# API Specifications Document

**For**

**OJAS HIRING MANAGEMENT SYSTEM**

### Ver no. 1.0

Disclaimer: This document is classified as 'internal' and used by Ojas Innovative Technologies Pvt. Ltd. employees only until and unless it is specified otherwise.

***Document Revision History***

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Author** | **Change Information** |
| 0.1 | 12-sep-2024 | Nagarjuna Reddy | Initial draft version |
| 1.0 | 12-sep-2024 | Nagarjuna Reddy | Baselined after receiving approval |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Reviewer** | **Name of the Approver** |
| 0.1 | 12-sep-2024 | Venkata Ravindranath Gundugola | Venkata Ravindranath Gundugola |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Template Revision History (For SEPG only)***

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of Author** | **Change Information** |
| 0.1 | 30-Nov-2021 | Vanima Prasad | Initial draft copy |
| 1.0 | 24-Dec-2021 | Vanima Prasad | Baselined after Receiving Approval |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver. No.** | **Date**  **(dd-Mmm-yyyy)** | **Name of the Reviewer** | **Name of the Approver** |
| 0.1 | 15-Dec-2021 | Srinivas Bollapragada | Srinivas Bollapragada |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Table of Contents***

1.0 Assest Search API Development............................................................5

2.0 Low Level Design Document 5

3.0 Objective 5

4.0 Process workflow 5

5.0 Sequence Diagram 5

6.0 Data Flow Diagram 5

7.0 API's Specifications.................................................................................. 5

API intro.................................................................................................5

Assumptions...........................................................................................5

API Details..............................................................................................5

Request.................................................................................................6

Response...............................................................................................6

8.0 Security Considerations..........................................................................6

9.0 conclusion..............................................................................................6

### 1.0 Asset Search API Overview

### The RRF (Request for Recruitment) API in the OHAMS project allows authorized users to submit new recruitment requests with details like job title, department, skills, and vacancies. Once submitted, the system processes the request, assigns it to users, and streamlines the recruitment process.

### 2.0 Low-Level Design

The **RRF POST Request API** integrates with the OHAMS system to handle recruitment requests by:

* **Input**: Accepts job title, department, skills, number of positions, experience level, and location.
* **Validation**: Checks for completeness and correct data format.
* **Database**: Stores request details in relevant tables.
* **Notifications**: Alerts stakeholders for action.
* **Response**: Provides a status and confirmation message.

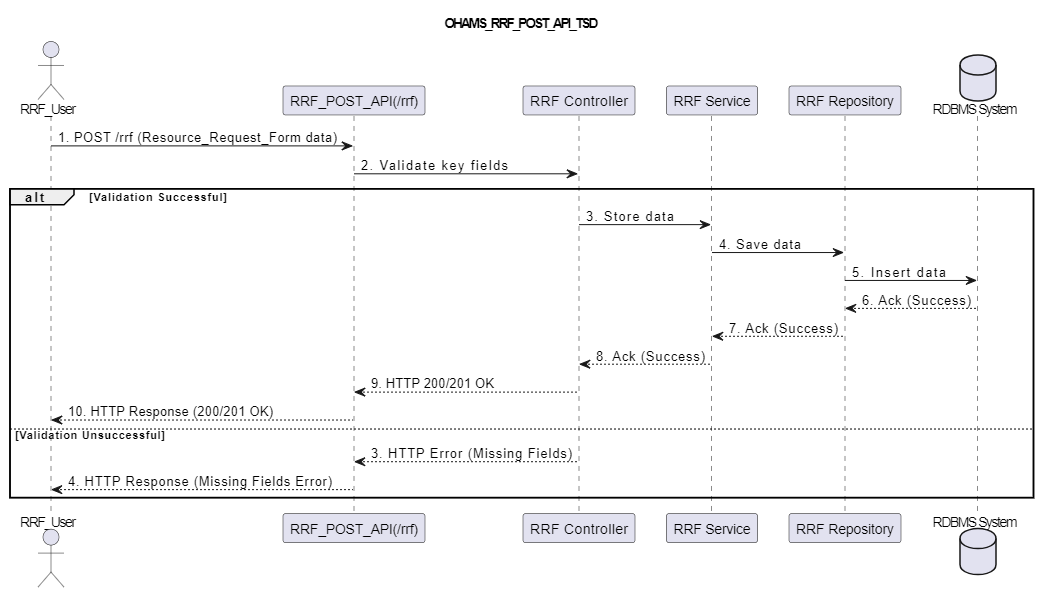
### 3.0 Objective

### The **RRF POST Request API** enables PMO and Tag Teams to quickly create and submit recruitment requests, streamlining hiring and improving resource and contract management

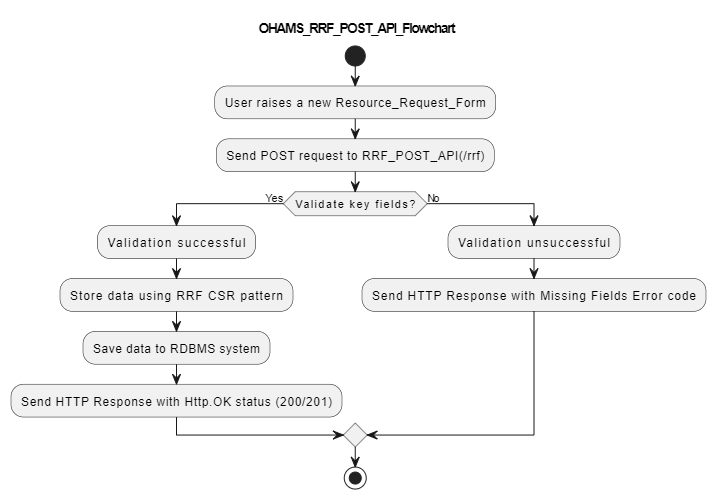
### 4.0 Workflow

1. User initiates a search from the OHAMS dashboard.
2. API processes search criteria.
3. Database is queried based on filters (e.g., asset ID, status).
4. Matching records are retrieved.
5. Results are displayed on the dashboard for further actions.

