**IDMS ICD and Design Identification Management System**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Author** | **Description** |
| 1.0 | 5/1/2012 | Robert Lantry | Initial Version |
| 1.4 | 9/18/2012 | Scott Salley | Updated configuration, removed meta api, added CreateXBand, UpdateXBandType, createGuest, updateGuest |
| 1.5 | 01/15/2013 | Ted Crane | Clean up |

**Document Approvers & Sign-Off**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Approver** | **Role** | **Document Accept/Reject** |
| 5/1/2012 | Ric Merrifield | Release Manager | Accept |
| 5/11/2012 | John Stiehl | Release Manager | Accept |

**Table of Contents**

[1 Document Purpose and IDMS Background 8](#_Toc325103546)

[2 Technology Used 10](#_Toc325103547)

[3 Installation 10](#_Toc325103548)

[4 Dependencies 10](#_Toc325103549)

[5 Actors and Terms 10](#_Toc325103550)

[6 External Libraries in Build Path 12](#_Toc325103551)

[7 Architecture Overview 13](#_Toc325103552)

[8 Configuration 14](#_Toc325103553)

[9 IDMS Application API 15](#_Toc325103554)

[10 Celebrations 15](#_Toc325103555)

[10.1 getCelebrationByGuestXId 17](#_Toc325103556)

[10.1.1 Request 17](#_Toc325103557)

[10.1.2 Example 17](#_Toc325103558)

[10.1.3 Response Headers 17](#_Toc325103559)

[10.1.1 Response Body 17](#_Toc325103560)

[10.2 getCelebrationsByGuestId 19](#_Toc325103561)

[10.2.1 Request 19](#_Toc325103562)

[10.2.2 Example 19](#_Toc325103563)

[10.2.3 Response Headers 19](#_Toc325103564)

[10.2.4 Response Body 19](#_Toc325103565)

[10.3 getCelebrationById 21](#_Toc325103566)

[10.3.1 Request 21](#_Toc325103567)

[10.3.2 Example 21](#_Toc325103568)

[10.3.3 Response Headers 21](#_Toc325103569)

[10.3.4 Response Body 21](#_Toc325103570)

[11 IDMS Meta View 22](#_Toc325103571)

[11.1 GetIDMSIdFromName 24](#_Toc325103572)

[11.1.1 Request 24](#_Toc325103573)

[11.1.2 Example 24](#_Toc325103574)

[11.1.3 Response Headers 24](#_Toc325103575)

[11.1.4 Response Body 24](#_Toc325103576)

[11.2 GetIDMSTypeKeys 24](#_Toc325103577)

[11.2.1 Request 25](#_Toc325103578)

[11.2.2 Example 25](#_Toc325103579)

[11.2.3 Response Headers 25](#_Toc325103580)

[11.2.4 Response Body 25](#_Toc325103581)

[11.2.1 Usage Note 25](#_Toc325103582)

[11.3 getIDMSTypeById 25](#_Toc325103583)

[11.3.1 Request 26](#_Toc325103584)

[11.3.2 Example 26](#_Toc325103585)

[11.3.3 Response Headers 26](#_Toc325103586)

[11.3.4 Response Body 26](#_Toc325103587)

[11.4 CreateIDMSType 26](#_Toc325103588)

[11.4.1 Request 26](#_Toc325103589)

[11.4.2 Example 27](#_Toc325103590)

[11.4.3 Response Headers 27](#_Toc325103591)

[11.4.4 Response Body 27](#_Toc325103592)

[11.1 OneViewGuestService 27](#_Toc325103593)

[11.2 getGuestXBandsByIDType 31](#_Toc325103594)

[11.2.1 Request 31](#_Toc325103595)

[11.2.2 Example 31](#_Toc325103596)

[11.2.3 Response Headers 31](#_Toc325103597)

[11.2.4 Response Body 31](#_Toc325103598)

[11.3 getGuestIdentifiers 32](#_Toc325103599)

[11.3.1 Request 33](#_Toc325103600)

[11.3.2 Example 33](#_Toc325103601)

[11.3.3 Response Headers 33](#_Toc325103602)

[11.3.4 Response Body 33](#_Toc325103603)

[11.4 getGuestCelebrations 34](#_Toc325103604)

[11.4.1 Request 34](#_Toc325103605)

[11.4.2 Example 34](#_Toc325103606)

[11.4.3 Response Headers 35](#_Toc325103607)

[11.4.4 Response Body 35](#_Toc325103608)

[11.5 putGuestIdentifier 36](#_Toc325103609)

[11.5.1 Request 36](#_Toc325103610)

[11.5.2 Example 36](#_Toc325103611)

[11.5.3 Response Headers 37](#_Toc325103612)

[11.5.4 Response Body 37](#_Toc325103613)

[11.6 postGuestIdentifier 37](#_Toc325103614)

[11.6.1 Request 37](#_Toc325103615)

[11.6.2 Example 37](#_Toc325103616)

[11.6.3 Response Headers 38](#_Toc325103617)

[11.6.4 Response Body 38](#_Toc325103618)

[11.7 getGuestProfileById 38](#_Toc325103619)

[11.8 getGuestOnlineProfile 38](#_Toc325103620)

[11.9 getGuestProfile 38](#_Toc325103621)

[11.9.1 Request 38](#_Toc325103622)

[11.9.2 Example 39](#_Toc325103623)

[11.9.3 Response Headers 39](#_Toc325103624)

[11.9.4 Response Body 39](#_Toc325103625)

[11.10 getOneViewParty 42](#_Toc325103626)

[11.10.1 Request 42](#_Toc325103627)

[11.10.2 Example 42](#_Toc325103628)

[11.10.3 Response Headers 42](#_Toc325103629)

[11.10.4 Response Body 42](#_Toc325103630)

[11.11 Area 43](#_Toc325103631)

[11.11.1 Request 43](#_Toc325103632)

[11.11.2 Example 43](#_Toc325103633)

[11.11.3 Response Headers 44](#_Toc325103634)

[11.11.4 Response Body 44](#_Toc325103635)

[11.1 PartyService 44](#_Toc325103636)

