

# **EHR.Network**

Open | Scalable | Private | Secure

## Basic howtos

*by*

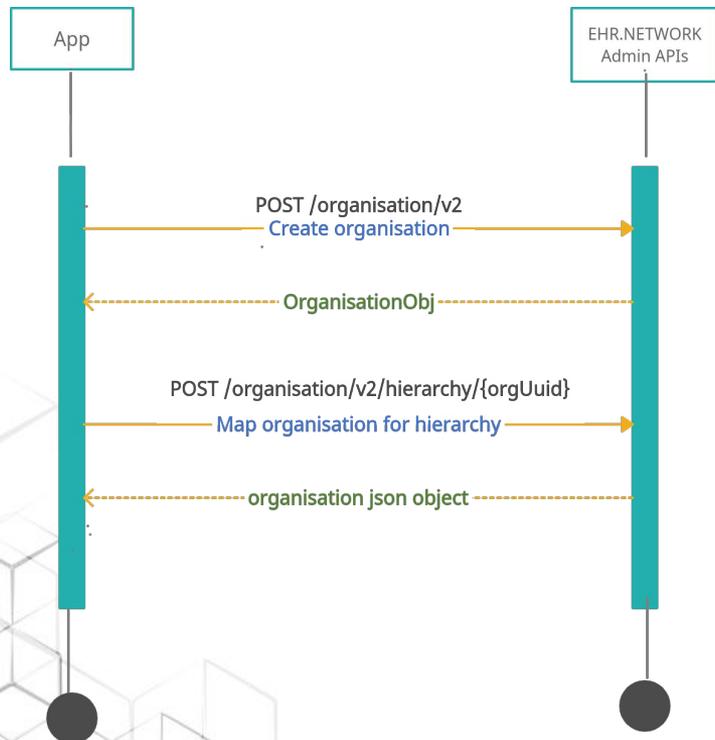


HealthLife

# Contents

1. Create a new organisation
2. Create a new user & assign roles/professions in an organisation
3. Register a patient & map to an organization
4. Search patients
5. Create visit/encounter
6. Add clinical data & link to a visit/encounter
7. Create task

# 1 - Create organisation



- Create a new organisation
- Platform also supports creating child organisations by passing parent organisationUuid.

# Services and resources used

## Services:

- AAA
- DGS

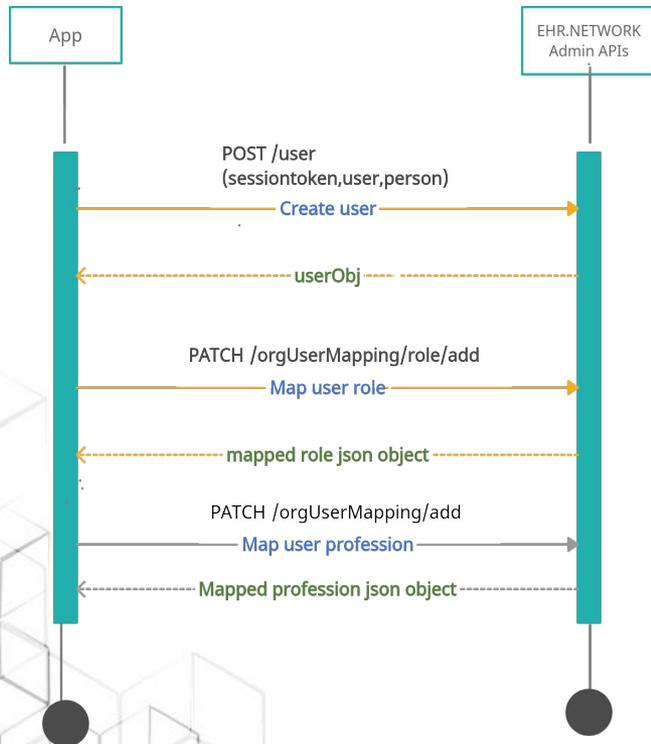
## Resources:

- OrganisationService Resource

## APIs used

- Create organisation -  
<https://docs.ehr.network/apidocs/organization.html#operation/post-organization-v2>
- Update hierarchy -  
<https://docs.ehr.network/apidocs/organization.html#operation/post-organization-v2-hierarchy-uuid>

## 2 - Create user, Map User role & profession



- Create a new user. This creates a person and associates that with the user object.
- Once created, this user has to be mapped to an organization with a **role** and **profession** to enable the required authorizations.

