLY","FINANCIAL"] ]}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| The sqlString property name of the request is incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “sqlsting": "Invalid Value" |
| The sqlString request property value i.e. the sql could not be validated successfully. Please change the sql and retry.  {"code": 40000,  "message”: “Validation failed for SQL string", "errorMap":  {<field name> : <error message>}} E.g.:  "sqlString": "Cannot fetch more than X rows of data in one query."  "message": "Invalid SQL statement." "message": "Invalid identifier." |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 402  Payment Required | The getSqlResult request was successful; however, the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The getSqlResult request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden for the current registration  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| Web method request forbidden for the current registration  {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The SQL execution resulted in no rows to be returned.  {"code": 40400,  "message": No data was found for this SQL string”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

REST API 3rd Party Interface Specification Version 1.29

# GET SQL RESULT AS A FILE

Request sent from the client for executing and fetching of results of a SQL query submitted to the S7000.

This request creates a delimited test output file that contains the SQL result set.

Please note that the number of data rows sent back in the response may be restricted by the S7000 system.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/export/getSQLResultAsFile |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| filename | String | Y | The name of the output file to be created with the SQL result set. |
| sqlString | String | Y | The SQL query string that the client intends to be executed on the S7000 system. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Request**

{ "fileName" : "sqlResultFile.txt",

"sqlString" : "select membernumber, firstname || ' ' || surname, membershipstatus from V\_PUB\_MEMBERS where membernumber < 10" }

**Sample Response**



© Aristocrat Technologies Australia Pty Limited 2023. *All rights reserved.*

COMMERCIAL-IN-CONFIDENCE

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success", "resultSet":

[["MEMBERNUMBER","FIRSTNAME||''||SURNAME","MEMBERSHIPSTATUS"], [" 100626","FNAME2 SNAME2","PENDING"],

[" 100627","FNAME4 SNAME4","PENDING"],

[" 100628","FNAME5 SNAME5","PENDING"], [" 100629","TESTAGAIN TEST","PENDING"], ["1","COOKER SALLLY","FINANCIAL"] ]}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0,  "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}}  E.g.: “filname": "Invalid Value" |
| The filename or the sqlString request property value could not be validated successfully. Please change the values and retry.  {"code": 40000,  "message”: “Validation failed for SQL string", "errorMap":  {<field name> : <error message>}} |
| 401  Unauthorized | HTTP Header is missing Authorization  {"code": 40100,  "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The getSqlResultAsFile request was successful; however, the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
|  | "message": "Success"} |
| The getSqlResultAsFile request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | The SQL execution resulted in no rows to be returned.  {"code": 40400,  "message": No data was found for this SQL string”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# EXPORT MEMBER FACIAL IMAGES

Request sent from the client for S7000 system to export out a range of facial images of the members.

This request results in a mapping text and the member image jpeg files.

The first line of the mapping text file is the header line that will contain the label text “MemberId, ImageFileName”. And every line after is the data i.e. member Id, followed by the image file name, separated by a comma.

##### Request

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Bearer<Session-token> |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **POST** | api/export/exportMemberImages |

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Type** | **Reqd.** | **Description** |
| firstMemberId | String (12) | Y | The first member in the range of members that you wish to retrieve an image for. MemberId should be an existing member membership number, cannot be NULL or 0. |
| lastMemberId | String (12) | Y | The last member in the range of members that you wish to retrieve an image for. MemberId should be an existing member membership number, cannot be NULL or 0. |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| licenceStatus | String | Contains “Expired” if the client registration has expired. Null if the client registration is still valid. |
| licenceExpiryDate | String | Contains the registration expiry date (format yyyy-mm- dd) if the client registration has expired.  Null if the client registration is still valid. |
| code | Number | Refer to HTTP Response Codes and Messages |
| message | String | Refer to HTTP Response Codes and Messages |