[11.2 findPartyByPartyName 46](#_Toc325103637)

[11.2.1 Request 46](#_Toc325103638)

[11.2.2 Example 46](#_Toc325103639)

[11.2.3 Response Headers 46](#_Toc325103640)

[11.2.1 Response Body 46](#_Toc325103641)

[11.3 GetVisitByVisitId 47](#_Toc325103642)

[11.3.1 Request 47](#_Toc325103643)

[11.3.2 Example 48](#_Toc325103644)

[11.3.3 Response Headers 48](#_Toc325103645)

[11.3.4 Response Body 48](#_Toc325103646)

[11.4 GetGuestsByVisitId 48](#_Toc325103647)

[11.4.1 Request 49](#_Toc325103648)

[11.4.2 Example 49](#_Toc325103649)

[11.4.3 Response Headers 49](#_Toc325103650)

[11.4.4 Response Body 49](#_Toc325103651)

[11.5 createParty 50](#_Toc325103652)

[11.5.1 Request 50](#_Toc325103653)

[11.5.2 Example 50](#_Toc325103654)

[11.5.3 Response Headers 51](#_Toc325103655)

[11.5.4 Response Body 51](#_Toc325103656)

[11.6 AddGuestToParty 51](#_Toc325103657)

[11.6.1 Request 51](#_Toc325103658)

[11.6.2 Example 52](#_Toc325103659)

[11.6.3 Response Headers 52](#_Toc325103660)

[11.6.4 Response Body 52](#_Toc325103661)

[11.7 Configuration 52](#_Toc325103662)

[11.8 getConfiguration 53](#_Toc325103663)

[11.8.1 Request 53](#_Toc325103664)

[11.8.2 Example 53](#_Toc325103665)

[11.8.3 Response Headers 53](#_Toc325103666)

[11.8.4 Response Body 53](#_Toc325103667)

[11.9 saveConfiguration 54](#_Toc325103668)

[11.9.1 Request 54](#_Toc325103669)

[11.9.2 Example 55](#_Toc325103670)

[11.9.3 Response Headers 55](#_Toc325103671)

[11.9.4 Response Body 55](#_Toc325103672)

[11.10 OneViewXBandView 55](#_Toc325103673)

[11.11 AssignXBandToGuest 57](#_Toc325103674)

[11.11.1 Request 58](#_Toc325103675)

[11.11.2 Example 58](#_Toc325103676)

[11.11.3 Response Headers 58](#_Toc325103677)

[11.11.4 Response Body 58](#_Toc325103678)

[11.12 RemoveXBandFromGuest 58](#_Toc325103679)

[11.12.1 Request 59](#_Toc325103680)

[11.12.2 Example 59](#_Toc325103681)

[11.12.3 Response Headers 59](#_Toc325103682)

[11.12.4 Response Body 59](#_Toc325103683)

[11.13 AssignXBand 59](#_Toc325103684)

[11.13.1 Request 60](#_Toc325103685)

[11.13.2 Example 60](#_Toc325103686)

[11.13.3 Response Headers 60](#_Toc325103687)

[11.13.4 Response Body 60](#_Toc325103688)

[11.14 GetXBandByxBandId 61](#_Toc325103689)

[11.15 GetXBandByxBandIdTwo 61](#_Toc325103690)

[11.16 GetXBandByBandId 61](#_Toc325103691)

[11.17 GetXBandBylrid 61](#_Toc325103692)

[11.18 GetXBandBytapid 61](#_Toc325103693)

[11.19 GetXBandBysecureid 61](#_Toc325103694)

[11.20 GetXBandByUid 61](#_Toc325103695)

[11.20.1 Request 61](#_Toc325103696)

[11.20.2 Examples 62](#_Toc325103697)

[11.20.3 Response Headers 62](#_Toc325103698)

[11.20.4 Response Body 62](#_Toc325103699)

[11.21 XBandView 63](#_Toc325103700)

[11.22 getXBandByXBandIdOld 65](#_Toc325103701)

[11.23 getXBandByXBandId 65](#_Toc325103702)

[11.24 getXBandByTapId 65](#_Toc325103703)

[11.25 getXBandBylrId 65](#_Toc325103704)

[11.26 getXBandByBandId 65](#_Toc325103705)

[11.27 getXBandBySecureId 65](#_Toc325103706)

[11.27.1 Request 65](#_Toc325103707)

[11.27.2 Examples 66](#_Toc325103708)

[11.27.3 Response Headers 66](#_Toc325103709)

[11.27.4 Response Body 66](#_Toc325103710)

[11.28 Importer 68](#_Toc325103711)

[11.29 uploadFile 68](#_Toc325103712)

[11.29.1 Request 69](#_Toc325103713)

[11.29.2 Example 69](#_Toc325103714)

[11.29.3 Response Headers 69](#_Toc325103715)

[11.29.4 Response Body 69](#_Toc325103716)

# Document Purpose and IDMS Background

This document helps to outline and define the functionality of the Identification Management System (IDMS), its use and dependencies, considerations during a deployment scenario, and what it provides client systems.

At its heart, IDMS is an ID to ID management system. It functions as a lookup or “phonebook” for various systems to look up data in other systems.

Though IDMS ultimately acts as an ID repository, the following needs have helped flesh out some of its functionality:

* The need to track the association between a guest and their redeemable entitlement.
* The need for guests to be associated to an entitlement instrument, either a Magic Band or an RFID card.

IDMS provides the means to associate the Magic Band or RFID card with a guest, as well as with all of the subsequent foreign system IDs that are registered in the system for that guest. Once this association is created, entitlements can be checked out of GxP and associated with the Magic Band or RFID card.

In the short term, IDMS also provides the means for a guest to be associated with a celebration type, such as an anniversary or a birthday. This information is used by other systems within the park either to display messages or to notify cast members so they can further personalize the guest’s park experience.

# Technology Used

IDMS was developed in Java using the Open JDK 1.6 for Ubuntu in an Eclipse Indigo development environment. IDMS operates as a servlet, with TC Server acting as the servlet container, providing a REST API over HTTPS.

IDMS uses MS SQL Server to provide database services through TSQL Stored Procedures and parameterized SQL statements with JDBC.

