
PACER

Release 0.1

Michael Riley

May 08, 2024

CONTENTS

1	What is PACER?	3
2	Contents	5
2.1	PACER Client	5
2.2	PACER Server	39
	HTTP Routing Table	45

This document exists as a guide to help users understand the **P**ublic **H**ealth Automated **C**ase **E**lectronic **R**eporting tool, providing user and developer documentation.

WHAT IS PACER?

PACER is a client-server architecture for collecting supplemental health care data for health department and public health organizations.

PACER describes both a client component, as well as a server component. PACER client consists of a local database, an ELR receiver, a manager api, and a ui frontend. The client program can be configured to connect to one or many separate PACER Server. PACER server is a hosted supplemental service for FHIR EHR servers, designed to retrieve supplemental information for relevant cases data

For users, You can review the [PACER Client](#) section and `:doc: server` section for a detailed breakdown of services within each components For developers, direct installation instructions can be found in the [Client Installation Instructions](#) and [Server Installation Instructions](#) page.

Note: This project is under active development.

2.1 PACER Client

2.1.1 Client Workflow

There are two components in a PACER platform, PACER-server and PACER-client. This section covers PACER-client (green section in Figure 1). PACER-client is deployed in the public health department network and controlled by health department IT.

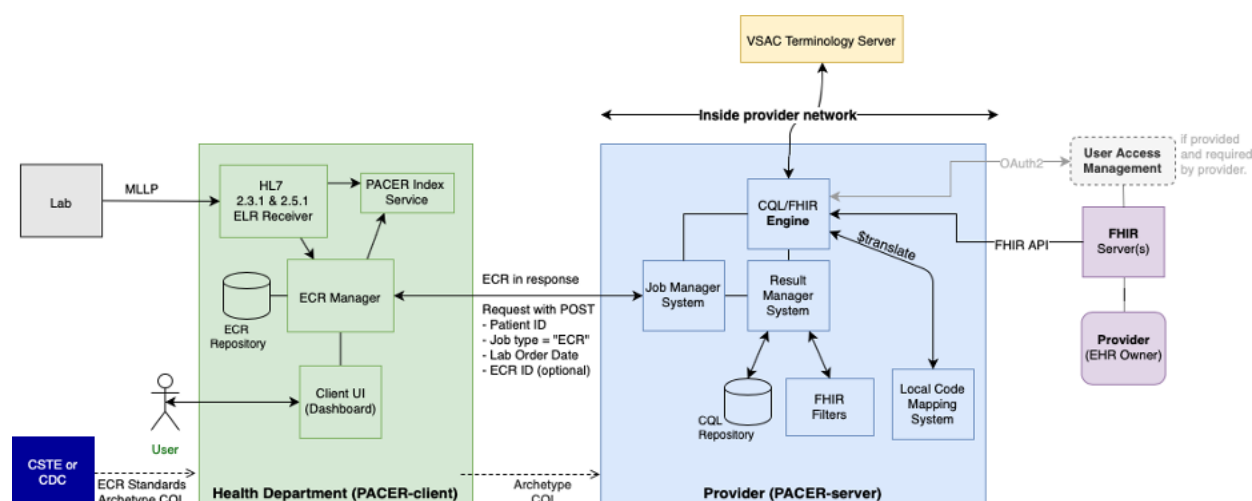


Figure 1: Overall PACER Architecture

Overall PACER workflow is shown in Figure 2. PACER-client starts at step #7 in Figure 2. When the PACER-client receives an STD electronic lab result (ELR) message, ECR operation is triggered, and initial ECR is created. Then, a PACER request is constructed and inserted into the job queue. Current version of PACER-client marks the job completed when an ECR from PACER-server is received. Otherwise, the job will be marked as failure.

The ECR response from PACER-server will be merged into the initial ECR and augment the ECR content. The PACER requests are stored in the job queue and never removed. Therefore, the requests can be restarted by changing the request status in the queue. This can be done either by manually triggering the query or next ELR for the same case.

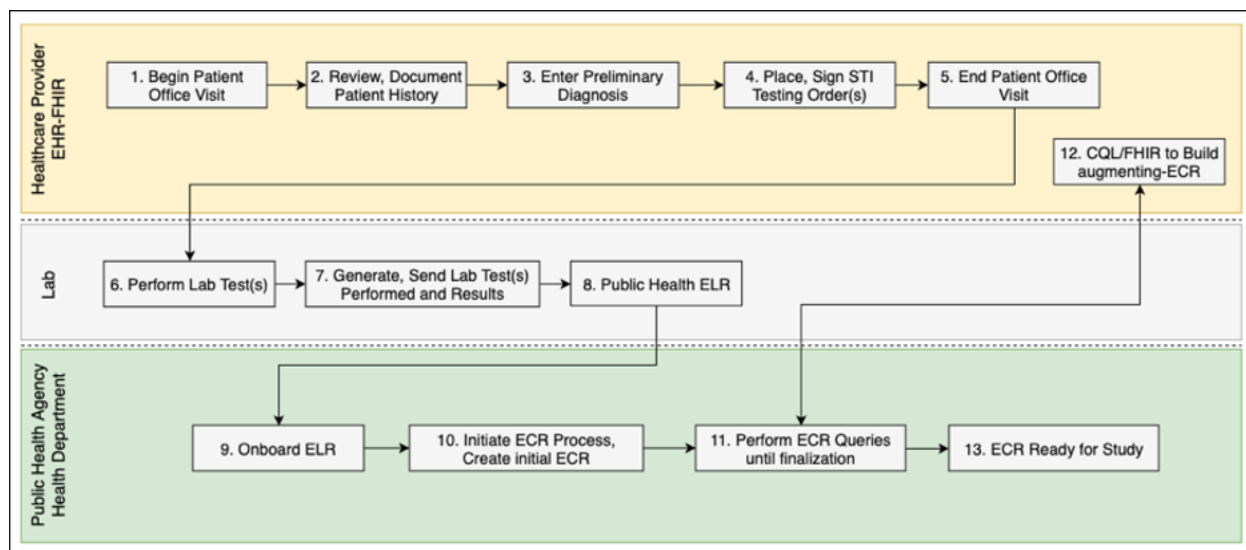


Figure 2: PACER Workflow

2.1.2 Installation and Configuration

PACER client supports both Linux and Windows platform. Mac is considered as a Linux environment. Please download appropriate package from GitHub repositories. Repository URLs as follows.

- Windows: <https://github.com/PACER-PHII/PACER-client-win.git>
- Linux or Mac: <https://github.com/PACER-PHII/PACER-client.git>

For the latest development version, please clone the repository with `git clone <repository_url>` from the command line in Linux or GitHub Desktop in Windows or Mac. However, if more tested version is wanted, go to the repository from browser and download either zip or gzip file from the released version. Link to the released version is available on the right middle side of the repository page.

Please note that this is a pilot project. The released version may have unseen issues. Contact the developers if any issues occur.

Database

Database must be created and properly configured before you proceed to deploy PACER-client service components. Any relational database can be used. PACER-client is tested with MS SQL and PostgreSQL server.

Note:

- PostgreSQL can be downloaded from <https://www.postgresql.org/download/>
- MS SQL is available at <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Database tables will be automatically created by ECR Manager. However, the schema must exist. In the database (MS-SQL or PostgreSQL), create a schema called “ecr” and setup a user for PACER-client.

Whether MS SQL or PostgreSQL is used, a schema needs to be named as “ecr”. Tables will automatically be created by ECR manager under the “ecr” schema.