**Sample Mapping and Image Files** (generated by S7000 as a result of this request.)



**Sample Request**

{ "firstMemberId" : "1", "lastMemberId" : "20" }

**Sample Response**

{ "licenceStatus": null, "licenceExpiryDate": null, "code": 0,

"message": "Success" }

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message.  "code": 0, "message": "Success" |
| 400  Bad Request | The HTTP Header may be in an incorrect format.  {"code": 40000,  "message": "Invalid input data"} |
| One or more property names of the request are incorrectly supplied.  {"code": 40000,  "message”: "Incorrectly formatted JSON packet", "errorMap":  {<field name> : <error message>}} E.g.: “firstMemId": "Invalid Value" |
| Some of the request property values could not be validated successfully. Please change the values and retry.  {"code": 40000,  "message”: “Validation failed for member images", "errorMap":  {<field name> : <error message>}} E.g.:  "firstMemberId": "firstMemberId must be numeric" "firstMemberId": "firstMemberId cannot be blank." "lastMemberId": "lastMemberId cannot be blank." "lastMemberId": "lastMemberId must be numeric" |
| 401 | HTTP Header is missing Authorization  {"code": 40100, |

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| Unauthorized | "message": "Authentication not found in header"} |
| HTTP Header is missing the Session token  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| The Session token has expired.  {"code": 40100,  "message": “The session ‘X’ has expired”} |
| The Session token supplied with the request is not valid  {"code": 40100,  "message": "Session token ‘X’ was not found"} |
| 402  Payment Required | The exportMemberImage request was successful; however the client registration has expired.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  code": 0,  "message": "Success"} |
| The exportMemberImage request was NOT successful as the client registration has expired and needs to be renewed.  {"licenceStatus": "Expired”, "licenceExpiryDate": "2014-12-31",  "code": 40200,  "message": "The licence for registration code 'X' has expired."} |
| 403  Forbidden | Web method request forbidden  {"code": 40300,  "message": "The registration code 'X' is not valid for this web method"} |
| {"code": 40300,  "message": "The table or view is not accessible"} |
| 404  Not Found | None of the members within the requested member range have facial images captured within the S7000 system.  {"code": 40400,  "message": No record found for member Ids xxxxxxxxxx - ” yyyyyyyyyy”} |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# GET API VERSION

Request from the client to read WebAPI version and System 7000 release version.

**Request**

|  |  |
| --- | --- |
| **Header** | |
| **AUTHORIZATION** | Session-token is not required. |

|  |  |
| --- | --- |
| **Method** | **URL** |
| **GET** | api/session/getAPIVersion |

**Response**

|  |  |  |
| --- | --- | --- |
| **Property** | **Type** | **Description** |
| systemVersion | String | Contains System 7000 release version. |
| apiVersion | String | Contains 3rd Party Interface release version also known as WebAPI. |

**Sample Response**

{

"code": 0, "message": "Success", "versionResponse": {

"systemVersion": "S7000 7.4.8.10 Single-Site", "apiVersion": "WebAPI 3.0.48"

}

}

**HTTP Response Codes and Messages**

|  |  |
| --- | --- |
| **HTTP Code Status** | **Description** |
| 200  OK | The web method call was successful, returns the below code and message along with the additional property values as specified in the Response section.  "code": 0, "message": "Success" |
| 500  Internal Server Error | An unexpected condition that prevented the processing of the request from the client. Please contact Aristocrat Support.  {"code": 50000,  "message": "Internal server error"} |

# APPENDIX : S7000 DATABASE VIEWS

All of the database views described in this section can be accessed via the getSqlResult and getSqlResultAsFile requests.

Please note that for format consistency, some of the DATE fields may return the time component as 00:00:00 for the fields that have date only without the time component recorded.

Below is a description of all the S7000 database views:

## V\_PUB\_MEMBERS

The database view V\_PUB\_MEMBERS may be used to retrieve member details of an existing member from the S7000 system.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| MEMBERNUMBER | VARCHAR2(12) | Unique Identifier for the member, the membership number. All information regarding the member is associated with this Member Number. |
| SURNAME | VARCHAR2(25) | Surname of member. |
| FIRSTNAME | VARCHAR2(25) | First name of member. |
| ASSOCIATEDNAME | VARCHAR2(25) | Linked name of member |
| TITLE | VARCHAR2(25) | Title of member, E.g. Mr., etc. |
| ADDRESSLINE1 | VARCHAR2(60) | Street address of member (line 1). |
| ADDRESSLINE2 | VARCHAR2(60) | Street address of member (line 2). |
| ADDRESSLINE3 | VARCHAR2(60) | Street address of member (line 3). |
| ADDRESSLINE4 | VARCHAR2(60) | Street address of member (line 4). |
| ADDRESSPOSTCODE | VARCHAR2(12) | Postal code for member. |
| PRIMARYPHONE | VARCHAR2(20) | Home or primary telephone number. |
| DATEREGISTERED | DATE | Date when member first joined or registered. |
| DATEOFBIRTH | DATE | Date of birth of member. |
| NEWMEMBERFLAG | NUMBER(1) | Indicates the member is a new member. 1 = member is new  0 = not a new member |
| SUSPENSIONFLAG | NUMBER(1) | Indicates the current suspension status of the member.  0 = member is not suspended  1 = member is suspended. |
| GENDER | VARCHAR2(6) | ‘MALE’ or ‘FEMALE’ indicating member’s gender. |
| MEMBERSHIPSTATUS | VARCHAR2(16) | Indicates the current membership status of the |