# Installation

Installation for the IDMS system is discussed in the document *NGE Alpha Lab Installation*.

# Dependencies

IDMS has two components, a middle tier library called IDMSLib.jar and the IDMS application. The IDMS application is a RESTful API over HTTP. The API uses standard HTTP, mapping its calls to **GET**, **PUT**, **POST** and **DELETE** calls and responds with HTTP headers, response types and response codes. This is accomplished using the Jersey JAX-RS implementation.

# Actors and Terms

IDMS has several components and language features that are intrinsic to it and that may not map directly to other external systems. It has undergone several name changes over its life cycle, and many of the actors have also changed. When discussing the system, it is important to understand that IDMS may have analogs that are not equivalent to external systems like GxP. For example, Magic Bands were originally called xBands, and that older terminology still exists in IDMS – and will for the foreseeable future, since maintaining backwards compatibility for IDMS clients has been extremely important. Philosophically, IDMS prefers to deprecate but maintain older calls when replacing actors, names, or concepts in the system.

In the context of this document, *actors* refer to the primary types of data that IDMS deals with and exposes through its API. Below is a high level matrix and description of IDMS actors:

| **Actor** | **IDMS API** | **Description** |
| --- | --- | --- |
| Celebration | IDMS/celebration | Magical Celebrations are a temporary API in IDMS as there is not a system of record for them in time for the GxP Pilot. Eventually this portion of the API will be deprecated and IDMS will provide pass-through to the Magical Celebration SOR. |
| Guest | IDMS/guests | This is an older API, designed when IDMS was originally XView. It should, at some point in the future, be deprecated. This API provides means to retrieve a guest by Id. |
| Guest | IDMS/guest | This is the current API that is compatible with GxP and other clients. It provides the ability to create guests, edit guests and provides a guest object with identifiers, assigned xbands, as well as some search capabilities. |
| Visit | IDMS/visit | Provides information about a guest’s visit and the number and members of their party. This has been deprecated and replaced with the Party API |
| Party | IDMS/party | Provides means to create, add, update guests and their party members. |
| xBand | IDMS/xbands | Deprecated. This is the original XView xband object with guest. |
| xBand | IDMS/xband | This API provides means to assign guests to xbands, unassign them, and retrieve xBand by their xband Id, tapId, long range Id, secureId, UId and bandId. This API also provides a bulk xband uploader. |

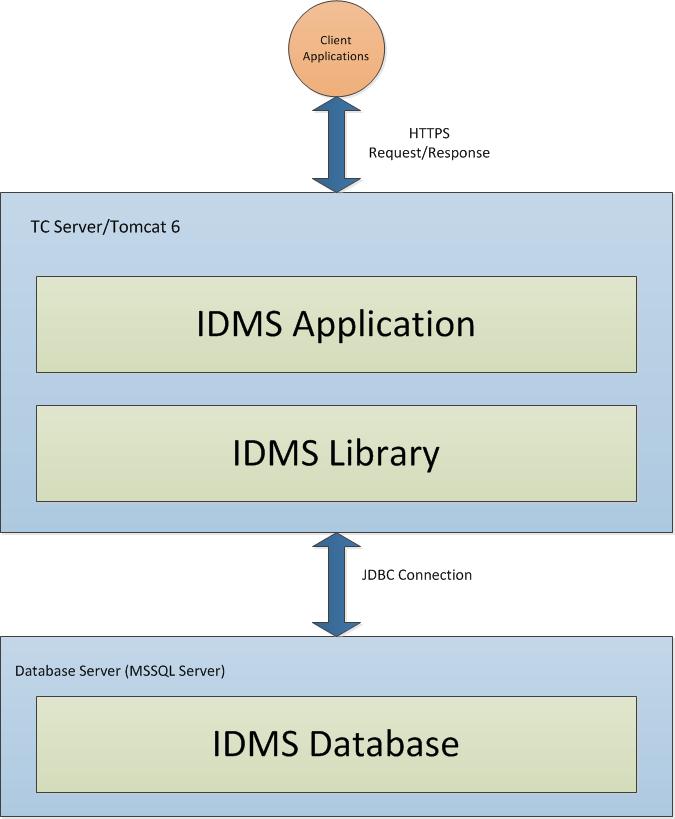
Each of these areas will be further described later in this document.

# External Libraries in Build Path

The following libraries provide an inventory of the technologies used in IDMS.

| **Library** | **Description** |
| --- | --- |
| IDMSLib.jar | Provides the middle-tier database connectivity as well as the inventory of business objects used and methods called to support the IDMS application. |
| asm.jar | A dependency library used by Jackson libraries. |
| c3p0.jar | A connection pooling library. |
| jackson-core-asl.jar | A component of Jackson that provides JSON portability. |
| jacsson-jaxrs.jar | Jaxrs provides mapping capabilities for java objects and XML. |
| jackson-mapper-asl.jar | Provides mapping services from java objects to other types of objects, such as JSON. |
| jackson-xc.jar | Provides XML services functionality for Jackson for handlers such as JAXB. |
| jersey-core.jar | Core functionality creating RESTful web services. Defines the attributes and objects used by other components in Jersey. |
| jersey-json.jar | Jersey expression object, responsible for creating JSON messages. |
| Jersey-server.jar | Provides container information for the tomcat server to act as an servelet container for a RESTful API. |
| jettison.jar | Provides RESTful implementation to HTTPS endpoints. |
| jsr311-api.jar | Java Specification Request is part of the Java EE extensions. |
| junit.jar | Provides automated unit testing features to Java. |
| log4j.jar | Logging and metrics monitoring system for Java. |
| commons-lang.jar | Provides common utilities for Java, such as string manipulation and escaping and unescaping values passed into a RESTful system. |
| jtds.jar | JTDS is an alternate database connection library for Java. |
| jcs.jar | Java Cache System – provides caching mechanism for Java. |
| commons-dbutils.jar | Provides a wrapper on JDBC database primitives, managing connection pooling and providing other services to JDBC code. |

# Architecture Overview



# Configuration

IDMS uses the file /etc/nge/config/environment.properties. Any change to environment.properties requires Tomcat or TC Server to be restarted.

