**Call For Service (CFS)**

**Call For Service (CFS) management service**

**Technical Specifications Document**

**Prepared by**

**DXC**

1. **Business Requirement Description**

Dispatch++ team is building a Call For Service (CFS) management service with the following requirements

1. Dispatcher should be able to create a CFS with the following information: event number, event type (with type code), event time, dispatch time, responder.
2. Dispatcher should be able to search for CFS within a time range with paging and sorting order.
3. Dispatcher should be able to search for CFS that assigned to a responder.
4. CFS belongs to different agencies are not allowed to be exposed to other agencies.
5. Dispatcher and responder should belong to only one agency.

Your task is to design CFS service to support the above requirements and implement the API to support CFS search by time range.

1. **Base Functionality Overview**
2. Dispatcher should be able to create a CFS with the following information: event number, event type (with type code), event time, dispatch time, responder.
3. Dispatcher should be able to search for CFS within a time range with paging and sorting order.
4. Dispatcher should be able to search for CFS that assigned to a responder.
5. CFS belongs to different agencies are not allowed to be exposed to other agencies.
6. Dispatcher and responder should belong to only one agency..
7. **Requirement Overview**

Your task is to design CFS service to support the above requirements and implement the API to support CFS search by time range.

**2. Business Requirements Solution Description**

* 1. **Base Functionality**
* Authentication API
  + JWT check token for calling API (Stateless)
  + Authenticate API to login into system.
  + Register Account for Dispatcher.
* Agency API – Get information’s Agency for Dispatch process
  + Create API
  + Get All Agency
  + Search Agency by Name , Phone , ID Card
* Dispatcher API
  + Create API
  + Search CFS event
  + Search Responder CFS Event

**3. Tables**

* 1. **Database**
* MongoDB
  1. **Table**
     1. **Agency**
* Agency is a Agency table that maintain the parameters solely regarding information type

|  |  |  |
| --- | --- | --- |
| Item Code | Item Long Description | Item Short Description |
| **cfs\_agency** | CFS Agency information | CFS Agency information |

The new fields and definition.

|  |  |
| --- | --- |
| **Field** | **Description** |
| ID | A protected field which displays id. |
| Phone | A protected field which displays phone number |
| idCard | A protected field which displays agency’s id card. |
| name | A protected field which displays agency’s name. |
| createdBy | A protected field which displays dispatcher. |
| createdDate | Default value of created date – current date |
| lastModifiedBy | A protected field which displays modified dispatcher. |
| lastModifiedDate | Default value of last modified date – current date |

* + 1. **CFS Event**
* CFS Event is Call for service event table that maintain the parameters solely regarding event information type

|  |  |  |
| --- | --- | --- |
| Item Code | Item Long Description | Item Short Description |
| **cfs\_event** | CFS Event information | CFS Event |

The new fields and definition.

|  |  |
| --- | --- |
| **Field** | **Description** |
| ID | A protected field which displays id. |
| agencyId | A protected field which displays agency id |
| eventNumber | A protected field which displays event number |
| eventTypeCode | A protected field which displays event type code. |
| eventTime | A protected field which displays event time |
| dispatcherTime | A protected field which displays dispatcher time |
| responder | A protected field which displays responder code. |
| createdBy | A protected field which displays dispatcher. |
| createdDate | Default value of created date – current date |
| lastModifiedBy | A protected field which displays modified dispatcher. |
| lastModifiedDate | Default value of last modified date – current date |

* + 1. **User**
* User is a User table that maintain the parameters solely regarding user information type

|  |  |  |
| --- | --- | --- |
| Item Code | Item Long Description | Item Short Description |
| **cfs\_user** | User information | User model |

The new fields and definition.

|  |  |
| --- | --- |
| **Field** | **Description** |
| ID | A protected field which displays id. |
| userName | A protected field which displays user name |
| password | A protected field which displays password |
| enabled | A protected field which displays enabled type |
| role | A protected field which displays role |
| createdBy | A protected field which displays dispatcher. |
| createdDate | Default value of created date – current date |
| lastModifiedBy | A protected field which displays modified dispatcher. |
| lastModifiedDate | Default value of last modified date – current date |