For MS SQL server, in order to use Windows Authentication for MS SQL, make sure “ecr” database is owned by the account that will run PACER-client or the account has a db writer/reader permission. Please note that the schema name also needs to be “ecr”

Windows Server 2019

PACER-client uses a wrapper to run the Java application as a window service. Windows Service Wrapper (WinSW) is used for the wrapper.exe. All PACER-client components in this repository already have this wrapper application. Thus, nothing needs to be done for this wrapper. If you want to learn about the WinSW, please refer to <https://github.com/winsw/winsw>

Prerequisites

OpenJDK installation

Java runtime environment is required to run this service. Either JRE or JDK needs to be installed.

- Go to <https://docs.microsoft.com/en-us/java/openjdk/download>
- Download OpenJDK17 msi file (microsoft-jdk-17.0.2.8.1-windows-x64.msi) to install OpenJDK.
- After installation, type `java --version` at the command line (or powershell) to verify its installation

Authorization library installation

After *OpenJDK* installation, authorization library must be downloaded and installed in JDK bin folder. download the following dll file.

- `mssql-jdbc_auth-10.2.0.x64.dll` from <https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15> in zip or tar.gz file.
- Once the downloaded file is uncompressed, go to `auth/` folder and choose the one that meets your host machine hardware configuration.
- Copy the file, eg: `mssql-jdbc_auth-10.2.0.x64.dll`, to JDK's bin folder. If installation msi file is used for the Java installation, then the JDK bin folder should be at `C:\Program Files\Microsoft\jdk-17.0.2.8-hotspot\bin`

Certificates for SSL

All the traffic from client to external need to be on the secure socket layer. SSL requires a public signed certificate. This needs to be done by server side. However, if client network (or client firewall) manipulates the incoming certificate(s) and modify the chain of trust, then the PACER client applications won't be able to successfully establish the secure connections because the applications cannot validate the modified certificate(s).

In this case, the certificate must be trusted and added to the keystore. Here is the procedure to trust the certificate in the trust-store in Java.

- Export the server's certificate to file. This can be done by copying the server's endpoint URL and pasting the URL into the browser's address bar. Then, click on the lock icon to export the certificate to file.
- Copy the exported certified file to `C:\Program Files\Microsoft\jdk-17.0.2.8-hotspot\libsecurity` folder.
- Open Powershell, and go to `C:\Program Files\Microsoft\jdk-17.0.2.8-hotspot\libsecurity`
- Run the following command,

```
keytool -import -alias <alias_name> -file <exported cert filename> -keystore cacerts
```

- If you are asked for a password but haven't set it before, then the default password is “changeit”. In production environment, this password needs to be updated. The certificate will be added to the trust-store.

Deployment

There are three folders in the PACER-client-win repository. It is recommended to create a separate folder to copy the following three folders. In this way, when updates are made, the original folder can be kept as a backup folder.

The applications must be deployed or started in the following order.

- pacer-index-api
- ecr-manager
- elr-receiver

In each folder, there is an xml file. Open the XML file and make necessary changes for the environment variables. After all the environment variables are set correctly, run the executable (exe) file. This will create a service for the application. The account information should be correctly entered as well.

Warning: If any user access level is changed (for example, adding permission to the account used by PACER-client applications), then service(s) MUST be restarted so that the new changes to the account can be effective.

Detail application installation instructions for each application are provided below.

PACER-INDEX-API

At Powershell (in Admin mode), go to `pacer-index-api/` folder. And open `pacer-index-api.xml` file. Then, check the environment variables and change them as needed. `JAVA_HOME` should work as is if the same version of JDK in this document is used. If you are running this in the environment that security needs to be tightened, please change BASIC Auth parameters. `SERVER_PORT` can also be changed. Please note these variables are important as these will be used in another application. When everything is done, please run the following command at the Powershell.

```
.\pacer-index-api.exe install
```

This will install the `pacer-index-api` as a service. After the installation, open 'services' application (built-in app in Windows). From the list of services, locate the PACER Index API service. Right click on it and choose Properties. There, go to 'Log On' tab and choose 'this account' option. Then, add username and password. Please note that this account should have a permission to access local harddrive, otherwise the application will have an issue writing data to `PIDB.db` file.

pacer-index-api service configuration

`pacer-index-api` is used by `ecr-manager`. In order for `ecr-manager` to talk to PACER-server, we need to populate the `pacer-index-api` with PACER-server information. From a Chrome browser, go to "<http://localhost:8086/pacer-index-api/1.0.0/>" And, use the 'manage-api-controller' option to add the index entry in the following format. Use POST option.

```
1 {
2   "providerName": "<provider name if available>",
3   "identifier": "<provider identifier. ex, ORDPROVIDER|P49430>",
4   "pacerSource":
5   {
6     "name": "<any name>",
7     "serverUrl": "<PACER-server Job Management System URL>",
8     "security":
9     {
10      "type": "basic",
11      "username": "<username of list manager in the PACER server>",
```

(continues on next page)

(continued from previous page)

```

12     "password": "<password of list manager in the PACER server>"
13   },
14     "version": "1.0.0",
15     "type": "ECR"
16   }
17 }
18 }

```

If provider information is not available, or facility information is preferred for the indexing, the following format can be used. `providerName` is left to blank. And, the `identifier` is used for the facility information.

```

1 {
2   "providerName": "",
3   "identifier": "appfac|CYBERLAB|City of Houston",
4   "pacerSource":
5   {
6     "name": "<any name>",
7     "serverUrl": "<PACER-server Job Maanagement System URL>",
8     "security":
9     {
10      "type": "basic",
11      "username": "<username of list manager in the PACER server>",
12      "password": "<password of list manager in the PACER server>"
13    },
14      "version": "1.0.0",
15      "type": "ECR"
16    }
17  }
18 }

```

The `identifier` value for both provider and facility should be identical to HLv2 ELR's provider and facility segment.

ECR manager uses the index API to get the PACER-server endpoint to send a query request. Thus, if there are multiple providers or facilities for the ELR messages, they all need to be added to this API service with their PACER-server endpoint.

ECR-MANAGER

In `ecr-manager/` folder. Open `ecr-manager.xml` file. Example xml file is shown below.

```

1 <service>
2   <id>ecrmanager</id>
3   <name>ECR Manager</name>
4   <description>This manages ECR data from lab report and EHR data</description>
5   <env name="JAVA_HOME" value="C:\Program Files\Microsoft\jdk-17.0.2.8-hotspot\"/>
6   <env name="JDBC_DRIVER" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
7   <env name="JDBC_URL" value="jdbc:sqlserver://<host>:1433;databaseName=ecr;
8   <env name="LOCAL_BULKDATA_PATH" value="C:\workspace\PACER-client-win\ecr-manager\
9   <env name="LOCAL_PACER_SECURITY" value="Basic username:password"/>
10  <env name="LOCAL_PACER_URL" value="http://musctest.hdap.gatech.edu:8082/

```

(continues on next page)

(continued from previous page)

```
11 <JobManagementSystem/List"/>
    <env name="PACER_INDEX_SERVICE" value="http://localhost:8086/pacer-index-api/1.0.0/
12 <search"/>
13     <env name="TRUST_CERT" value="true"/>
14     <env name="SERVER_PORT" value="8085"/>
15     <executable>java</executable>
16     <arguments>-jar "%BASE%\ecr-manager.jar"</arguments>
17     <log mode="roll"></log>
</service>
```

Please make sure the environment variables are accurate.