5.0 Sequence Diagram



6.0 Data Flow Diagram:



### 7.0 API Specifications

API Introduction:

This **POST RRF Request** helps stakeholders in the RRF process to search for and get asset data quickly based on specific criteria. It supports managing resource requests, contracts, and hiring processes within the OHAMS system.

Assumptions:

* Users have necessary access rights.
* Search filters are correctly applied to database fields.
* Only assets meeting the search criteria will be returned.
* Assets from different modules, including RRF, can be searched.

API Details:

* Endpoint: POST /api/v1/rrf
* Method: POST
* Purpose: Create a new Resource Request Form (RRF) entry in OHAMS.

Request:

* URL: /api/v1/rrf
* Headers:
  + Authorization: Bearer Token
  + Content-Type: application/json

8.0 Security Considerations

The API uses secure authentication with bearer tokens and role-based access control (RBAC). Sensitive data like client details and salary info is encrypted in transit and storage.

9.0 Conclusion

The Asset Search API enhances the RRF module in the OHAMS project, improving data availability, system efficiency, and secure operations.

1. Submit Order API

* + 1. *Main information*

|  |  |
| --- | --- |
| **Task** | Provide a concise description of what the API is designed to do within the OHAMS project. |
|  | Describe the business purpose and the value this API adds to the OHAMS system.    *Example*: "This API supports the automation of the RRF process, allowing the Senior  Management Team and PMO Team to efficiently manage resource requirements and track the  hiring process |
| **Related documents** | Include links or file paths to any related technical documents, API guidelines, or user manuals relevant to this API.  *Example*: "[OHAMS API Design Document](https://inc-word-edit.officeapps.live.com/we/link) and [Hiring Process Workflow](https://inc-word-edit.officeapps.live.com/we/link)." |
| **API Flow** | Break down how the API works step by step. Ensure the steps reflect the actual flow in OHAMS, such as creating, updating, or retrieving RRF data.  1. The hiring manager initiates a resource request via the OHAMS dashboard.  2. OHAMS sends a request to the API to create a new Resource Request Form (RRF).  3. The API stores the RRF details, including job roles, requirements, and timelines.  4. The API notifies relevant stakeholders (Senior Management, PMO) for review and approval.  5. Once approved, the API marks the RRF as active and sends notifications to the Tag Team.  6. The status of the RRF is updated in OHAMS and displayed on the dashboard. |
| **Sequence Flow** | Image of the sequence flow diagram or something similar |

* + 1. *Request/Response*

|  |  |  |  |
| --- | --- | --- | --- |
| **API Name** | **Endpoint URL** | **Verb** | **Description** |
| The API name | The URL for this endpoint. Also specify the environments if there are more than one. | HTTP  Method | Short description of the API |
|  |  |  |  |
| e.g. Submit Order API | http://192.168.1.98:8084/hiring/api/postRRF | e.g. /POST | E.g.  The API will submit an order |
|  | Headers of the request/response Specify the authorization process.  Example:  Content-Type: application/json Authorization: Bearer <Bearer Token> | | |
| **Request Body** | **Request sample:**  {    "job-description": "Runn",    "customer-name": "Keep",    "open-positions": "3",    "closed-positions": "0",    "hiring-type": "Internal",    "job-level": "L1",    "budget": "4.0",    "primary-skills": "SQL",    "candidate-count": "0",    "requirement Name": "java1",    "published-on": "12-12-2024",    "visibility": "0",    "email-address": "SADFASD@gamil.com",    "created-by": "JAI BALAYA",    "employee Id": "0",    "job Title": "java",    "job-type": "Full-time",    "priority": "Medium",    "mode-of Work": "Remote",    "owner Of Requirement": "Srinivas B",    "city": "HYD",    "state": "TG",    "totalPositions": "3",    "requirement Status": "New",    "location": "ASDFASDF",    "experience": "5"  } | | |
| **Response Body** | **Response sample:**  {"status":"RRF Successfully Created"} | | |

## Output parameters

From Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Parameter | Data type | Mandatory | Description | Value options |
| 1 | status | Var char string | Y | COMPLETED, IN\_PROGRESS,  ERRORED | IN\_PROGRESS |
| Response:  {  “status”:”IN\_PROGRESS”  } | | | | | |

## Response codes

|  |  |  |
| --- | --- | --- |
| Responde code | Message | Meaning |
| 200 - OK |  | The request was successful |
| 400 – Bad Request | Missing argument! | Example will send this if any details are missing or there is a  mismatch with our records. |
| 401 – Authentication error | Authentication error | E.g. bad or missing token. |
| 404 – Not Found |  |  |
| 503 – Service Unavailable |  | The resource is currently busy, i.e. already being processed. |
| 500 – Internal Server Error | TBC | All other exceptions |