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
|  |  | member. Returns one of the following:   * PENDING * FINANCIAL * UNFINANCIAL * RETIRED/DECEASED |
| BADGENUMBER | NUMBER(16) | Membership Badge Number assigned to the member. This may be blank if the venue does not use badge numbers for members. |
| LASTUPDATEDUSER | VARCHAR2(16) | Indicates the name of the System7000 user who has last updated the member details. |
| LASTUPDATEDTIME | DATE | Timestamp when the above user has last updated any of the member details as seen in the following views:   * v\_pub\_members * v\_pub\_cards * v\_pub\_member\_analysis * v\_pub\_mem\_otherdetails |

## V\_PUB\_CARDS

This database view V\_PUB\_CARDS may be used to retrieve details of any or all the cards which have been issued to a member.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| CARDNUMBER | VARCHAR2(20) | Unique Identifier of the member, the card number. All information regarding the card is associated with this Card Number. |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the card beNumbers to |
| MEMCARDTYPE | VARCHAR2(20) | Indicates the type of the card E.g. Membership Card, Gym Card. This card type is pre- configured by venues, hence may be different from venue to venue. |
| CARDISSUEDATE | DATE | Date on which the card was issued to the member. |
| CARDEXPIRYDATE | DATE | Date on which the card expires |
| CARDLOST | NUMBER(1) | Indicates if the card has been currently marked as lost.  1 = card is lost  0 = card is not lost |

## V\_PUB\_MEMBER\_ANALYSIS

This database view V\_PUB\_MEMBER\_ANALYSIS may be used to retrieve the member analysis i.e. member attribute details of a member, E.g.: Current Player Tiering, Language, Country of Birth etc.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the analysis details belong to. |
| ANALYSISNUMBER | NUMBER(10) | Unique Identifier of the Analysis Attribute Type |
| ANALYSISTYPECODE | VARCHAR2(20) | The code that needs to be used when calling the **addUpdateMember** or **updateMemberDetails** api calls |
| ANALYSISNAME | VARCHAR2(100) | Name of the Analysis Attribute type. E.g.: Tiering |
| ANALYSISVALUECODE | VARCHAR2(20) | Unique Identifier of the Analysis Attribute selected for the member.  This column value will be empty for Sub Clubs |
| ANALYSISVALUEFORMEMBER | VARCHAR2(20) | The value of the Analysis Attribute selected for the member. E.g.: GOLD MEMBER |
| ISSUBCLUB | VARCHAR2(1) | Indicate if Analysis is a Sub Club. ‘Y’ for Yes and ‘N’ for No |
| SUBCLUBEXPIRY | VARCHAR2(1000) | Sub Club Expiry |
| SUBCLUBDATEJOINED | VARCHAR2(1000) | Sub Club Date Joined |
| SUBCLUBCARDNUMBER | VARCHAR2(1000) | Sub Club Card Number |

Please note the view may return multiple records associated with an analysis name for any member if multiple member attribute values were assigned to the member. Below is an example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MEMBERNUMBER** | **ANALYSISN UMBER** | **ANALYSIS TYPECODE** | **ANALYSISNAME** | **ANALYSIS VALUECODE** | **ANALYSISVALUE FORMEMBER** |
| 1234567890 | 1 | 05 | LANGUAGE | 100 | ENGLISH |
| 1234567890 | 1 | 05 | LANGUAGE | 101 | FRENCH |
| 1234567890 | 2 | 05 | TIERING | 20 | GOLD |
| 2345678901 | 2 | 05 | TIERING | 30 | DIAMOND |
| 3456789012 | 3 | 10 | SPORT | 50 | CRICKET |
| 3456789012 | 3 | 10 | SPORT | 60 | SOCCER |
| 3456789012 | 3 | 10 | SPORT | 70 | RUGBY |
| 3456789012 | 2 | 05 | TIERING | 10 | SILVER |
| 3456789012 | 11 | A11 | DINER CLUB |  |  |