In the `ecr-manager.xml`, `JDBC_URL` must be set to the MS-SQL database where you will be storing the PACER data. `LOCAL_*` environment variables are mostly place holders. Even though they will not be used, please set them to correct values. `LOCAL_BULKDATA_PATH` needs to be pointing to existing folders. If not, path not available error message will be shown until the folder is created.

After configuring the XML file, save it and run the following command,

```
.\ecr-manager.exe install
```

This will install the `ecr-manager` as a service. After the installation, open 'services' application (built-in app in Windows). From the list of services, locate the ECR Manager service. Right click on it and choose Properties. There, go to 'Log On' tab and choose 'this account' option. Then, add username and password. Please note that this account should have a permission to access (read and write) the MS SQL server.

Note: *Exporting cases in CSV file*

ECR-Manager has an API that will dump entire cases in csv file. The endpoint is `http(s)://<yourhost>/ecr-manager/exportCSV`. If you run it from the browser, it will save the file in the download folder with name = `csv_[datetime].csv`.

ELR-RECEIVER

In `elr-receiver/` folder, update `elr-receiver.xml` file. `ECR_URL` in the `elr-receiver.xml` is an environment variable that may need to be updated. However, if default values are used for ECR-MANAGER installation, and ECR-MANAGER and ELR-RECEIVER are running in the same machine, then the default configuration may be used without modifications.

After the XML file is configured, please save it and run the following command from the Powershell,

```
.\elr-receiver.exe install
```

This will install the `elr-receiver` as a service. After the installation, open 'services' application (built-in app in Windows). From the list of services, locate the ELR Receiver service. Right click on it and choose Properties. There, please go to 'Log On' tab and choose 'this account' option. Then, add username and password. Please note that this account should have a permission to access the local hard disk. ELR-RECEIVER needs to have read and write permission to the hard disk so that a queue file can be created and managed.

PACER-UI

This is a user dashboard that shows the case reports in PACER. The dashboard is written in Angular, and the source codes are available in <https://github.com/PACER-PHII/pacer-ui.git> if you are interested in and willing to contribute in the development.

If you just want to deploy the dashboard, please follow the instruction below.

1. In the Server Manager, Enable IIS. You may also need to configure user so that the IIS server can access the folder
2. Download the zip file from the release tag (<https://github.com/PACER-PHII/PACER-client-win/releases>)
3. Unzip the downloaded file and copy the folder named, “pacer-ui” to the place where you want to run your IIS server on.
4. Locate config.json file located in /pacer-ui/config/config.json and edit the line “api”: “<http://yellowisland01.icl.gtri.org:8085>” to your API URL. For example “api”: “<http://myapi.org:8080/>” if you will be using the UI from the same host that ECR-MANAGER is deployed.
5. Add a new application in the IIS setup and name the alias as “pacer-ui” and set the path to the “pacer-ui” folder that you created in the step 3.

Use the web browser and go to <http://localhost/pacer-ui> if you are running from the same host.

If you want to have authentication on the UI, please follow the instruction at <https://learn.microsoft.com/en-us/iis/configuration/system.webserver/security/authentication/windowsauthentication/> The IIS will ask the Windows Serverx user credential. Please note that the browser saves the credentials. So, if you are using the shared computer, please make sure you clean the web data.

Linux Server

Prerequisites

PACER-client can be installed in the Linux environment in two ways. One is using Docker, and the other is using Java and runnint them at the command line.

Docker Installation For Docker installation, the following packages must be installed.

- Docker Engine : (Unbuntu) <https://docs.docker.com/engine/install/ubuntu/> (Redhat)
- Docker Compose : <https://docs.docker.com/compose/compose-file/>

Docker downloads base images from Docker server. Therefore, incoming HTTP traffic from outside must be allowed.

PACER-client Deployment using Docker

Once the docker is installed, run the following command to install PACER-client in Docker container.

1. Go to the folder where the PACER-client is cloned or downloaded.
2. Open docker-compose.yml and check the envrionment varialbes. In most cases, the variables can be used as is. However, if you wish to change, please do for your environment. Only *ecr-postgresql* is set to restart when host restarted. If the other components need to be reatarted, please put *restart: always* in each component you want to enable the restart.
3. Run the following command,

```
sudo docker-compose up --build -d
```

4. Please check the status of each component by running,

```
sudo docker ps -a
```

Output of the command should look like follows if all components are successfully deployed. STATUS should be all “Up”.

CON-TAINER ID	IMAGE	COM-MAND	CRE-ATED	STA-TUS	PORTS	NAMES
dfa79bbed1	pacер-client-ecr-manager	“java -jar ecr-manag...”	22 seconds ago	Up 20 seconds	0.0.0.0:8085->8080/tcp, :::8085->8080/tcp	pacер-client-ecr-manager-1
e3d262a16	pacер-client-elr-receiver	“java -jar elr_recei...”	22 seconds ago	Up 21 seconds	8888/tcp, 0.0.0.0:8087->8887/tcp, :::8087->8887/tcp	pacер-client-elr-receiver-1
5262d3788	pacер-client-pacer-index-api	“java -jar /pacер-in...”	22 seconds ago	Up 21 seconds	8080/tcp, 0.0.0.0:8086->8086/tcp, :::8086->8086/tcp	pacер-client-pacer-index-api-1
1b999305e	pacер-client-ecr-postgresql	“docker-entrypoint.s...”	23 seconds ago	Up 21 seconds	0.0.0.0:5432->5432/tcp, :::5432->5432/tcp	pacер-client-ecr-postgresql-1

5. To uninstall all components, run the following command. Please note that this command will remove entire package and data.

```
sudo docker-compose down
```

2.1.3 ECR Manager

Overview

ECR Manager is the core component of PACER-client. PACER-client accepts an initial ECR from ELR Receiver and stores the ECR in the ECR repository. Then, ECR Manager creates a request in the job queue. ECR Manager periodically pulls the request from the job queue and processes it to send out to PACER-server.

PACER-server endpoint information is retrieved from [PACER Index API](#). ECR Manager is responsible for the request to be successfully made to the PACER-server. Any response other than HTTP 200 (OK) or 201 (CREATED) will be considered as failed. ECR successfully received from PACER-server will be merged into ECR repository. Any existing fields in the ECR repository will be updated. The information from PACER-server precedes information from the initial (or existing) ECR.

API Documentation

APIs for ECR Manager are as follows. These APIs should only be used within PACER-client components. No external components (from outside network) should be calling the APIs

Retrieving ECRs

GET /ecr-manager/ECR

This API is to get current ECRs. If no query parameters are provided, then the page will default to 0. First up to 50 ECRs will be returned in the response payload. Subsequent calls are required to get all ECRs. The last page has less than 50 entries.