# Services and Resources used

## Services:

- AAA - To authorize token
- DGS - To stores user demographics information

## Resources:

- UserService Resource
- OrgUserMappingService Resource

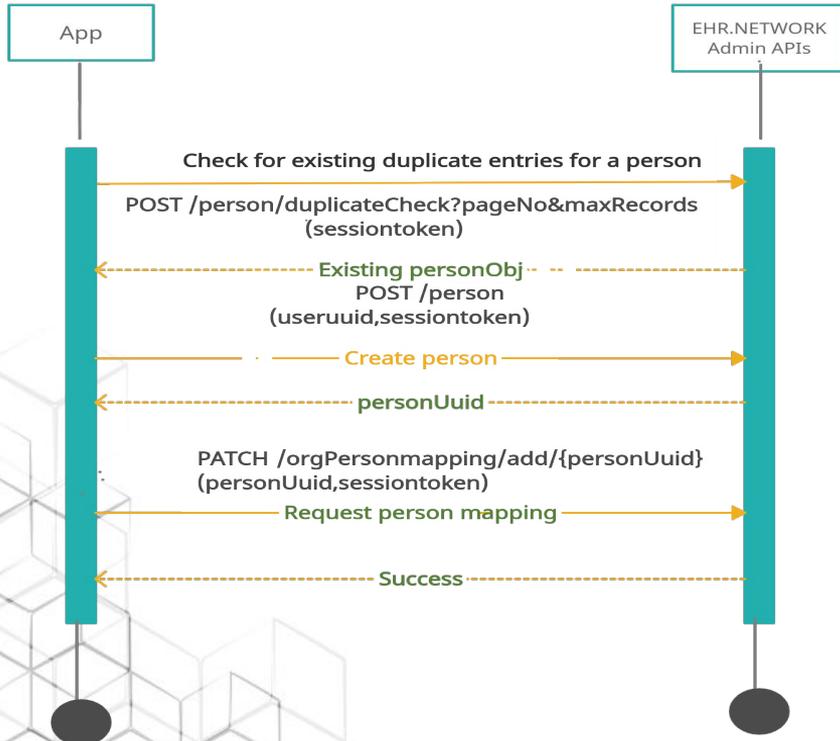
# APIs used

- Create user -  
<https://docs.ehr.network/apidocs/user.html#operation/post-user>
- Map user role -  
<https://docs.ehr.network/apidocs/user.html#operation/patch-orgUserMapping-role-add>
- Map user profession -  
<https://docs.ehr.network/apidocs/user.html#operation/patch-orgUserMapping-profession-add>

## 3 - Patient registration

- We have to register a patient before data can be added to their EHR.
- After logging into an organisation, register person involves checking for existing duplicate entries for a patient, mapping his/her personUuid to an organisation.
- This will make the patient visible to the users in the organisation and also allow the users to work with them.

# Patient registration- Design & Workflow



- Check for the duplicates entries of an existing patient in an organisation.
- Make a request by calling the duplicate check api, returns the personObject with givenName, email & phoneNumber.
- Create a person if not existing in an organisation which return personUuid.
- Patient/Person has to be mapped to the respective organisation by sending valid personUuid.

# Services & Resources used

## Services:

- **AAA** - Authorization (Token)
- **DGS** - Person which has to be created is sent to DGS which in return sends the personUuid as a response.
- **EHR** - For every person, EHR will create a unique ehrlid.

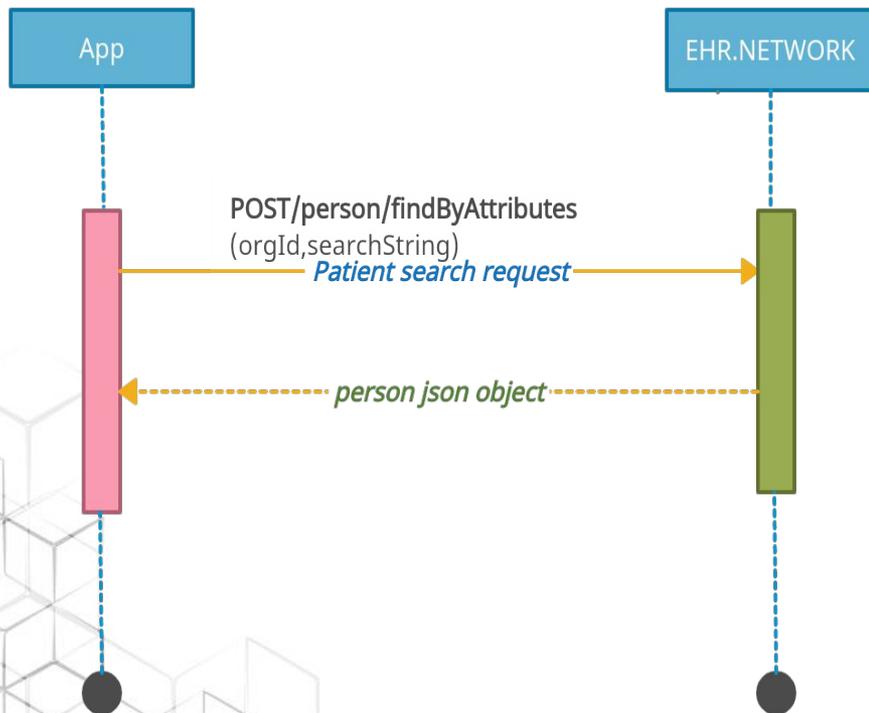
## Resources:

- PersonService resource
- OrgPersonMappingService Resource

# APIs used to register patient flow

- **Duplicate check** -  
<https://docs.ehr.network/apidocs/person.html#operation/person/duplicateCheck>
- **Create person** -  
<https://docs.ehr.network/apidocs/person.html#operation/person>
- **Map person to current organisation** -  
<https://docs.ehr.network/apidocs/person.html#operation/patch-orgPersonMapping-add-personUuid>
- **Map person to an organisation** -  
<https://docs.ehr.network/apidocs/person.html#operation/patch-orgPersonMapping-add-personUuid-orgUuid>

## 4 - Search patient



- Search for a person in a logged in organisation by sending the parameters in `searchString`.
- `searchString` accepts `givenName`, `emailId`, `phoneNumber` of a person.
- If any of the parameters matches with the existing persons, it returns `person json object`.

# Services & Resources used

## Services:

**DGS** - Search for an existing persons in a logged in organisation by taking the parameters requested and return the valid person DGS information.

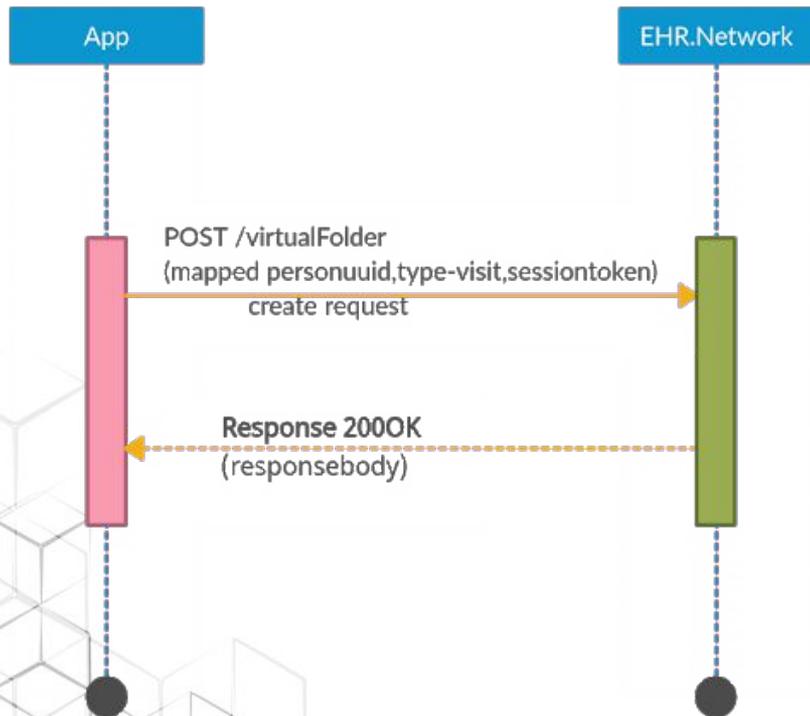
## Resources:

- PersonService resource

# APIs used to search person

- **Search person-**  
<https://docs.ehr.network/apidocs/person.html#operation/person/search>
- **Search person using string -**  
<https://docs.ehr.network/apidocs/person.html#operation/get-person-find-ByAttributes>

## 5 - Create Visit/Encounter



- Create a new visit/encounter
- VirtualFolders can be used to create a folders and contain IDs of compositions(sets of EHR data) that they contain.
- VirtualFolders are used to group compositions in ways that applications want to manage their data