+ Additional columns for Sub Clubs

|  |  |  |  |
| --- | --- | --- | --- |
| **ISSUBCLUB** | **SUBCLUBEXPIRY** | **SUBCLUBDATEJOIN ED** | **SUBCLUBCARD NUMBER** |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 0 |  |  |  |
| 1 | 1-Apr-2021 | DINER CLUB | 1234567890 |

## V\_PUB\_MEM\_OTHERDETAILS

This database view V\_PUB\_MEM\_OTHERDETAILS may be used to retrieve the extended information recorded for any member analysis i.e. other telephone numbers, member occupation, email etc.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the analysis details belong to. |
| TELEPHONE1 | VARCHAR2(20) | Additional Telephone number1 recorded if any for this member. |
| TELEPHONE2 | VARCHAR2(20) | Additional Telephone number2 if any recorded for this member. |
| TELEPHONE3 | VARCHAR2(20) | Additional Telephone number3 if any recorded for this member. |
| FORMALTITLE1 | VARCHAR2(35) | Additional Formal Title1 if any recorded for this member. |
| FORMALTITLE2 | VARCHAR2(35) | Additional Formal Title2 if any recorded for this member. |
| PROPOSER1 | VARCHAR2(50) | Proposer1 information recorded for this member. |
| PROPOSER2 | VARCHAR2(50) | Proposer2 information recorded for this member. |
| OCCUPATION | VARCHAR2(50) | Occupation details recorded for this member. |
| EMAIL | VARCHAR2(64) | Email Id recorded for this member. |
| WEBURL | VARCHAR2(64) | Any web URL address recorded for this member. |

Please note, this database view may return multiple records for a member if there exists multiple records in S7000 database.

## V\_PUB\_MEMBER\_SECADDRESS

This database view V\_PUB\_MEMBER\_SECADDRESS may be used to retrieve the extended information recorded for any member analysis i.e. other telephone numbers, member occupation, email etc.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the analysis details belong to. |
| ADDRESSNUMBER | NUMBER(5) | If multiple addresses were recorded for this member, this number represents the address number. |
| ADDRESSLINE1 | VARCHAR2(60) | Street address of member (line 1). |
| ADDRESSLINE2 | VARCHAR2(60) | Street address of member (line 2). |
| ADDRESSLINE3 | VARCHAR2(60) | Street address of member (line 3). |
| ADDRESSLINE4 | VARCHAR2(60) | Street address of member (line 4). |
| ADDRESSPOSTCODE | VARCHAR2(15) | Postal code for member. |
| ADDRESSTELEPHONE | VARCHAR2(20) | Telephone number if any recorded against this address record for the member. |
| ADDRESSCOMMENTS | VARCHAR2(25) | Any comments recorded by the user when the address details were captured. |
| ISPOSTALADDRESS | NUMBER(1) | Indicates if this address is marked as postal address:  1 = is postal address  0 = is not the postal address |
| ADDRESSBARCODE | VARCHAR2(37) | Barcode number recorded for this address. |

Please note, this database view may return multiple records for a member if there exists multiple address records in S7000 database.

## V\_PUB\_MEM\_POINT\_BALANCE

This database view V\_PUB\_MEM\_POINT\_BALANCE may be used to retrieve the current reward point balances for any member for each reward point. The view will return a separate row for each member for each reward point type.

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the reward point balance belongs to. |
| REWARDTYPE | VARCHAR2(6) | Type of the reward point associated with the below reward point Identifier |
| REWARDID | NUMBER(20) | The unique Identifier of the type of the reward point |

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| CURRENTBALANCE | NUMBER(24,4) | Current Reward point Balance of the reward point for the member |
| REDEMPTIONORDER | VARCHAR2(6) | The Redemption order as configured within the S7000 system. Refer to REDEEM MEMBER POINTS section for more detail. |

