API Documentenation for

League Management System

League Management System

**Date:**

**File Name:**

“League Management – API Document v1.docx”

**Version: 1.0**

**Document Revision History:**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author/Reviewer/Approver** | **Description/Change** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Introduction
   1. Purpose

**Purpose of the document**

The document is used to define the Requirements Specification for the NIS OMS project. This helps in providing a clear and complete picture of the Solution Project in terms of specifications with functional and non-functional requirements. The RS Process helps the Project Staff to understand the various requirements of the Solution.

**Intended Audience**

The Requirements Specification (RS) Process is defined for providing a clear and complete picture of the Solution System to the Project Team. The RS Process document is meant for Project Managers, Project Team members and my SQA.

* 1. Scope of This Document

The purpose of the RS document is to provide a description of the solution being specified and its purpose, including relevant benefits, objectives, and goals and to relate solution to corporate goals or business strategies.

* 1. References

The Requirement Specification and design document is built on the basis of the Clients Requirements Definition, as outlined in our Fit-Gap Analysis Document.

The Related document is attached and a proper description is provided in order to make the communication clear between the client and contractor.

|  |  |  |  |
| --- | --- | --- | --- |
| S. No. | Document Name | Description | Location |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |

1. Functional Requirements
   1. User Management

API

* Create

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user |
| Description | Create user |
| Method | POST |
| Header | Content-Type: application/json |
| Request | {  "email": "mac9487@gmail.com",  "username": "Gupt",  "full\_name": "Shashank sharma",  "password": "Abc12345",  "office\_phone": "(541) 754-3010",  "home\_phone": "(541) 754-3010",  "cell\_phone": "(541) 754-3010",  "address1": "2050 Bamako Place",  "address2": "",  "city": "Washington",  "user\_state": "DC",  "country": "US",  "zip": "20521",  "height": "170cm",  "dob": "01/09/1990",  "gender": "M",  "preferred\_location": "Washington",  "handedness": "12",  "rating\_type": "USP",  "rating": "4.5",  "user\_others": "",  "strength": "",  "weakness": ""  } |
| Response | {  "res": “user created successfully”  } |
| Error Response |  |

* Update

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/{id} |
| Description | Update Order by magento id |
| Method | PUT |
| Request | {  "email": "mac9487@gmail.com",  "username": "Gupta",  "full\_name": "Shashank",  "password": "Abc12345",  "office\_phone": "(541) 754-3010",  "home\_phone": "(541) 754-3010",  "cell\_phone": "(541) 754-3010",  "address1": "2050 Bamako Place",  "address2": "",  "city": "Washington",  "user\_state": "DC",  "country": "US",  "zip": "20521",  "height": "170cm",  "dob": "01/09/1990",  "gender": "M",  "preferred\_location": "Washington",  "handedness": "12",  "rating\_type": "USP",  "rating": "4.5",  "user\_others": "",  "strength": "",  "weakness": ""  } |
| Response | {  "res": “user updated successfully”  } |
| Error Response |  |

* List

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user |
| Description | List all users |
| Method | GET |
| Request |  |
| Response | [  {  "email": "mac9487@gmail.com",  "id": "39fa6625-283e-4984-9b62-8a9f21977a01",  "username": "Gupt",  "age": "25",  "full\_name": "Shashank",  "office\_phone": "(541) 754-3010",  "home\_phone": "(541) 754-3010",  "cell\_phone": "(541) 754-3010",  "address1": "2050 Bamako Place",  "address2": "",  "city": "Washington",  "user\_state": "DC",  "country": "US",  "zip": "20521",  "height": "170cm",  "age": "30",  "dob": "01/09/1990",  "gender": "M",  "preferred\_location": "Washington",  "handedness": "12",  "rating\_type": "USP",  "rating": "4.5",  "user\_others": "",  "strength": "",  "weakness": ""  }  },  {  "username": "user",  "id": "3caec5a7-a0dd-447e-843d-ee94f942c0a7",  "email": "test@damcogroup.com",  "full\_name": "test",  "office\_phone": "(541) 754-3010",  "home\_phone": "(541) 754-3010",  "cell\_phone": "(541) 754-3010",  "address1": "2050 Bamako Place",  "address2": "",  "city": "Washington",  "user\_state": "DC",  "country": "US",  "zip": "20521",  "height": "170cm",  "age": "25",  "dob": "01/09/1990",  "gender": "M",  "preferred\_location": "Washington",  "handedness": "12",  "rating\_type": "USP",  "rating": "4.5",  "user\_others": "",  "strength": "",  "weakness": ""  }  ] |
| Error Response |  |