# Services & Resources used

## Services:

**Virtualfolder** - Creates a virtualfolder of type 'visit', 'encounter', 'episode', 'event'.

## Resources:

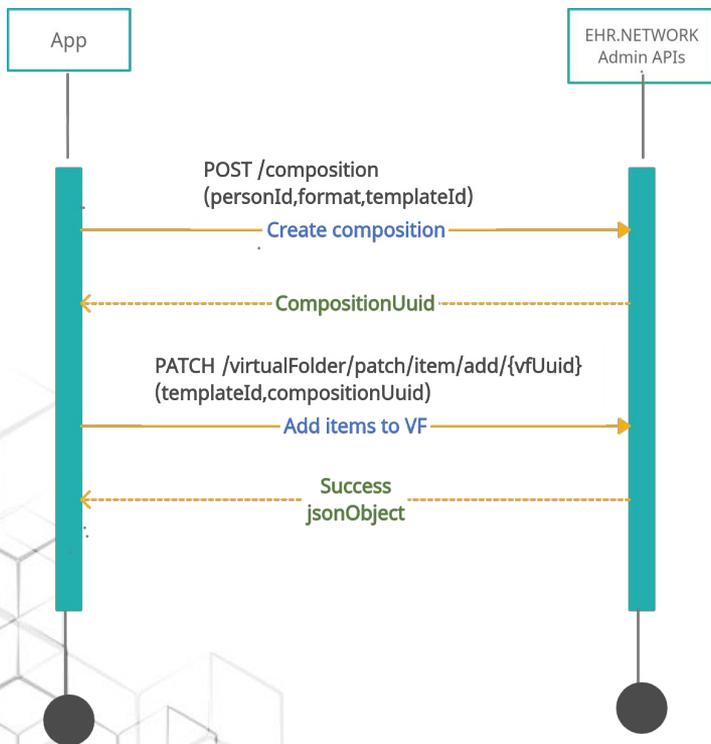
- VirtualfolderService Resource

# APIs used in Virtual folder

## Create Virtualfolder -

<https://docs.ehr.network/apidocs/ehr.html#operation/post-virtualFolder>

## 6 - Add composition & map to a visit



- Record clinical data for an encounter, which return the compositionUuid.
- Next, we can add recorded clinical data as an item in virtualfolder by sending valid templateId and compositionUuid as a json object.

# Services & Resources used

## Services:

- Ethercis
- Virtualfolder
- AAA

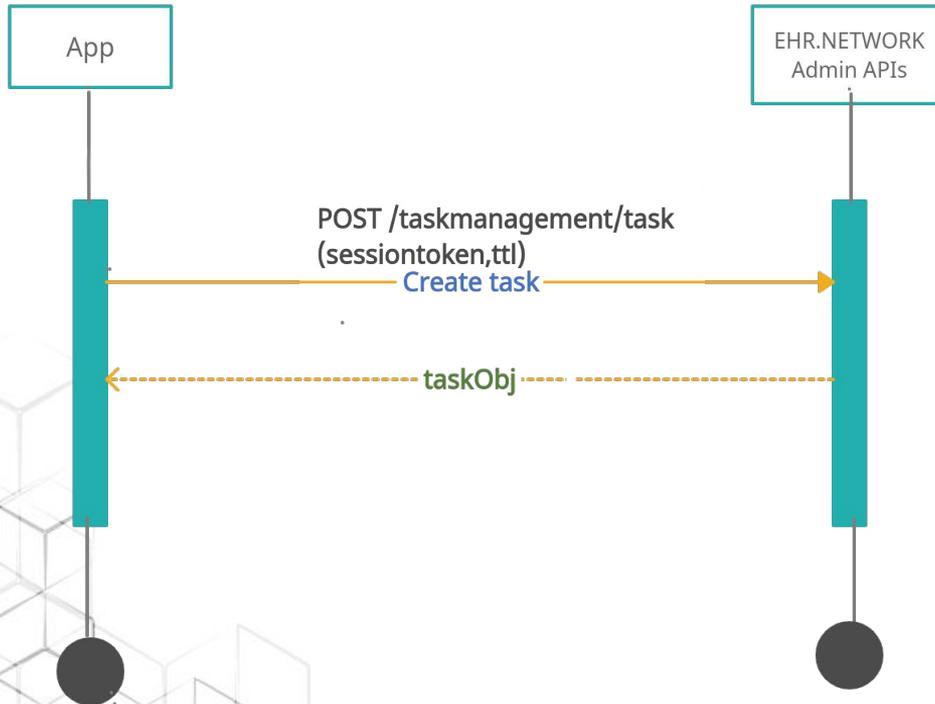
## Resources:

- ContributionService Resource
- VirtualfolderService Resource

## APIs used to add items

- Create composition -  
<https://docs.ehr.network/apidocs/ehr.html#operation/post-composition>
- Add items to VF -  
<https://docs.ehr.network/apidocs/ehr.html#operation/patch-virtualFolder-patch-item-add-uuid>

## 7 - Create task



- Create a task and assign it to other user(s).
- You can also specify the subject(s) for whom the task is planned.
- The status field is to set and track the status of the task and initiate workflow actions.

# Services and Resources used

## Services:

- AAA
- Taskmanager

## Resources:

- TaskmanagerService Resource

## APIs used

- Create task -  
<https://docs.ehr.network/apidocs/task.html#operation/post-taskmanagement-task>