Please note the view may return multiple records for any member if multiple reward points are in operation in a venue. Therefore, the current reward balance for any member read from this view is relevant only if it is retrieved using the first four fields as key Identifiers. Below is an example of the information available from this view:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMBERNUMBER** | **REWARDTYPE** | **REWARDID** | **CURRENTBALANCE** | **REDEMPTIONORDER** |
| 1234567890 | XREWARD | 1 | 12345.65 | 3 |
| 1234567890 | YREWARD | 2 | 23.75 |  |
| 1234567890 | ZREWARD | 3 | 7654.90 | 1 |
| 2345678901 | XREWARD | 1 | 800 |  |
| 3456789012 | XREWARD | 1 | 0 |  |
| 45678901234 | YREWARD | 2 | 600.50 | 2 |

## V\_PUB\_MEM\_PLAY\_SESSION

This database view V\_PUB\_MEM\_PLAY\_SESSION may be used to retrieve the EGM play activity of any registered S7000 member.

Note: Please provide SessionEndDateTime in the SQL condition where possible to speed up the response i.e. SessionEndDateTime >= TO\_DATE('2024-03-16 11:31:39','YYYY-MM-DD HH24:MI:SS')

The view will return

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| PLAYERSESSIONPK | VARCHAR2 (40) | Unique identifier of the EGM Play session |
| TRADEDATE | DATE | Trade date when the play session happened |
| SESSIONSTARTDATETIME | DATE | Calender date and time as recorded when the play session started |
| SESSIONENDDATETIME | DATE | Calender date and time as recorded when the play session ended |
| SESSIONDURATIONINSEC | NUMBER | Duration of the play session in seconds i.e. the difference between SESSIONSTARTDATETIME and the SESSIONENDDATETIME |
| MEMBERNUMBER | VARCHAR2(12) | Member Number of the member to whom the play session is associated with. |

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| CARDNUMBER | VARCHAR2(20) | The card that was used by the member for the play session. |
| PROMOTIONNAME | VARCHAR2(255) | Name of the venue promotion active at the time of the play session as a result of which the member was awarded reward points. |
| REWARDTYPE | VARCHAR2(6) | Short name of the reward points that were awarded to the member based on the promotion active at the time. |
| POINTSAWARDED | NUMBER(24,4) | Number of the points awarded for the play session of the above reward type. |
| POINTVALUEAWARDED | NUMBER(24,4) | Value of the points awarded for the play session of the above reward type. |
| EGMNAME | VARCHAR2(64) | Name of the EGM where the play session occurred. |
| TURNOVER | NUMBER(24,4) | Turnover as recorded for the play session (in cents) |
| CREDITWINS | NUMBER(24,4) | Credit Wins as recorded for the play session (in cents) |
| JACKPOT | NUMBER(24,4) | Jackpot Wins as recorded for the play session (in cents) |
| STROKE | NUMBER(24,4) | Number of strokes recorded for the play session |
| GAMENAME | VARCHAR2(50) | Name of the Game played by the member during the play session. |

## V\_PUB\_OCCUPATION\_LIST

This database view V\_PUB\_OCCUPATION\_LIST may be used to retrieve the master occupation list along with the occupation code and their active status. The view will return

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| OCCUPATION\_CODE | NUMBER | Unique identifier of the occupation |
| OCCUPATION\_DESC | VARCHAR2(50) | Occupation description |
| OCCUPATION\_ISACTIVE | NUMBER | Indicates if this occupation is marked as active or not:  1 = active  0 = inactive |

[END OF DOCUMENT]

# CAVEAT

For /membership/addUpdateMember and /membership/updateMemberDetails APIs, in case any of the address field is locked, send either do not send those fields in API request send null as a value for those fields.

In /membership/addUpdateMember and /membership/updateMemberDetails APIs, cardType, membershipType, cardLayout refer to the same thing. So if the API will pass all 3 fields in request, make sure they all have same value.

If any of them is/are passed in API request, the API will use that value and update the same after validating it

If more than one field is passed in API request, API will first compare those values to validate and after successful validation, API will update the value in database.

REST API 3rd Party Interface Specification Version 1.29

# TECHNICAL SUPPORT

For technical assistance, please contact:

###### Aristocrat Systems Support

Tel: 1800 150 432

Email: [systemsservice@ali.com.au](mailto:systemsservice@ali.com.au) Web: [www.aristocratgaming.com.au](http://www.aristocratgaming.com.au/)

© Aristocrat Technologies Australia Pty Limited 2023. All rights reserved.

ABN 22 001 660 715, ACN 001 660 715