**4. Technology Used**

* Java, Spring Boot , Rest API , Swagger , JWT
* MongoDB , Spring Data MongoDB

**5. Test Scenario**

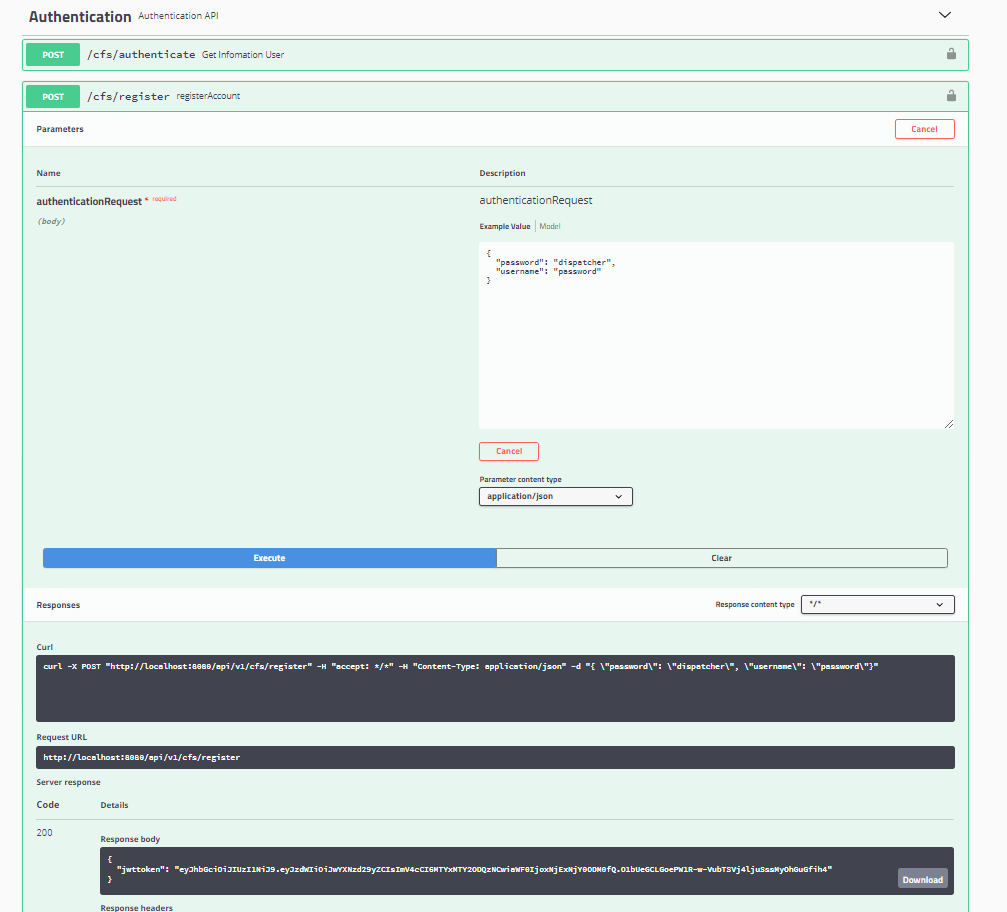
1. **Authentication API .**
   * + - 1. **Register Dispatcher API** .
     1. Json Model.

