

egov-hrms

Objective:

The objective of HRMS is to provide a service that manages all the employees enrolled onto the system. HRMS provides extensive APIs to create, update and search the employees with attributes like assignments, service history, jurisdiction etc. HRMS can be treated a sub-set of the egov-user service, Every employee created through HRMS will also be created as a user in egov-user.

Requirements:

- Prior Knowledge of Java/J2EE
- Prior Knowledge of SpringBoot
- Prior Knowledge of REST APIs and related concepts like path parameters, headers, JSON etc
- Prior Knowledge of Kafka and related concepts like Producer, Consumer, Topic etc.

Setup:

Step 1: Start the ID-Gen service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-idgen>

Or port forward the ID-Gen service. This service is used to generate ID for some fields.

Step 2: Start the egov-mdms service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-mdms-service>

Or port forward the egov-mdms service.

Step 3: Start the egov-user service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-user>

Or port forward the egov-user.

Step 4: Start the egov-filestore service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-filestore>

Or port forward the egov-filestore service.

Step 5: Start the egov-localization service present in the following repo:

<https://github.com/egovernments/core-services/tree/master/egov-localization>

Or port forward the egov-localization service.

Configurable Properties:

Following are the properties in application.properties file in collection service which are configurable.

Property	Value	Remarks
egov.hrms.employee.app.link	https://mseva.lgpunjab.gov.in/employee/user/login	This is the link to the mseva app, which differs based on the environment .
egov.hrms.default.pagination.limit	200	This is the pagination limit on search results of employee search, it can be set to any numeric value without decimals.
egov.hrms.default.pwd.length	10	This is the length of password to be generated at the time

		of employee creation. However, please ensure this is in sync with the egov-user pwd policy.
open.search.enabled.roles	SUPERUSER,ADMIN	This is a list of Role codes that are allowed to perform an open-search in hrms.
kafka.topics.notification.sms	egov.core.notification.sms	Kafka Topic to push the sms request. Please ensure this is in sync with egov-notification-sms service.
kafka.topics.save.service	save-hrms-employee	Kafka topic to push the save request in hrms. Please ensure this in sync with the persister.
kafka.topics.update.service	update-hrms-employee	Kafka topic to push the update request in hrms. Please ensure this in sync with the

		persistier.
egov.idgen.ack.name	hrms.employeecode	Key to be configured in Idgen alongwith the ID format to generate employee code.
egov.idgen.ack.format	EMP-[city]-[SEQ_EG_HRMS_EMP_CODE]	Format to be configured in ID gen to generate employee code.

APIs:

1. **_create:**

API to create an employee in the system. Information about the employee such as - Basic Details, Jurisdiction, Assignment, Education, Service History, Tests etc are captured during this process.

1. **_update:**

API to update an employee in the system. Information about the employee such as - Basic Details, Jurisdiction, Assignment, Education, Service History, Tests etc are captured during this process.

NOTE: Deactivation / Reactivation of employee is part of the same update flow. Inorder deactivate / reactivate an employee, Deactivation / Reactivation details such as reason for deactivation / reactivation, deactivation / reactivation effective from, order number etc have to be passed.

1. **_search:**

API to search employees from the system. Information about the employee such as - Basic Details, Jurisdiction, Assignment, Education, Service History, Tests etc are retrieved during this process. This API provides both: protected and open search. Open search is allowed only for a set of roles that are configures, for others, it protected by default.

Postman Collection: <https://www.getpostman.com/collections/7205a669dbd888affe8b>

API Contract: <https://raw.githubusercontent.com/egovernments/egov-services/master/docs/rainmaker/egov-hrms/v1-0-0.yml>

Entities:

Jurisdiction: It is the geographical area within which an employee can operate. This area can range from a small zone with an ULB to the entire state or more.

Jurisdictions can be defined within multiple ULB, wherein multiple localities/zones/blocks can fall under one jurisdiction.

Assignment: It is the combination of Position, Designation and Department associated with an employee for a given period of time. An employee can have multiple assignments. However, at a given point in time only one can be called primary assignment or active assignment.

Service History: It is the record of all previous professional engagements of an employee. This maintains a history of the employee's professional experience.

Educational Qualification: It is the record of education of the employee. Contains details of the degree, year of passing, institution, scores, supporting documents etc.

Departmental Test: It is the record of all the departmental tests undertaken by the employee. In the govt, such tests are given under the scope of promotions, transfers etc.

Employee Document: Documents required to register an employee in the system. These documents can belong to assignment details, service history, educational details etc.

Configuration

MDMS:

Following is the list of masters required for HRMS:

common-masters:

1. Department -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/common-masters/Department.json>

1. Designation -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/common-masters/Designation.json>

egov-hrms masters:

1. Degree -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/Degree.json>

1. EmployeeStatus -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/EmployeeStatus.json>

1. EmployeeType -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/EmployeeType.json>

1. ReasonForDeactivation -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/DeactivationReason.json>

1. Departmental Tests -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/EmploymentTest.json>

1. Specialization -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/egov-hrms/Specalization.json>

access-control masters:

1. Roles -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/ACCESSCONTROL-ROLES/roles.json>

egov-location masters:

1. Boundary -

<https://raw.githubusercontent.com/egovernments/egov-mdms-data/master/data/pb/amritsar/egov-location/boundary-data.json> (Amritsar)

Localization:

Create message to be sent on employee creation.

```
Request: {"RequestInfo":{"apild":"asset-
services","ver":null,"ts":null,"action":null,"did":null,"key":null,"msgId":"search with from and to
values","authToken":"c1d189c7-c239-4e12-b5b6-
9c2eb004100a"},"tenantId":"pb","messages":[{"code":"hrms.employee.create.notification","message": "Hi
$employeename, Welcome to mSeva. Your profile has been successfully set-up : Username - $username
Password - $password Visit your profile at $aplink, Thank you."},"module":"egov-hrms","locale":"en_IN"]}}
```

URI: <https://egov-micro-dev.egovernments.org/localization/messages/v1/ create>

ROLEACTIONS:

egov-hrms/employees/_create - SUPERUSER

egov-hrms/employees/_update - SUPERUSER, EMPLOYEE

egov-hrms/employees/_search - SUPERUSER, EMPLOYEE, CITIZEN and all module specific roles (like GRO, DGRO) that needs employees to be displayed on the screen.

Persister configuration:

[HRMS Persister](#)

Dependencies:

User:

HRMS depends on egov-user for creating user on every create employee call. Similarly update user and search user whenever required. HRMS uses /user/users/_createnvalidate, /user/users/_updatenvalidate, /user/v1/_search APIs.

ID Gen: (Service has to be promoted)

HRMS makes use of egov-idgen for generating employee code (labelled as Employee ID on the screen.). Format of the id and the key are configurable, they're part of the application.properties file.

MDMS: (Service has to be restarted)

HRMS makes calls to egov-mdms-service to fetch required masters as stated above. These are significant in validations.

Localization:

HRMS makes calls to localization to fetch localised values, currently there's only one message to fetched which is the sms content to be sent on employee creation.

Migration

Please find attached migration script for migrating existing employee data to egov-hrms.

For Details on entities and constraints go through [README.md](https://github.com/egovsolutions/business-services/tree/master/egov-hrms) at <https://github.com/egovsolutions/business-services/tree/master/egov-hrms>.

Interaction Diagram