* List

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user?q=name |
| Description | List by name |
| Method | GET |
| Request |  |
| Response | [  {  "email": "mac9487@gmail.com",  "id": "39fa6625-283e-4984-9b62-8a9f21977a01",  "full\_name": "Shashank"  },  {  "id": "3caec5a7-a0dd-447e-843d-ee94f942c0a7",  "email": "test@damcogroup.com",  "full\_name": "test"  }  ] |
| Error Response |  |

* Retrieve User Detail

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/{id} |
| Description | List user detail by Id |
| Method | GET |
| Request |  |
| Response | {  “id”: “hiuebguioerg-wdkjbfjbkg-dqewdaf”,  "email": "mac9487@gmail.com",  "username": "Gupta",  "full\_name": "Shashank",  "password": "Abc12345",  "office\_phone": "(541) 754-3010",  "home\_phone": "(541) 754-3010",  "cell\_phone": "(541) 754-3010",  "address1": "2050 Bamako Place",  "address2": "",  "city": "Washington",  "user\_state": "DC",  "country": "US",  "zip": "20521",  "height": "170cm",  "dob": "01/09/1990",  "gender": "M",  "preferred\_location": "Washington",  "handedness": "12",  "rating\_type": "USP",  "rating": "4.5",  "user\_others": "",  "strength": "",  "weakness": ""  } |
| Error Response |  |

* Delete User

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/{id} |
| Description | Delete a user |
| Method | DELETE |
| Request |  |
| Response | {  "res": “Player deleted successfully”  } |
| Error Response | {  "res": “Player is involved in one or more league”  } |

* User Login

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/login |
| Description | Login API |
| Method | POST |
| Request | {  "email": "mac9487@gmail.com",  "password": "Abc12345",  } |
| Response | {  "res": {  “id”: “kdjkhbgoje-kjnr45-212h1khj2gb1h”,  "email": "shashanks1@damcogroup.com",  "username": "sharma",  "full\_name": "Shashank"  }  } |
| Error Response | {  "res": “Login information is incorrect”  } |

* Forgot Password

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/forgot-password |
| Description | Forgot Password API |
| Method | POST |
| Request | {  "email": "mac9487@gmail.com",  } |
| Response | {  "res": “Your password has been sent to your authenticated mail successfully”  } |
| Error Response | {  "res": “Provided email address does not exist, please check”  } |

* Change Password

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/user/change-password |
| Description | Change Password API |
| Method | POST |
| Request | {  "email": "[mac9487@gmail.com](mailto:mac9487@gmail.com)",  “old\_password”: “Abc12345”,  “new\_password”: “Damco12345”  } |
| Response | {  "res": “Your password has been sent to your authenticated mail successfully”  } |
| Error Response | {  "res": “Provided email address does not exist, please check”  } |

* 1. League Management

API

* Create

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league |
| Description | Create a league |
| Method | POST |
| Header | Content Type – application/json |
| Request | {  "league\_name": "Wimbledon",  "description": "bot",  "league\_location": "New York",  "round\_robin\_period\_from": "01/03/2018",  "round\_robin\_period \_to": "07/03/2018",  "playoff\_period\_from": "01/03/2018",  "playoff\_period\_to":"07/03/2018",  "categories": ["Mens & Womens", "Mens Singles", "Wormen Singles", "Men Doubles", "Women Doubles"],  "scoring\_point": "30.98"  } |
| Response | {  "res": “league created successfully”  } |
| Error Response |  |