The following is an example configuration:

nge.xconnect.idms.databasedriver=net.sourceforge.jtds.jdbc.Driver

nge.xconnect.idms.dbserver.url=jdbc:jtds:sqlserver://localhost:1433/IDMS

nge.xconnect.idms.dbserver.uid=EMUSER

nge.xconnect.idms.dbserver.pwd=Mayhem!23

nge.xconnect.idms.c3p0.maxPoolSize=20

nge.xconnect.idms.c3p0.maxStatementsPerConnection=50

nge.xconnect.idms.c3p0.maxStatements=400

nge.xconnect.idms.cache.enable=true

nge.xconnect.idms.cache.maxGuests=1000000

nge.xconnect.idms.cache.maxBands=3000000

nge.xconnect.idms.cache.resetTime=17:00

nge.xconnect.idms.ssl.keystore.client=/etc/nge/ssl/xconnect/.client.ks

nge.xconnect.idms.ssl.keystorePassword.client=ENC(sai+/7P4uf69LweE1nP9gae2YhPZwpGoZJ0vI9MTXOE=)

| **Property** | **Description** |
| --- | --- |
| nge.xconnect.idms.databasedriver | The name of the database driver used to create an instance of a database connection. |
| nge.xconnect.idms.dbserver.url | A JDBC database connection URL. |
| nge.xconnect.idms.dbserver.uid | A user created in the IDMS database that has read/write ability and the ability to run the stored procs associated with IDMS. |
| nge.xconnect.idms.dbserver.pwd | The password for the databaseUser. |
| nge.xconnect.idms.readOnly | Interfaces that cause a change in the database will return an error when set to true. Default: false. |
| nge.xconnect.idms.c3p0.maxPoolSize |  |
| nge.xconnect.idms.c3p0.maxStatementsPerConnection |  |
| nge.xconnect.idms.c3p0.maxStatements |  |
| nge.xconnect.idms.ssl.keystore.client | Location of the client SSL keystore installed. |
| nge.xconnect.idms.ssl.keystorePassword.client | The password for the SSL keystore. |
| nge.xconnect.idms.cache.enable | Set to ‘true’ to turn on caching. Default: false |
| nge.xconnect.idms.cache.maxGuests | The maximum number of guest objects to hold in memory. Default: 1000000 |
| nge.xconnect.idms.cache.maxBands | The maximum number of band objects to hold in memory. Default: 3000000 |
| nge.xconnect.idms.cache.resetTime | Time of day (HH:mm, 24-hr clock) to clear cache. Default: “” |

# IDMS Application API

This section describes the IDMS RESTful API and provides guidance and examples of the types of objects used in an IDMS transaction.

## Configuration

The configuration API calls allow configuration applications, such as the xBRMS to view and alter the IDMS configuration.

|  |
| --- |
| @SuppressWarnings("rawtypes")  @GET  @Produces({"application/json"})  **public** JResponse getConfiguration();  @SuppressWarnings("rawtypes")  @POST  @Consumes({"application/json"})  **public** JResponse saveConfiguration(String settings); |

### getConfiguration

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/configuration |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| None |  |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| **None** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/configuration |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Configuration values are returned. |

#### Response Body

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | {   * "databaseURL": "jdbc:jtds:sqlserver://localhost:1433/IDMS", * "emUserName": "EMUSER", * "emUserPassword": "Mayhem!23", * "remoteSettings": "", * "restartIDMS": false   } | | databaseURL – the jdbc URL to the location of the IDMS database. | | emUserName : the IDMS database user with databaseReader/databaseWriter privileges in IDMS. | | emUserPassword: The password for the emUserName. | | remoteSettings: The location of remote (not in the build path of the IDMS Web Application) configuration file. Settings would alternately be read from this location. The path must include the name of the properties file, i.e., /usr/share/settings/idms/idmslib.properties | | restartIDMS – Not used at this time. | |

### saveConfiguration

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/configuration |
| **HTTP Method** | POST |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| {  "databaseDriver": "net.sourceforge.jtds.jdbc.Driver",  "databaseURL": "jdbc:jtds:sqlserver://localhost:1433/IDMS",  "databaseUser": "EMUSER",  "maxPoolSize": "20",  "maxStatements": "400",  "maxStatementsPerConnection": "50",  "restartIDMS": false,  "cacheMaxGuests": "1000000",  "cacheMaxBands": "3000000",  "cacheResetTime": "17:00"  } |
|  |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/configuration |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | The configuration was successful applied. IDMS database connections are automatically restarted. |

#### Response Body

|  |
| --- |
| None |

## Status

|  |
| --- |
| @SuppressWarnings("rawtypes")  @GET  @Produces({"application/json"})  **public** JResponse getStatus(); |

### getStatus

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/status |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| None |  |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| **None** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/status |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Status values are returned. |

#### Response Body

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | <status>  <databaseURL>jdbc:jtds:sqlserver://localhost:1433/IDMS</databaseURL>  <databaseUserName>EMUSER</databaseUserName>  <databaseVersion>1.4.0.0001</databaseVersion>  <hostname>scottsa-ubuntu</hostname>  <startTime>2012-09-25T09:28:46.284-04:00</startTime>  <status>Green</status>  <readOnly>true</readOnly>  <statusMessage> Guest objects 0/1000000  Band objects 0/3000000  Cache is enabled</statusMessage>  <version>1.4.0.3007</version>  </status> | | databaseURL | The jdbc URL to the location of the IDMS database. | | databaseUserName | IDMS database user with connection and execute permissions. | | databaseVersion | Schema version as reported by database at databaseURL. | | Hostname | hostname of system where IDMS is running. | | Status | ‘Green’ if all is good, ‘Red’ otherwise. | | statusMessage | A message of use to people. Not intended to be parsable. | | readOnly | ‘true’ if IDMS is in read-only mode. | |

## OneViewGuestService

The OneViewGuestService provides the calls for IDMS to interact with Guests, providing services such as search, and updating guest identifiers.