{

"password": "dispatcher",

"username": "password"

}

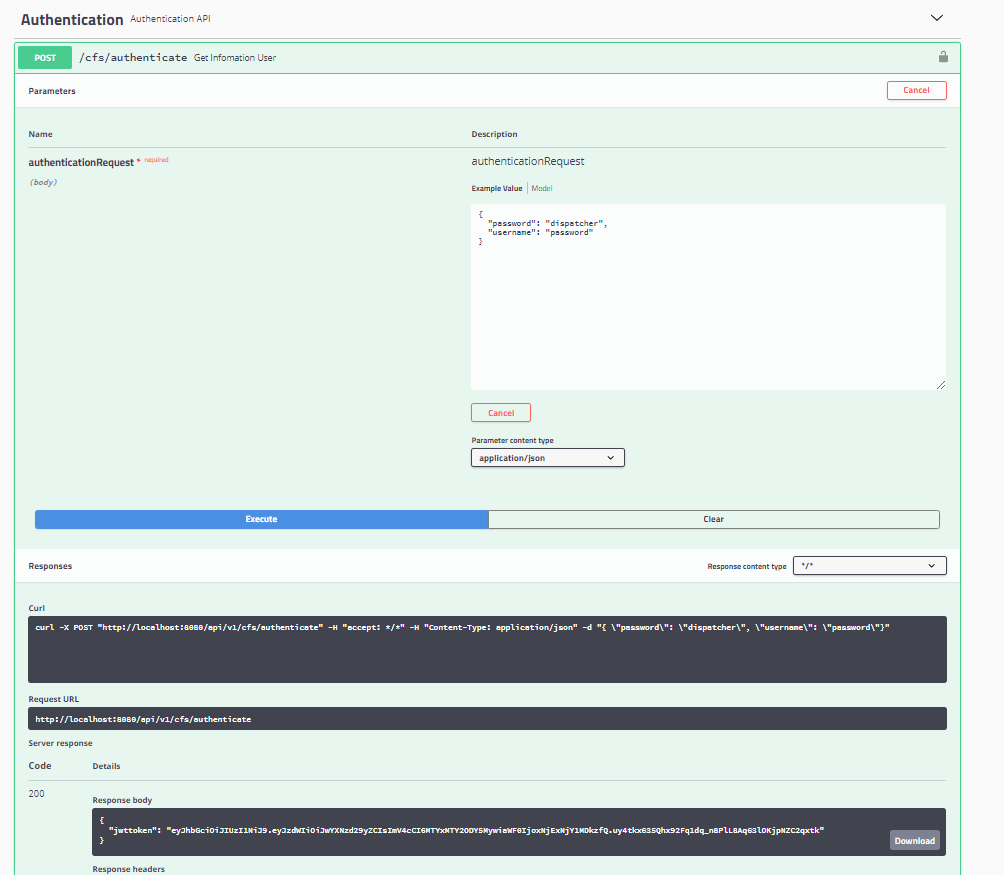
* + 1. API - **http://localhost:8080/api/v1/cfs/register**
    2. Result   
       
       - 1. **Login API** .
    3. Json Model.

{

"password": "dispatcher",

"username": "password"

}

* + 1. API - **http://localhost:8080/api/v1/cfs/authenticate**
    2. Result   
       

1. Dispatcher API
   * + - 1. **Create CFS event API** .
     1. Json Model.

{

"agency\_id": "4f9b99eb-490a-484e-bade-15e3841dfda9",

"event\_id": "562c89de-f140-4482-8ef5-5f1703b286b6",

"event\_number": "0000001",

"event\_type\_code": "SMO",

"event\_time": "2020-11-25 07:36:04.193",

"dispatch\_time": "2020-11-26 13:55:46.466",

"responder": "OFFICER\_001"

}

* + 1. API - [**http://localhost:8080/api/v1/cfs/dispatcher/create**](http://localhost:8080/api/v1/cfs/dispatcher/create)
    2. Result