Example ECR Request

```
GET /ecr-manager/ECR HTTP/1.1
Host: example.org:8085
Accept: */*
```

Example Response

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 07 May 2024 14:47:26 GMT

[
  {
    "Id": "4602",
    "Status": "A",
    "StatusLog": null,
    "Provider": [
      {
        "ID": {
          "value": "GT|Reliable",
          "type": "appfac"
        },
        "Name": "",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
      },
      {
        "ID": {
          "value": "P49430",
          "type": "ORDPROVIDER"
        },
        "Name": "D ATKINSON",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
      }
    ]
  }
]
```

(continues on next page)

(continued from previous page)

```

    },
    {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "John Duke",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    }
  ],
  "Facility": {
    "ID": null,
    "Name": "",
    "Phone": "",
    "Address": "",
    "Fax": "",
    "Hospital_Unit": ""
  },
  "Patient": {
    "ID": [
      {
        "value": "2000",
        "type": "urn:local:gtritest"
      },
      {
        "value": "5000000000",
        "type": "SS"
      },
      {
        "value": "82713",
        "type": "urn:local:gtritest"
      }
    ],
    "Name": {
      "given": "SOPHIE82713",
      "family": "STONE"
    },
    "Parents_Guardians": [],
    "Street_Address": "2222 Home Street, Ann Arbor MI 99999",
    "Birth_Date": "19750602",
    "Sex": "M",
    "PatientClass": "",
    "Race": {
      "Code": "",
      "System": "",
      "Display": ""
    }
  },

```

(continues on next page)

(continued from previous page)

```

    "Ethnicity": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Preferred_Language": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Occupation": "",
    "Pregnant": false,
    "Travel_History": [],
    "Insurance_Type": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Immunization_History": [],
    "Visit_DateTime": "",
    "Admission_DateTime": "",
    "Date_Of_Onset": "",
    "Symptoms": [],
    "Lab_Order_Code": [
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Laboratory_Results": [
          {
            "Code": "164200",
            "System": "L",
            "Display": "C. trachomatis - PCA",
            "Date": "Tue May 03 15:32:00 EDT 2005",
            "Value": "Positive",
            "Unit": {
              "Code": "",
              "System": "",
              "Display": ""
            }
          }
        ]
      },
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Tue May 03 15:32:00 EDT 2005",
        "Value": "Positive",
        "Unit": {
          "Code": "",
          "System": "",
          "Display": ""
        }
      }
    ],
    "Facility": {
      "ID": null,
      "Name": "",
      "Phone": "",
      "Address": "",
      "Fax": "",
      "Hospital_Unit": ""
    },
    "Provider": {

```

(continues on next page)

(continued from previous page)

```

        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        },
        "Name": "D ATKINSON",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    },
    {
        "Code": "164205",
        "System": "L",
        "Display": "N gonorrhoeae Competition Rflx",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Laboratory_Results": [
            {
                "Code": "164205",
                "System": "L",
                "Display": "N gonorrhoeae Competition Rflx",
                "Date": "Fri Apr 29 17:01:00 EDT 2005",
                "Value": "Negative",
                "Unit": {
                    "Code": "",
                    "System": "",
                    "Display": ""
                }
            },
            {
                "Code": "164212",
                "System": "L",
                "Display": "N gonorrhoeae DNA Probe w/Rflx",
                "Date": "Fri Apr 29 17:01:00 EDT 2005",
                "Value": "See Reflex",
                "Unit": {
                    "Code": "",
                    "System": "",
                    "Display": ""
                }
            }
        ],
        "Facility": {
            "ID": null,
            "Name": "",
            "Phone": "",
            "Address": "",
            "Fax": "",
            "Hospital_Unit": ""
        },
    },

```

(continues on next page)

(continued from previous page)

```

        "Provider": {
            "ID": {
                "value": "P49430",
                "type": "ORDPROVIDER"
            },
            "Name": "John Duke",
            "Phone": "",
            "Fax": "",
            "Email": "",
            "Facility": "",
            "Address": "",
            "Country": ""
        },
        "Placer_Order_Code": "",
        "Diagnosis": [],
        "Medication Provided": [],
        "Death_Date": "",
        "Date_Discharged": "",
        "Laboratory_Results": [],
        "Trigger_Code": [],
        "Lab_Tests_Performed": []
    },
    "Sending_Application": "",
    "Notes": []
}
]

```

Query Parameters

- **page** (*int*) – page number (default = 0) of ECRs. Each page contains upto 50 ECRs.
- **id** (*int*) – ECR record id
- **lastName** (*string*) – Last (or Family) name of the case patient
- **firstName** (*string*) – First (or Given) name of the case patient
- **zipCode** (*string*) – Zip code in address of the case patient
- **diagnosisCode** (*string*) – Diagnosis (or Condition) code of the case patient

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – no error

Adding an ECR

POST /ecr-manager/ECR

This API is to add an ECR. ECR must be included in the request body as a JSON format.

Example ECR Request

```
POST /ecr-manager/ECR HTTP/1.1
Host: example.org:8085
Content-Type: application/json
Accept: */*
Content-Length: 3941

{
  "Provider": [
    {
      "ID": {
        "value": "GT|Reliable",
        "type": "appfac"
      },
      "Name": "",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    },
    {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "D ATKINSON",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    },
    {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "John Duke",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    }
  ]
}
```

(continues on next page)

(continued from previous page)

```

    }
  ],
  "Facility": {
    "ID": null,
    "Name": "",
    "Phone": "",
    "Address": "",
    "Fax": "",
    "Hospital_Unit": ""
  },
  "Patient": {
    "ID": [
      {
        "value": "1234000",
        "type": "urn:local:gtritest"
      },
      {
        "value": "888770000",
        "type": "SS"
      }
    ],
    "Name": {
      "given": "Example",
      "family": "Patient"
    },
    "Parents_Guardians": [],
    "Street_Address": "1234 Olympic Street, Atlanta Ga 99999",
    "Birth_Date": "19700712",
    "Sex": "M",
    "PatientClass": "",
    "Race": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Ethnicity": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Preferred_Language": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Occupation": "",
    "Pregnant": false,
    "Travel_History": [],
    "Insurance_Type": {
      "Code": "",
      "System": "",
      "Display": ""
    }
  }

```

(continues on next page)

(continued from previous page)

```

    },
    "Immunization_History": [],
    "Visit_DateTime": "",
    "Admission_DateTime": "",
    "Date_Of_Onset": "",
    "Symptoms": [],
    "Lab_Order_Code": [
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Laboratory_Results": [
          {
            "Code": "164200",
            "System": "L",
            "Display": "C. trachomatis - PCA",
            "Date": "Tue May 03 15:32:00 EDT 2005",
            "Value": "Positive",
            "Unit": {
              "Code": "",
              "System": "",
              "Display": ""
            }
          }
        ]
      },
      {
        "Facility": {
          "ID": null,
          "Name": "",
          "Phone": "",
          "Address": "",
          "Fax": "",
          "Hospital_Unit": ""
        },
        "Provider": {
          "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
          },
          "Name": "D ATKINSON",
          "Phone": "",
          "Fax": "",
          "Email": "",
          "Facility": "",
          "Address": "",
          "Country": ""
        }
      }
    ],
    {
      "Code": "164205",
      "System": "L",
      "Display": "N gonorrhoeae Competition Rflx",

```

(continues on next page)

(continued from previous page)

```

    "Date": "Fri Apr 29 17:01:00 EDT 2005",
    "Laboratory_Results": [
      {
        "Code": "164205",
        "System": "L",
        "Display": "N gonorrhoeae Competition Rflx",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Value": "Negative",
        "Unit": {
          "Code": "",
          "System": "",
          "Display": ""
        }
      },
      {
        "Code": "164212",
        "System": "L",
        "Display": "N gonorrhoeae DNA Probe w/Rflx",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Value": "See Reflex",
        "Unit": {
          "Code": "",
          "System": "",
          "Display": ""
        }
      }
    ],
    "Facility": {
      "ID": null,
      "Name": "",
      "Phone": "",
      "Address": "",
      "Fax": "",
      "Hospital_Unit": ""
    },
    "Provider": {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "John Duke",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    }
  },
  "Placer_Order_Code": "",
  "Diagnosis": [],

```

(continues on next page)

(continued from previous page)