|  |
| --- |
| @Path("/guest")  **public** **interface** IOneViewGuestService {  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse createGuest();  @PUT  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse updateGuest();  @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/xbands")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestXBandsByIDType(@PathParam("id") PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/identifiers")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestIdentifiers(@PathParam("id") PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @PUT  @Path("/{id}/identifiers")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse putGuestIdentifier(@PathParam("id") PathSegment pathSegment, String identifier);    @SuppressWarnings("rawtypes")  @POST  @Path("/{id}/identifiers")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse postGuestIdentifier(@PathParam("id") PathSegment pathSegment, String identifier);    @SuppressWarnings("rawtypes")  @GET  @Path("/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestProfileById(@PathParam("id")PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/online-profile")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestOnlineProfile(@PathParam("id")PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/profile")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestProfile(@PathParam("id")PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/visit/current")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getOneViewParty(@PathParam("id")PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @GET  @Path("/search/{searchName}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse searchByName(@PathParam("searchName")String searchName);    @SuppressWarnings("rawtypes")  @GET  @Path("/searchEmail/{email}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetGuestByEmail(@PathParam("email")String email);    @SuppressWarnings("rawtypes")  @GET  @Path("/name/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getGuestNameById(@PathParam("id")PathSegment pathSegment);    @SuppressWarnings("rawtypes")  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse SaveGuest(String guestString);    @SuppressWarnings("rawtypes")  @PUT  @Path("/email")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse UpdateGuestEmail(String guestEmailPutString);    @SuppressWarnings("rawtypes")  @PUT  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse updateGuest(String guestString);    } |
|  |

### createGuest

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest |
| **HTTP Method** | POST |
| **Response Format** | application/json |

#### Request

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

**Request Body**

|  |
| --- |
| **{**  **"swid": "1234ABCD-1234-ABCD-123456ABCDEF",**  **"name": {**  **"lastName": "Doe",**  **"firstName": "John",**  **"middleName": "Mo"**  **}**  **"visitCount": "2",**  **"status": "Active",**  **"gender": "MALE"**  **}** |
|  |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/guest/ |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Guest was created |

#### Response Body

|  |
| --- |
| Location: http://localhost:8080/IDMS/guest/12345 |

### updateGuest

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest |
| **HTTP Method** | PUT |
| **Response Format** | application/json |

#### Request

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

**Request Body**

|  |
| --- |
| **{**  **"guestId": "12345"**  **"visitCount": "4",**  **}** |
| **Many fields may be updated.** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/guest/ |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 204 | Guest was updated. |

#### Response Body

|  |
| --- |
| Location: http://localhost:8080/IDMS/guest/12345 |

### getGuestXBandsByIDType

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/xbands |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**Request Body**

#### ****Example****

|  |  |
| --- | --- |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/xbands |  |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | XBands found for guest in the IDMS system by the given guestId or guest Identifier. |
| 404 | No xbands found for the given guestId or guest Identifier. |
| 500 | Internal server error. |

#### Response Body

|  |
| --- |
| {  "startItem": 0,  "itemLimit": 0,  "itemCount": 1,  "xbands": [  {  "state": "Active",  "publicId": "1D49BBCDA12244",  "xbandId": 35001106,  "bandType": "Guest",  "guests": [],  "xbandRequestId": "4069065A7436F000",  "secureId": "0000000975363400",  "shortRangeTag": "F0BAE76556AC00",  "longRangeTag": "FF4B449400",  "secondaryState": "Active",  "productId": "4069065A7436F000",  "assignmentDateTime": 1324515916300,  "links": {  "self": {  "href": "/xband/35001106"  },  "ownerProfile": {  "href": "/guest/3425/profile"  }  },  "bandId": "4069065A7436F000"  }  ]  } |

### getGuestIdentifiers

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/identifiers |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**Request Body**

#### ****Example****

|  |
| --- |
| <http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/identifiers> |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Identifiers were found for the given guestId or guest Identifier. |
| 404 | No identifiers were found for the given guestId or guest Identifier. |

#### Response Body

|  |
| --- |
| {  "identifiers": [  {  "value": "5B1856DAD9BA44B5A3CA2F3615762F98",  "type": "xid",  "guestId": 35001106  },  {  "value": "503052",  "type": "gxp-link-id",  "guestId": 35001106  }  ],  "links": {  "self": {  "href": "/guest/35001106/identifiers"  },  "ownerProfile": {  "href": "/guest/35001106/profile"  }  }  } |

### putGuestIdentifier

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/identifiers |
| **HTTP Method** | PUT |
| **Response Format** | Http Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| Id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**Request Body**

|  |
| --- |
| **{**  **"identifiers": [**  **{**  **"value":"487147",**  **"type":"gxp‑link‑id",**  **"guestId":55181**  **},**  **{**  **"value":"22045DE9DE7BF63C0D2097629255271A",**  **"type":"xid",**  **"guestId":55181**  **}],**  **"links":**  **{**  **"self":**  **{**  **"href":"/guest/55181/identifiers"**  **},**  **"ownerProfile":**  **{**  **"href":"/guest/55181/profile"**  **}**  **}**  **}** |
|  |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/identifiers |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Created – the identifier was created for the given guest Id. |
| 400 | Bad request – the identifier was not created. The identifier type probably did not match the give IDMS types for SOURCESYSTEM in the IDMS\_Type table. |
| 404 | FIXME – Code can thow this but not sure how it would happen. |

#### Response Body

|  |
| --- |
| None |

### postGuestIdentifier

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/identifiers |
| **HTTP Method** | POST |
| **Response Format** | Http Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**Request Body**

|  |  |
| --- | --- |
| |  | | --- | | **{**  **"identifier-type":"gxp-link-id",**  **"identifier-value":"500151"**  **}** | |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/identifiers |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Created – the identifier was created for the given guest Id. |
| 400 | Bad request – the identifier was not created. The identifier type probably did not match the give IDMS types for SOURCESYSTEM in the IDMS\_Type table. |
| 404 | FIXME – Code can thow this but not sure how it would happen. |

#### Response Body

|  |
| --- |
| None |

### getGuestProfileById

### getGuestOnlineProfile

### getGuestProfile

This call in IDMS encapsulates several paths but returns the same guest object

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/  <idms-context>/IDMS/guest/{id}/profile  <idms-context>/IDMS/guest/{id}/guest-profile |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**None**