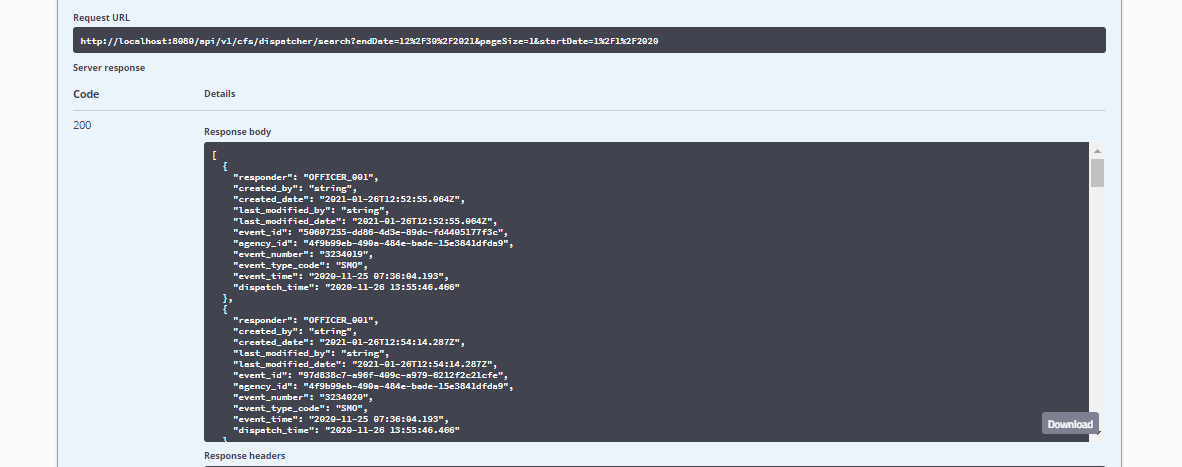




* + - * 1. **Search CFS event by time API** .
    1. API

[http://localhost:8080/api/v1/cfs/dispatcher/search? startDate=1/1/2020&endDate=12/30/2021](http://localhost:8080/api/v1/cfs/dispatcher/search?%20startDate=1/1/2020&endDate=12/30/2021)&pageSize=1

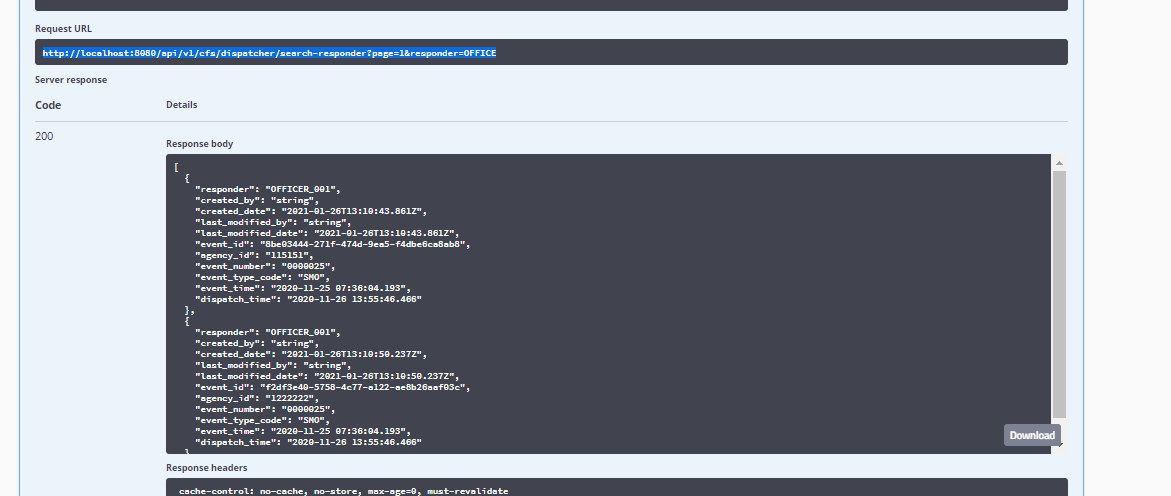
* + 1. Result



* + - * 1. **Search CFS event by Responder API** .
    1. API

<http://localhost:8080/api/v1/cfs/dispatcher/search->responder?page=1&responder=OFFICE

* + 1. Result



1. **Appendix**

N/A

1. **Help Text**

Git hub repo :

#### <https://github.com/MinhTin/cfs-service-api.git>