```
    "Medication_Provided": [],
    "Death_Date": "",
    "Date_Discharged": "",
    "Laboratory_Results": [],
    "Trigger_Code": [],
    "Lab_Tests_Performed": []
  },
  "Sending_Application": "",
  "Notes": []
}
```

Example Response

```
HTTP/1.1 201
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 07 May 2024 15:52:49 GMT

{
  "Id": "5104",
  "Status": "R",
  "StatusLog": null,
  "Provider": [
    {
      "ID": {
        "value": "GT|Reliable",
        "type": "appfac"
      },
      "Name": "",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    },
    {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "D ATKINSON",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    }
  ],
}
```

(continues on next page)

(continued from previous page)

```

    {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "John Duke",
      "Phone": "",
      "Fax": "",
      "Email": "",
      "Facility": "",
      "Address": "",
      "Country": ""
    }
  ],
  "Facility": {
    "ID": null,
    "Name": "",
    "Phone": "",
    "Address": "",
    "Fax": "",
    "Hospital_Unit": ""
  },
  "Patient": {
    "ID": [
      {
        "value": "1234000",
        "type": "urn:local:gtritest"
      },
      {
        "value": "888770000",
        "type": "SS"
      }
    ],
    "Name": {
      "given": "Example",
      "family": "Patient"
    },
    "Parents_Guardians": [],
    "Street_Address": "1234 Olympic Street, Atlanta Ga 99999",
    "Birth_Date": "19700712",
    "Sex": "M",
    "PatientClass": "",
    "Race": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Ethnicity": {
      "Code": "",
      "System": "",
      "Display": ""
    }
  },

```

(continues on next page)

(continued from previous page)

```

    "Preferred_Language": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Occupation": "",
    "Pregnant": false,
    "Travel_History": [],
    "Insurance_Type": {
      "Code": "",
      "System": "",
      "Display": ""
    },
    "Immunization_History": [],
    "Visit_DateTime": "",
    "Admission_DateTime": "",
    "Date_Of_Onset": "",
    "Symptoms": [],
    "Lab_Order_Code": [
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Fri Apr 29 17:01:00 EDT 2005",
        "Laboratory_Results": [
          {
            "Code": "164200",
            "System": "L",
            "Display": "C. trachomatis - PCA",
            "Date": "Tue May 03 15:32:00 EDT 2005",
            "Value": "Positive",
            "Unit": {
              "Code": "",
              "System": "",
              "Display": ""
            }
          }
        ]
      },
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Tue May 03 15:32:00 EDT 2005",
        "Value": "Positive",
        "Unit": {
          "Code": "",
          "System": "",
          "Display": ""
        }
      }
    ],
    "Facility": {
      "ID": null,
      "Name": "",
      "Phone": "",
      "Address": "",
      "Fax": "",
      "Hospital_Unit": ""
    },
    "Provider": {
      "ID": {
        "value": "P49430",
        "type": "ORDPROVIDER"
      },
      "Name": "D ATKINSON",

```

(continues on next page)

(continued from previous page)

```

        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    }
},
{
    "Code": "164205",
    "System": "L",
    "Display": "N gonorrhoeae Competition Rflx",
    "Date": "Fri Apr 29 17:01:00 EDT 2005",
    "Laboratory_Results": [
        {
            "Code": "164205",
            "System": "L",
            "Display": "N gonorrhoeae Competition Rflx",
            "Date": "Fri Apr 29 17:01:00 EDT 2005",
            "Value": "Negative",
            "Unit": {
                "Code": "",
                "System": "",
                "Display": ""
            }
        },
        {
            "Code": "164212",
            "System": "L",
            "Display": "N gonorrhoeae DNA Probe w/Rflx",
            "Date": "Fri Apr 29 17:01:00 EDT 2005",
            "Value": "See Reflex",
            "Unit": {
                "Code": "",
                "System": "",
                "Display": ""
            }
        }
    ],
    "Facility": {
        "ID": null,
        "Name": "",
        "Phone": "",
        "Address": "",
        "Fax": "",
        "Hospital_Unit": ""
    },
    "Provider": {
        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        }
    },

```

(continues on next page)

(continued from previous page)

```

        "Name": "John Duke",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    }
}
],
"Placer_Order_Code": "",
"Diagnosis": [],
"Medication Provided": [],
"Death_Date": "",
"Date_Discharged": "",
"Laboratory_Results": [],
"Trigger_Code": [],
"Lab_Tests_Performed": []
},
"Sending Application": "",
"Notes": []
}

```

Query Parameters

- **source** (*string*) – Source of ECR. Default is “elr”. If the source is EHR, then use “ehr”.

Status Codes

- 201 Created – Created

Triggering manual request

POST /ecr-manager/trigger

Request to PACER-server is triggered by ELR in the PACER-client workflow (see [Client Workflow](#)). However, in case that the manual triggering is needed, this API can be used to trigger an individual case.

Example Trigger Request

```

POST /ecr-manager/trigger?id=402 HTTP/1.1
Host: localhost:8085
Accept: */*
Content-Length: 0

```

Example Trigger Response

```

HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Length: 0
Date: Wed, 12 Jul 2023 04:16:19 GMT

```

Query Parameters

- **id** (*int*) – ECR record ID. This is required query parameter.

Status Codes

- 200 OK – Ok

Retrieving Incoming ECR History**GET /ecr-manager/ECRhistory**

This API retrieves list of incoming ECRs. ECRs coming from both ELR and EHR. Using this API, history of incoming ECRs can be obtained.

Example ECR History Request

```
GET /ecr-manager/ECRhistory?id=4603&source=elr HTTP/1.1
Host: example.org:8085
Accept: */*
```

Example ECR History Response

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: application/json
Transfer-Encoding: chunked
Date: Tue, 07 May 2024 16:34:58 GMT

[
  {
    "ecrId": 4603,
    "date": "2024-04-02 10:24:52.939",
    "source": "elr",
    "data": {
      "Id": "4603",
      "Status": "R",
      "StatusLog": null,
      "Provider": [
        {
          "ID": {
            "value": "GT|Reliable",
            "type": "appfac"
          },
          "Name": "",
          "Phone": "",
          "Fax": "",
          "Email": "",
          "Facility": "",
          "Address": "",
          "Country": ""
        }
      ]
    }
  }
]
```

(continues on next page)

(continued from previous page)

```

        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        },
        "Name": "D ATKINSON",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    },
    {
        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        },
        "Name": "John Duke",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    }
],
"Facility": {
    "ID": null,
    "Name": "",
    "Phone": "",
    "Address": "",
    "Fax": "",
    "Hospital_Unit": ""
},
"Patient": {
    "ID": [
        {
            "value": "2001",
            "type": "urn:local:gtritest"
        },
        {
            "value": "5000000001",
            "type": "SS"
        }
    ],
    "Name": {
        "given": "SOPHIE2001",
        "family": "STONE"
    },
    "Parents_Guardians": [],
    "Street_Address": "2222 Home Street, Ann Arbor MI 99999",
    "Birth_Date": "19750602",

```

(continues on next page)