**Request Body**

**None**

#### ****Example****

|  |  |
| --- | --- |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/ |  |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/profile |  |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/online-profile |  |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Guest found and returned in application/json format. |
| 404 | Guest was not found by guestId or identifier. |

#### Response Body

|  |
| --- |
| {  "identifiers": [  {  "guestId": 1,  "value": "8DED4950C99B4AEC847D96F2641B6A1E",  "type": "xid"  },  {  "guestId": 1,  "value": "1000",  "type": "gxp-link-id"  }  ],  "guestId": 1,  "partyId": 0,  "visitCount": 2,  "emailAddress": "mike.smith134@disneyworld.com",  "gender": "MALE",  "countryCode": "BR",  "swid": "94DBCE3F-6B15-4EC6-8794-1B78AFA1FE77",  "experiences": null,  "xbands": [  {  "xbandId": 1,  "xbandRequestId": "xBand1",  "secureId": "845902699",  "shortRangeTag": "F070A6FC07E200",  "longRangeTag": "FF40FF3C00",  "secondaryState": "Active",  "productId": "xBand1",  "assignmentDateTime": 1331863138870,  "links": {  "self": {  "href": "/xband/1"  },  "ownerProfile": {  "href": "/guest/1/profile"  }  },  "bandId": "xBand1",  "state": "Active"  },  {  "xbandId": 1,  "xbandRequestId": "xBand1",  "secureId": "845902699",  "shortRangeTag": "F070A6FC07E200",  "longRangeTag": "FF40FF3C00",  "secondaryState": "Active",  "productId": "xBand1",  "assignmentDateTime": 1331863138870,  "links": {  "self": {  "href": "/xband/1"  },  "ownerProfile": {  "href": "/guest/1/profile"  }  },  "bandId": "xBand1",  "state": "Active"  }  ],  "name": {  "lastName": "Smith",  "firstName": "Mike"  },  "status": "Active"  } |

### getOneViewParty

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/guest/{id}/visit/current |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | An IDMS guestId or guest Identifier in matrix format. |

**Request Header**

**None**

**Request Body**

**None**

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/guest/id;xid=5B1856DAD9BA44B5A3CA2F3615762F98/visit/current |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Visits were found for this guest. |
| 404 | No visits were found for this guest Id. |

#### Response Body

|  |
| --- |
| {   * "count": 1, * "members": [   + {     - "guestId": 36001106,     - "links": {       * "ownerProfile": {         + "href": "/guest/36001106/profile"   }   * + - },     - "isPrimary": true   }   * ], * "partyId": 1, * "primaryGuestId": 36001106, * "partyName": "tomLantry@gmail.com" |

## OneViewXBandView

The OneViewXBandView provides xband services to retrieve xbands by various IDs and to assign or unassign xbands to guests.

|  |
| --- |
| @Path("/xband")  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse CreateXBand();  @SuppressWarnings("rawtypes")  @GET  @Path("/{xbandId}/{guestId}")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse AssignXBandToGuest(@PathParam("xbandId")**long** xbandId, @PathParam("guestId")**long** guestId );  @SuppressWarnings("rawtypes")  @GET  @Path("/unassign/{xbandId}/{guestId}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse RemoveXBandFromGuest(@PathParam("xbandId")**long** xbandId, @PathParam("guestId")**long** guestId );  @SuppressWarnings("rawtypes")  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse AssignXBand(String oneviewGuestXBandAssign) **throws** JsonParseException, JsonMappingException, IOException;  @SuppressWarnings("rawtypes")  @GET  @Path("/id/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandByxBandId(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandByxBandIdTwo(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/bandid/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandByBandId(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/lrid/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandBylrid(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/tapid/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandBytapid(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/secureid/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandBysecureid(@PathParam("id") String Id);  @SuppressWarnings("rawtypes")  @GET  @Path("/uid/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetXBandByUid(@PathParam("id") String Id |

### CreateXBand

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband |
| **HTTP Method** | POST |
| **Response Format** | application/json |

#### Request

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

**Request Body**

|  |
| --- |
| **{**  **"bandId":1021,**  **"longRangeTag": "1234ABCD”,**  **"shortRangeTag": "12345832323",**  **"secureId": "4723985AB3523C32557A34234B435345A",**  **"publicId": "12349876"**  **}** |
| **The short range tag is an unsigned integer created from by assuming least significant byte first from the hardware short range tag/rfid.** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/xband/ |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | XBand was created |

#### Response Body

|  |
| --- |
| Location: http://localhost:8080/IDMS/xband/38234 |

### UpdateXBandType

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband/updateType/{xbandId}/{IDMSBANDTYPE} |
| **HTTP Method** | GET |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| xbandId | The IDMS xbandId of the band to assign. |
| guestId | The IDMS type of XBAND (Guest, Puck) |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/xband/updateType/1021/Puck |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Ok. The band type was successfully changed. |

#### Response Body

|  |
| --- |
| None |

### AssignXBandToGuest

This call has been deprecated and replaced with a proper POST method.

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband/{xbandId}/{guestId} |
| **HTTP Method** | GET |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| xbandId | The IDMS xbandId of the band to assign. |
| guestId | The IDMS guestId of the guest to assign to the xband. |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/xband/1021/1658 |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Ok. The guest was successfully assigned to the xband. |

#### Response Body

|  |
| --- |
| None |

### RemoveXBandFromGuest

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband/unassign/{xbandId}/{guestId} |
| **HTTP Method** | GET |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| xbandId | The IDMS xbandId of the xband to unassign. |
| guestId | The IDMS guest Id to unassign from the xband. |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| None |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/xband/ |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Guest was unassignedfrom the xband. |

#### Response Body

|  |
| --- |
| None |

### AssignXBand

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband |
| **HTTP Method** | POST |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| None |  |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| **{**  **"xbandId":1021,**  **"guestId":1658**  **}** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/xband/ |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Created. The xband/guest association was created. |
| 400 | Either the xbandId or the guestId were incorrect in the JSON post object. |

#### Response Body

|  |
| --- |
| None |

### GetXBandByxBandId

### GetXBandByBandId

### GetXBandBylrid

### GetXBandBytapid

### GetXBandBysecureid

### GetXBandByUid

All of these get calls retrieve the same xband object by the various types of IDs associated with an xband.