* Update

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league/{league\_id} |
| Description | Update league by league id – The request will be PUT not patch, so need all request data instead of updated one. |
| Method | PUT |
| Request | {  "league\_name": "Wimbledon",  "description": "bot",  "league\_location": "New York",  "round\_robin\_period\_from": "01/03/2018",  "round\_robin\_period \_to": "07/03/2018",  "playoff\_period\_from": "01/03/2018",  "playoff\_period\_to":"07/03/2018",  "categories": ["Mens & Womens", "Mens Singles", "Wormen Singles", "Men Doubles", "Women Doubles"],  "scoring\_point": "30.98"  To add group in a league: add this request data while update league:  “groups”: [{  "group\_name": "Wimbledon",  "players":["shashank@damcogroup.com", "sam@damcogroup.com"]  },  {  "group\_name": "Us Open",  "players":["shashanks1@damcogroup.com", "sam@damcogroup.com"]  }]  } |
| Response | {  "res": “league updated successfully”  } |
| Error Response |  |

* List

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league |
| Description | List leagues |
| Method | GET |
| Request |  |
| Response | [{  "league\_location": "New York",  "categories": "['Mens & Womens', 'Mens Singles', 'Wormen Singles', 'Men Doubles', 'Women Doubles']",  "scoring\_point": "30,98",  "description": "bot",  "id": "00b228af-0b6f-4287-b8b7-9b78c45e14fd",  "league\_name": "Wimbledon",  "round\_robin\_period\_from": "01/03/2018",  "round\_robin\_period \_to": "07/03/2018",  "playoff\_period\_from": "01/03/2018",  "playoff\_period\_to": "07/03/2018",  “groups”: [{  "group\_name": "Wimbledon",  "players": ["shashank@damcogroup.com", "sam@damcogroup.com"]  },  {  "group\_name": "Us Open",  "players": ["shashanks1@damcogroup.com", "sam@damcogroup.com"]  }  ]  }, {  "league\_location": "user",  "categories": "['Mens & Womens', 'Mens Singles', 'Wormen Singles', 'Men Doubles', 'Women Doubles']",  "scoring\_point": "30,98",  "description": "bot",  "id": "28c96536-1c0f-4a23-96cd-6de47eaad898",  "league\_name": "US Open",  "round\_robin\_period\_from": "01/03/2018",  "round\_robin\_period \_to": "07/03/2018",  "playoff\_period\_from": "01/03/2018",  "playoff\_period\_to": "07/03/2018",  “groups”: [{  "group\_name": "Wimbledon",  "players": ["shashank@damcogroup.com", "sam@damcogroup.com"]  },  {  "group\_name": "Us Open",  "players": ["shashanks1@damcogroup.com", "sam@damcogroup.com"]  }  ]  }] |
| Error Response |  |

* Delete League

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league/{id} |
| Description | Delete a league |
| Method | DELETE |
| Request |  |
| Response | {  "res": “League deleted successfully”  } |
| Error Response |  |

* 1. Group Management

API

* Create

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league/group |
| Description | Create a league group |
| Method | POST |
| Header | Content Type – application/json |
| Request | {  "group\_name": "Wimbledon",  "players":["shashank@damcogroup.com", "sam@damcogroup.com"]  } |
| Response | {  "res": “Group created successfully”  } |
| Error Response |  |

* Update

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league/{group\_id} |
| Description | Update group |
| Method | PUT |
| Request | {  "league\_name": "Wimbledon",  "players":["shashank@damcogroup.com", "sam@damcogroup.com"]  } |
| Response | {  "res": “Group updated successfully”  } |
| Error Response |  |

* List

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league |
| Description | List leagues |
| Method | GET |
| Request |  |
| Response | [{  "id": "28c96536-1c0f-4a23-96cd-6de47eaad898",  "group\_name": "Wimbledon",  "players":["shashank@damcogroup.com", "sam@damcogroup.com"]  }, {  "id": "28c96536-1c0f-4a23-96cd-6de47eaad898",  "group\_name": "US Open",  "players":["shashank@damcogroup.com", "sam@damcogroup.com"]  }] |
| Error Response |  |

* Delete League

|  |  |
| --- | --- |
| URL | <https://8gmku063fh.execute-api.us-east-2.amazonaws.com/demo>/league/{id} |
| Description | Delete a group |
| Method | DELETE |
| Request |  |
| Response | {  "res": “Group deleted successfully”  } |
| Error Response |  |