(continued from previous page)

```

"Sex": "M",
"PatientClass": "",
"Race": {
  "Code": "2106-3",
  "System": "CDCREC",
  "Display": "White"
},
"Ethnicity": {
  "Code": "N",
  "System": "HL70189",
  "Display": "Not Hispanic or Latino"
},
"Preferred_Language": {
  "Code": "eng",
  "System": "ISO6392",
  "Display": "English"
},
"Occupation": "",
"Pregnant": false,
"Travel_History": [],
"Insurance_Type": {
  "Code": "",
  "System": "",
  "Display": ""
},
"Immunization_History": [],
"Visit_DateTime": "",
"Admission_DateTime": "",
"Date_Of_Onset": "",
"Symptoms": [],
"Lab_Order_Code": [
  {
    "Code": "164200",
    "System": "L",
    "Display": "C. trachomatis - PCA",
    "Date": "Fri Apr 29 17:01:00 EDT 2005",
    "Laboratory_Results": [
      {
        "Code": "164200",
        "System": "L",
        "Display": "C. trachomatis - PCA",
        "Date": "Tue May 03 15:32:00 EDT 2005",
        "Value": "Positive",
        "Unit": {
          "Code": "",
          "System": "",
          "Display": ""
        }
      }
    ]
  }
],
"Facility": {
  "ID": null,

```

(continues on next page)

(continued from previous page)

```

        "Name": "",
        "Phone": "",
        "Address": "",
        "Fax": "",
        "Hospital_Unit": ""
    },
    "Provider": {
        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        },
        "Name": "D ATKINSON",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    }
},
{
    "Code": "164205",
    "System": "L",
    "Display": "N gonorrhoeae Competition Rflx",
    "Date": "Fri Apr 29 17:01:00 EDT 2005",
    "Laboratory_Results": [
        {
            "Code": "164205",
            "System": "L",
            "Display": "N gonorrhoeae Competition Rflx",
            "Date": "Fri Apr 29 17:01:00 EDT 2005",
            "Value": "Negative",
            "Unit": {
                "Code": "",
                "System": "",
                "Display": ""
            }
        },
        {
            "Code": "164212",
            "System": "L",
            "Display": "N gonorrhoeae DNA Probe w/Rflx",
            "Date": "Fri Apr 29 17:01:00 EDT 2005",
            "Value": "See Reflex",
            "Unit": {
                "Code": "",
                "System": "",
                "Display": ""
            }
        }
    ],
    "Facility": {

```

(continues on next page)

(continued from previous page)

```

        "ID": null,
        "Name": "",
        "Phone": "",
        "Address": "",
        "Fax": "",
        "Hospital_Unit": ""
    },
    "Provider": {
        "ID": {
            "value": "P49430",
            "type": "ORDPROVIDER"
        },
        "Name": "John Duke",
        "Phone": "",
        "Fax": "",
        "Email": "",
        "Facility": "",
        "Address": "",
        "Country": ""
    }
},
"Placer_Order_Code": "",
"Diagnosis": [],
"Medication_Provided": [],
"Death_Date": "",
"Date_Discharged": "",
"Laboratory_Results": [],
"Trigger_Code": [],
"Lab_Tests_Performed": []
},
"Sending_Application": "GT 1234 CLIA",
"Notes": []
}
]

```

Query Parameters

- **id** (*int*) – ECR record ID. Default = -1 (Id <= 0 indicates all records)
- **source** (*string*) – Source of ECR. “elr” or “ehr”
- **date** (*string*) – date with inequality of “eq”, “lt”, “le”, “gt”, and “ge”. Example “le 2022/12/01” means “<= 2022/12/01”

Status Codes

- 200 OK – Ok

Export ECRs to CSV File

GET /ecr-manager/ExportCSV

This API exports all ECRs to CSV file.

Example ECR Export Request

```
GET /ecr-manager/exportCSV/ HTTP/1.1
Host: yellowisland01.icl.gtri.org:8085
Accept: */*
```

Example ECR Export Response

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Disposition: attachment; filename=ecr_20240507124728.csv
Content-Type: text/csv
Transfer-Encoding: chunked
Date: Tue, 07 May 2024 16:47:28 GMT

[ecr_20240507124728.csv is attached]
```

CSV file is attached and can be downloaded.

Version Information

GET /ecr-manager/version

This API displays the version of ECR-MANAGER.

Example ECR-MANAGER Version Request

```
GET /ecr-manager/version HTTP/1.1
Host: yellowisland01.icl.gtri.org:8085
User-Agent: insomnia/9.1.0
Accept: */*
```

Example ECR-MANAGER Version Response

```
HTTP/1.1 200
Vary: Origin
Vary: Access-Control-Request-Method
Vary: Access-Control-Request-Headers
Content-Type: text/plain; charset=UTF-8
Content-Length: 5
Date: Tue, 07 May 2024 21:50:58 GMT

1.5.3
```

2.1.4 ELR Reciever

Overview

ELR Receiver is a starting point for PACER-client workflow (see [Client Workflow](#)) and expects the electronic lab reports (ELRs) in HL7 v2 with a message type, ORU^R01

It supports [MLLP](#) for the v2 messaging protocol.

During the operation, ELR Receiver parses the incoming ELR and converts it to electronic case report (ECR). Currently, the parser supports HL7v2.5.1 and HL7v2.3.1. The ECR converted the ELR message is submitted to the ECR-MANAGER API (see [Adding an ECR](#)). This ECR is called, “initial ECR”, in [ECR Manager](#) and stored in the ECR database.

ELR Receiver implements automatic failure recovery using a queue. ECR submission is queued and only removed from the queue when the submission is successfully completed. If the ECR submission to [ECR Manager](#) MANAGER fails, the failed submission will be retrieved from the queue and resubmitted. This process repeats until the submission succeeds.

2.1.5 PACER Index API

Overview

PACER Index Service provides the PACER-server endpoint information. ECR Manager uses PACER Index Service API to search for a PACER-server endpoint and authorization method using organization and/or provider in the initial ECR.

The PACER Index Service is managed by the health department (or organization where the PACER-client platform is deployed). Index API itself does not require authorization. However, the management APIs require Basic authorization. The Basic authorization is set using environment variables during deployment (see [Installation and Configuration](#)). The API Documentation section below explains how to manage the indexing and search the PACER-server endpoint.

API Documentation

PACER-Index-API service has two operations. One is management operation, and the other one is search operation.

Manament Operation

GET /pacer-index-api/1.0.0/manage/{id}

This API retrieves entire index entries.

Example PACER-INDEX-API Entry Request

```
GET /pacer-index-api/1.0.0/manage HTTP/1.1
Host: example.org:8086
Authorization: Basic encoDedcReDentiAl==
Accept: */*
```

Example PACER-INDEX-API Entry Response

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
```

(continues on next page)

(continued from previous page)

```

Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 08 May 2024 03:35:51 GMT

{
  "count": 2,
  "created": "2024-05-07T23:35:51.128-04:00",
  "list": [
    {
      "id": 1,
      "providerName": "John Duke",
      "identifier": "ORDPROVIDER|P49430",
      "pacerSource": {
        "name": "PACER test",
        "serverUrl": "https://pacer-server.org/JobManagementSystem/List",
        "security": {
          "type": "basic",
          "username": "username",
          "password": "password"
        },
        "version": "1.0.0",
        "type": "ECR"
      }
    },
    {
      "id": 2,
      "providerName": "",
      "identifier": "appfac|PACERv1|GTRI",
      "pacerSource": {
        "name": "PACER test",
        "serverUrl": "https://pacer-server.org/JobManagementSystem/List",
        "security": {
          "type": "basic",
          "username": "username",
          "password": "password"
        },
        "version": "1.0.0",
        "type": "ECR"
      }
    }
  ]
}

```

Query Parameters

- **id** (*int*) – PACER-server Index ID. Optional. If {id} is provided, the response will be single entry.