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xband/{id}  <idms-context>/IDMS/xband/bandid/{id}  <idms-context>/IDMS/xband/lrid/{id}  <idms-context>/IDMS/xband/tapid/{id}  <idms-context>/IDMS/xband/secureid/{id}  <idms-context>/IDMS/xband/uid/{id} |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | The specific id type to look up an xband by. i.e., bandId, secureId, TapId, LRId or Uid. |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| **None** |

#### ****Examples****

|  |
| --- |
| http://localhost:8080/IDMS/xband/1021 |
| http://localhost:8080/IDMS/xband/bandid/178E095AC574E100 |
| http://localhost:8080/IDMS/xband/lrid/FFECF57200 |
| http://localhost:8080/IDMS/xband/tapid/F01ED5BF9AAF80 |
| http://localhost:8080/IDMS/xband/secureid/411480721 |
| http://localhost:8080/IDMS/xband/uid/7A12C8 |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | The xband was found and a json object with xband information was returned. |
| 404 | No xband was found for the given id. |

#### Response Body

|  |
| --- |
| {  "state": "Active",  "publicId": "1734F5AC574E100",  "xbandId": 1021,  "bandType": "Guest",  "xbandRequestId": "178E095AC574E100",  "shortRangeTag": "F01ED5BF9AAF80",  "longRangeTag": "FFECF57200",  "secondaryState": "Active",  "productId": "178E095AC574E100",  "assignmentDateTime": 1335531244047,  “secureId” : 411480721,  "links": { },  "guests": [  {  "name": {  "lastName": "RUSSEL",  "firstName": "PATOLOT"  },  "status": "Active",  "guestId": 1658,  "partyId": 0,  "visitCount": 26,  "links": [  {  "name": "ownerProfile",  "href": "/guest/1658/profile"  },  { }  ]  }  ],  "bandId": "178E095AC574E100"  } |

## Celebration

Celebrations provide a form of data for a guest, for example if they are celebrating a birthday or anniversary in the park. IDMS is NOT the SOR (System of Record) for this data, however, none exists at this time so it is acting as a place holder until a new Celebrations SOR comes on-line.

|  |
| --- |
| @Path("/celebration")  **public** **interface** ICelebrationService {    /\*  \* return the celebration with the given id.  \*/  @SuppressWarnings("rawtypes")  @GET  @Path("/{id}”)  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse getCelebrationById(@PathParam("id")PathSegment pathSegment);    /\*  \* Create a celebration and add it to a guest.  \*/  @SuppressWarnings("rawtypes")  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse addCelebrationToGuest(String celebration);    /\*  \* updates an existing celebration. Celebration Id is required.  \*/  @SuppressWarnings("rawtypes")  @PUT  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse updateCelebration(String celebration);    /\*  \* Delete a guest from a celebration. A celebration Id is required.  \*/  @SuppressWarnings("rawtypes")  @DELETE  @Path("/guest")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse deleteCelebration(String celebration); |

### getCelebrationById

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/celebration/id;celebration-id={celebrationId} |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| celebrationId | The IDMS celebration id in the system. |

**Request Header**

**Request Body**

#### ****Example****

|  |  |
| --- | --- |
| http://localhost:8080/IDMS/celebration/id;celebration-id=18097 |  |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | The celebration was found in the system. |
| 404 | The celebration was not found in the system. |
| 500 | An internal error occurred. |

#### Response Body

|  |  |
| --- | --- |
| {  "links":{  "self":{  "name":"self",  "href":"/celebration/18097"  },  "reservation":{  "name":"reservation",  "href":"/resort-reservation/pid001"  }  },  "name":"18th Birthday",  "milestone":"18th Birthday",  "month":"1",  "day":"3",  "year":"2012",  "startDate":"2012-01-03",  "endDate":"2012-01-03",  "recgonitionDate":"2012-01-03",  "comment":"We has comment",  "guests":[{  "links":{  "self":{  "name":"self",  "href":"/guest/72B471DFD5D90C21FF92AF33302901BA/profile"  }  },  "role":"CELEBRANT",  "relationship":"OWNER,PARTICIPANT",  "lastname":"HUMINSKI",  "firstname":"CHUN"  }],  "type":"Birthday"  } | |
|  |

## PartyService

The party service provides the ability to create a guest party, add guests to a party and search for a party by party name (usually the email address).

|  |
| --- |
| @Path("/party")  **public** **interface** IPartyService {    /\*  \* Find a party by party name and return the party object with guests.  \*/  @SuppressWarnings("rawtypes")  @GET  @Path("/search/{partyName}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse findPartyByPartyName(@PathParam("partyName")String partyName);      /\*  \* Get a party by a party Id.  \*/  @SuppressWarnings("rawtypes")  @GET  @Path("/{id}")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetVisitByVisitId(@PathParam("id")**long** id);  /\*  \* Get a list of guests by a visitId.  \*/  @SuppressWarnings("rawtypes")  @GET  @Path("/{id}/guests")  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse GetGuestsByVisitId(@PathParam("id")**long** id);    /\*  \* Create a party taking a JSON Party Object.  \*/  @SuppressWarnings("rawtypes")  @POST  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse createParty(String stringParty);      /\*  \* Add a guest to a Party Object.  \*/  @SuppressWarnings("rawtypes")  @POST  @Path("/guest")  @Consumes({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  **public** JResponse AddGuestToParty(String guestPartyAssignString);    } |

### findPartyByPartyName

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/party/search/{partyname} |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | The name of the party. Usually the primary guest email address. In the case of non-unique values, this call will return the most recently created party. |

**Request Header**

**None**

**Request Body**

**None**

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/party/search/tomlantry@gmail.com |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | The party list object is returned for partyname received. |
| 404 | No visits were found for the given partyname. |