Request Headers

- [Authorization](#) – Basic Authorization

Status Codes

- [200 OK](#) – Ok
- [401 Unauthorized](#) – Authentication information is missing or invalid
- [403 Forbidden](#) – Forbidden
- [404 Not Found](#) – Not Found

POST /pacer-index-api/1.0.0/manage

This API creates a new entry with the entry provided in the payload.

Example PACER-INDEX-API Entry Create Request

```
POST /pacer-index-api/1.0.0/manage HTTP/1.1
Host: yellowisland01.icl.gtri.org:8086
User-Agent: insomnia/9.1.0
Content-Type: application/json
Authorization: Basic dXNlcjpwYXNzd29yZA==
Accept: */*
Content-Length: 318

{
  "providerName": "Test PROVIDER",
  "identifier": "TEST_PROVIDER|1",
  "pacerSource": {
    "name": "TEST_PACER_1",
    "serverUrl": "https://pacer-server.org/JobManagementSystem/List",
    "security": {
      "type": "basic",
      "username": "username",
      "password": "password"
    },
    "version": "1.4.3",
    "type": "ECR"
  }
}
```

Example PACER-INDEX-API Entry Create Response

```
HTTP/1.1 201
Location: http://example.org:8086/pacer-index-api/1.0.0/manage/5
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 0
Date: Wed, 08 May 2024 05:26:49 GMT
```

Request Headers

- [Authorization](#) – Basic Authorization

Status Codes

- 201 Created – Created
- 204 No Content – Content Not Found
- 400 Bad Request – Invalid input, object invalid
- 401 Unauthorized – Authentication information is missing or invalid
- 403 Forbidden – Forbidden
- 404 Not Found – Not Found
- 409 Conflict – Entry already exist

PUT /pacer-index-api/1.0.0/manage/{id}

This API updates the existing entry with the new entry provided in the payload.

Example PACER-INDEX-API Entry Update Request

```
PUT /pacer-index-api/1.0.0/manage/1 HTTP/1.1
Host: example.org:8086
Content-Type: application/json
Authorization: Basic dXNlcjpwYXNzd29yZA==
Accept: */*
Content-Length: 375

{
  "id": 1,
  "providerName": "LOCAL PROVIDER",
  "identifier": "LOCAL_PROVIDER|1",
  "pacerSource": {
    "name": "LOCAL_PACER_1",
    "serverUrl": "https://pacer-server.org/JobManagementSystem/List",
    "security": {
      "type": "basic",
      "username": "username",
      "password": "password"
    },
    "version": "1.4.3",
    "type": "ECR"
  }
}
```

Example PACER-INDEX-API Entry Update Response

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 0
Date: Wed, 08 May 2024 05:14:52 GMT
```

Query Parameters

- **id** (*int*) – PACER-INDEX-API ID. Required

Request Headers

- **Authorization** – Basic Authorization

Status Codes

- **200 OK** – Ok
- **204 No Content** – Content Not Found
- **400 Bad Request** – Invalid input, object invalid
- **401 Unauthorized** – Authentication information is missing or invalid
- **403 Forbidden** – Forbidden
- **404 Not Found** – Not Found

DELETE /pacer-index-api/1.0.0/manage/{id}

This API deletes the entry with id = {id}

Example PACER-INDEX-API Entry Delete Request

```
DELETE /pacer-index-api/1.0.0/manage/5 HTTP/1.1
Host: example.org:8086
Content-Type: application/json
Authorization: Basic dXNlcjpwYXNzd29yZA==
Accept: */*
Content-Length: 0
```

Example PACER-INDEX-API Entry Delete Response

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Length: 0
Date: Wed, 08 May 2024 05:35:41 GMT
```

Request Headers

- **Authorization** – Basic Authorization

Status Codes

- **200 OK** – Ok
- **204 No Content** – Content Not Found
- **400 Bad Request** – Invalid input, object invalid
- **401 Unauthorized** – Authentication information is missing or invalid
- **403 Forbidden** – Forbidden

Search Operation

GET /pacer-index-api/1.0.0/search

This API search the PACER-server endpoint with two search parameters, `provider-name` or `organization-id`. `provider-name` is compared with `providerName` in the entry. `organization-id` is compared with `identifier` in the entry.

`organization-id` and `provider-name` are retrieved from two different places in HL7v2. First, `organization-id` is formatted as `type|id`. if `type` is set to "appfac", then it means sending application and sending facility in MSH segment of HL7 v2 message. Sending application is in MSH-3. And, sending facility is in MSH-4. `id` is taking value from MSH-3.1 and MSH-4.1. These are namespace ids in the segment.

`provider-name` is from OBR and ORC segments where ordering provider information is specified. Each segment has a list of ordering providers (OBR-16 or ORC-12). If the ordering provider in OBR has an id number populated, then `type|id` will be overwritten with this value. In this case, the type is set to "ORDPROVIDER". OBR is first checked for the provider id number. If this is empty, then ORC is used. If both are empty, the `type|id` value remains as MSH's sending application and sending facility. So, based on the type, we can locate where the identifier of the organization is obtained.

The purpose of using MSH is to support the HIE network, where the ordering entities are defined by application/facility not ordering provider. When registering PACER-server at the PACER Index API Service, the identifier must be set correctly. Otherwise, ECR manager won't be able to retrieve a correct end point for the PACER server. If unsure whether MSH or OBR/ORC will be used, two entries can be registered with the same PACER server information so that it can be matched.

Example PACER-INDEX-API Search Request

```
GET /pacer-index-api/1.0.0/search?organization-id=LOCAL_PROVIDER%7C1 HTTP/1.1
Host: example.org:8086
Accept: */*
```

Example PACER-INDEX-API Search Response

```
HTTP/1.1 200
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Content-Type: application/json
Transfer-Encoding: chunked
Date: Wed, 08 May 2024 06:00:02 GMT
```

```
{
  "count": 1,
  "created": "2024-05-08T02:00:02.655-04:00",
  "list": [
    {
      "id": 1,
      "providerName": "LOCAL PROVIDER",
      "identifier": "LOCAL_PROVIDER|1",
      "pacerSource": {
        "name": "LOCAL_PACER_1",
        "serverUrl": "https://pacer-server.org/JobManagementSystem/List",
```

(continues on next page)

(continued from previous page)

```

        "security": {
            "type": "basic",
            "username": "username",
            "password": "password"
        },
        "version": "1.4.3",
        "type": "ECR"
    }
}
]
}

```

Query Parameters

- **provider-name** (*string*) – Provider Name
- **organization-id** (*string*) – Organization or facility Type:ID

Status Codes

- 200 OK – Ok
- 400 Bad Request – Invalid input, object invalid
- 403 Forbidden – Forbidden