#### Response Body

|  |
| --- |
| {   * "count": 1, * "members": [   + {     - "guestId": 36001106,     - "links": {       * "ownerProfile": {         + "href": "/guest/36001106/profile"   }   * + - },     - "isPrimary": true   }   * ], * "partyId": 1, * "primaryGuestId": 36001106, * "partyName": "tomLantry@gmail.com"   } |

### GetVisitByVisitId

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/party/{id} |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | The IDMS party ID for a party. Please note that this party ID is also returned on a guest profile object, if the guest is assigned to party. |

**Request Header**

**None**

**Request Body**

**None**

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/party/1 |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | The party was found with this party ID. |
| 404 | No party was found with this party ID. |

#### Response Body

|  |
| --- |
| {   * "count": 1, * "members": [   + {     - "guestId": 36001106,     - "links": {       * "ownerProfile": {         + "href": "/guest/36001106/profile"   }   * + - },     - "isPrimary": true   }   * ], * "partyId": 1, * "primaryGuestId": 36001106, * "partyName": "tomLantry@gmail.com"   } |

### GetGuestsByVisitId

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/party/{id}/guests |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | The IDMS party ID to retrieve the guest list for. |

**Request Header**

**None**

**Request Body**

**None**

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/party/1/guests |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | Guests were found for this party ID. |
| 404 | No party was found, or guests weren’t found for this party ID. |

#### Response Body

|  |
| --- |
| [   * {   + "guestId": 36001106,   + "links": {     - "ownerProfile": {       * "href": "/guest/36001106/profile"   }   * + },   + "isPrimary": true   }  ] |

### createParty

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/party |
| **HTTP Method** | POST |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| None |  |

**Request Header**

**None**

**Request Body**

|  |
| --- |
| **{**  **"primaryGuestId":"1",**  **"partyName":"robert.lantry@disney.com"**  **}** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/party |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | The party was successfully created. |
| 403 | The party was not created due to bad data in the request body. |
| 500 | An internal error occurred while creating the guest. |

#### Response Body

|  |
| --- |
| URI to the created party in the form of IDMS/party/partyId |

### AddGuestToParty

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/party/guest |
| **HTTP Method** | POST |
| **Response Format** | HTTP Response |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| None |  |

**Request Header**

**None**

**Request Body**

|  |
| --- |
| **{**  **"partyId" : 19,**  **"guestId" : 50**  **}** |

#### ****Example****

|  |
| --- |
| http://localhost:8080/IDMS/party/guest |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 201 | Created. The guest was successfully added to the party. The party guest association was created. |
| 403 | The party guest association was not created. Either the party ID or the guest ID were not correct in the request body. |
|  |  |

#### Response Body

|  |
| --- |
| None |

## XBandView

The calls in XBandView are considered deprecated. They remain in the IDMS API for backward compatibility with a couple of older systems that still use them. New development should use the OneViewXBandView calls as these XBandView calls may be removed at some point in the future.

|  |
| --- |
| @Path(“/xbands”)  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/id/{id}")  **public** JResponse getXBandByXBandIdOld(@PathParam("id") String id) **throws** Exception;  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/{id}")  **public** JResponse getXBandByXBandId(@PathParam("id") String id) **throws** Exception;  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/tapid/{id}")  **public** JResponse getXBandByTapId(@PathParam("id")String id);  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/lrid/{id}")  **public** JResponse getXBandBylrId(@PathParam("id")String id);  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/bandid/{id}")  **public** JResponse getXBandByBandId(@PathParam("id")String id);  @SuppressWarnings("rawtypes")  @GET  @Produces({MediaType.*APPLICATION\_JSON* + ";charset=utf-8"})  @Path("/secureid/{id}")  **public** JResponse getXBandBySecureId(@PathParam("id") String id); |

### getXBandByXBandIdOld

### GetXBandByxBandIdTwo

### getXBandByXBandId

### getXBandByTapId

### getXBandBylrId

### getXBandByBandId

### getXBandBySecureId

|  |  |
| --- | --- |
| **Path** | <idms-context>/IDMS/xbands/id/{id}  <idms-context>/IDMS/xband/{id}  <idms-context>/IDMS/xbands/{id}  <idms-context>/IDMS/xbands/tapid/{id}  <idms-context>/IDMS/xbands/lrid/{id}  <idms-context>/IDMS/xbands/bandid/{id}  <idms-context>/IDMS/xbands/secureid/{id} |
| **HTTP Method** | GET |
| **Response Format** | application/json |

#### Request

|  |  |
| --- | --- |
| **Variable** | **Description** |
| id | The type of ID of the xband to retrieve. |

**Request Header**

|  |
| --- |
| **None** |

**Request Body**

|  |
| --- |
| **None** |

#### ****Examples****

|  |
| --- |
| http://localhost:8080/IDMS/xbands/id/1021 |
| http://localhost:8080/IDMS/xbands/1021 |
| http://localhost:8080/IDMS/xbands/tapid/F01ED5BF9AAF80 |
| http://localhost:8080/IDMS/xbands/lrid/FFECF57200 |
| http://localhost:8080/IDMS/xbands/bandid/ 178E095AC574E100 |
| http://localhost:8080/IDMS/xbands/secureid/411480721 |

#### Response Headers

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | The xband was found and returned as a JSON object. |
| 404 | No xband was found with the given ID. |

#### Response Body

|  |
| --- |
| {  "secureId": null,  "printedName": null,  "guests": [  {  "state": null,  "guestId": "1658",  "lastName": "RUSSEL",  "firstName": "PATOLOT",  "countryCode": null,  "address1": null,  "address2": null,  "city": null,  "active": true,  "createdBy": "simulator",  "createdDate": "2012-03-10",  "updatedBy": "simulator",  "updatedDate": "2012-03-10",  "birthdate": null,  "XBMSId": null,  "zip": null,  "entitlements": null,  "xbands": null,  "guestInfos": null  }  ],  "bandId": "178E095AC574E100",  "tapId": "F01ED5BF9AAF80",  "bandFriendlyName": null,  "active": true,  "createdBy": null,  "createdDate": null,  "updatedBy": null,  "updatedDate": null,  "lrid": "FFECF57200",  "xbandId": "1021"  }} |