2.1.6 PACER UI

Overview

<PACER UI: Overview>

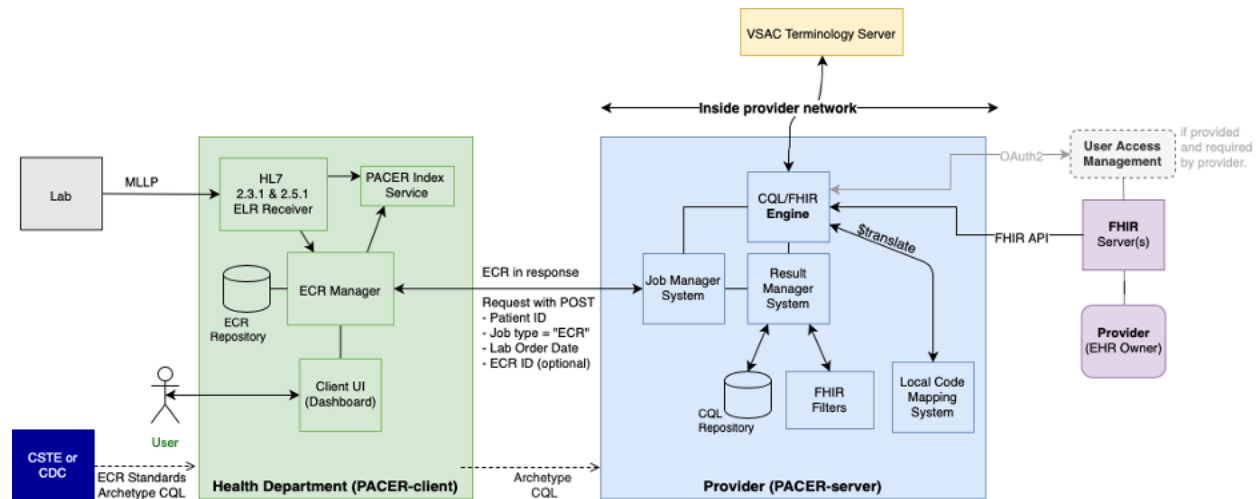
Walkthrough

<PACER UI User Walkthrough with pictures here>

2.2 PACER Server

This is the documentation for the server components of PACER. The server is responsible for managing a request from the PACER client, maintaining the query strategy and logic for interfacing to the local FHIR server, and responding with the collected ECR format. On installation; the server is pointed towards a singular endpoint for a FHIR interface. Using Clinical Quality Language(CQL) scripts and some custom handling, PACER retrieves related FHIR resources, and maps to the local ECR json format.

PACER consists of: a JobManagementSystem for handling requests, a ResultsManager for handling FHIR results, a CQL Execution Service for executing CQL, a CQLStorage service for maintaining CQL scripts, a “FHIR Filter” service for redacting sensitive data, and a Translation Service for translating local codesets into standardized codes.



2.2.1 Server Installation And Deploy Instructions

PACER is a dockerized-project. It is preferable to build each component as a docker image and deploy with the Docker-Compose tool. For more information refer to

- Docker-CE (Community Edition). Refer to <https://docs.docker.com/install/> for installation instructions
- Docker-Compose, version 1.3 or later. Refer to <https://docs.docker.com/compose/install/> for installation instructions.

Pacer consists of 2 separate docker-compose files: `docker-compose-db.yml` and `docker-compose-apps.yml`. * `docker-compose-db.yml` contains the configuration information for the database, and should be built and ran first before the `docker-compose-apps.yml` script. * `docker-compose-apps.yml` consist of the application components that manage the PACER workflow.

To build and deploy the compose project follow these steps from the command line. 1. Review and confirm configuration of the yaml files before installation. Refer to [configuration below](#) for more detailed instructions

2. Build the database and application projects

```
docker-compose -f docker-compose-db.yml build
docker-compose -f docker-compose-apps.yml build
```

This will build deployable images for 6 different containers.

- `cql-storage`
- `cql-execution`
- `db`
- `fhir-filter`
- `job-management-system`
- `results-manager`

3. Deploy dockerfiles in stages using compose.

Note: Again before deploying, it is recommended to refer to [server configuration below](#) to ensure connection to the FHIR server is correct

```
docker-compose -f docker-compose-db.yml up -d #wait for the database service to complete
↪it's initdb.script
docker-compose -f docker-compose-apps.yml up -d
```

This will turn on all application components and enable networking for the PACER application to receive requests.

2.2.2 Configuration

Before the server can be deployed, configuration within the `docker-compose-apps.yml` file must be reviewed. The `yml` file describes configuration for each of the above components. Configuration is passed as environment variables into each container.

Configuration Requirements

- `results-manager/CQL_EXECUTION_DATA_SERVICE` This is a defined baseurl to the hosted fhir server to be used. FHIR url paths should end in `/fhir/` as the baseurl.
- `results-manager/CQL_EXECUTION_DATA_USER` If the FHIR server uses [Basic Authentication](#) for authentication; provide a username here.
- `results-manager/CQL_EXECUTION_DATA_PASS` If the FHIR server uses [Basic Authentication](#) for authentication; provide a password here.
- `results-manager/CQL_EXECUTION_EPIC_CLIENT_ID` If the FHIR server uses [Bearer Token Authentication](#) generate a service token for PACER and apply the plaintext token string here.
- `results-manager/CQL_EXECUTION_TERMINOLOGY_SERVICE` If a terminology service is provided via FHIR, provide the service url. FHIR url paths should end in `/fhir/` as the baseurl.
- `results-manager/CQL_EXECUTION_TERMINOLOGY_USER` If the terminology server uses [Basic Authentication](#) for authentication; provide a username here.
- `results-manager/CQL_EXECUTION_TERMINOLOGY_PASS` If the terminology service uses [Basic Authentication](#) for authentication; provide a password here.
- `results-manager/CQL_EXECUTION_CODEMAPPER_SYSTEMS_MAP` If the FHIR server uses local system definitions for common terminology systems (LOINC, Snomed, RxNorm), provide a key:value pair for the general url to the local system url provided. An example is provided in the comments for creating this key:value pair.

2.2.3 External FHIR Server

The external FHIR server is the system managed FHIR server. In addition to read capabilities in each resource, It is expected to support these specific search capabilities

FHIR Server Requirements

- **Patient**
 - ?identifier Identifier search parameter is used to identify the patient.
- **Condition**
 - ?patient Patient id related to the Condition
 - ?category Code search with supported condition category types
 - ?code:in Code search param used with :in modifier to provide large concept sets to be searched.
- **Encounter**
 - ?patient Patient id related to the Encounter
- **Immunization**
 - ?patient Patient id related to the Immunization
- **Medication**
 - ?code:in Code search param used with :in modifier to provide large concept sets to be searched.
- **MedicationRequest**
 - ?patient Patient id related to the MedicationRequest
 - code:in Code search param used with :in modifier to provide large concept sets to be searched.
- **Observation**
 - ?patient Patient id related to the Observation
 - ?category Code search with supported observation category types
 - code:in Code search param used with :in modifier to provide large concept sets to be searched.

Warning: In cases where MedicationRequest search code is not available, a Medication Reference must be provided within the response json; in order to retrieve related medication data.

2.2.4 Job Management System

Job Management System: Overview

<ECR Manager Information here>

Job Management System: API Documentation

<ECR Manager API Review here>

2.2.5 Results Manager

Results Manager: Overview

<Results Manager Information here>

Results Manager: API Documentation

<Results Manager API Review here>

2.2.6 CQL Storage

CQL Storage: Overview

<CQL Storage Information here>

CQL Storage: API Documentation

<CQL Storage API Review here>

2.2.7 CQL Execution Service

CQL Execution Service: Overview

<CQL Execution Service here>

CQL Execution Service: API Documentation

<CQL Execution Service API Review here>

2.2.8 FHIR Filter

FHIR Filter: Overview

<FHIR Filter here>

FHIR Filter: API Documentation

<FHIR Filter API Review here>

2.2.9 Translate Concept Service

Translate Concept Service: Overview

<Translate Concept Service here>

Translate Concept Service: API Documentation

<Translate Concept Service API Review here>

HTTP ROUTING TABLE

/ecr-manager

GET /ecr-manager/ECR, 13
GET /ecr-manager/ECRhistory, 27
GET /ecr-manager/ExportCSV, 32
GET /ecr-manager/version, 32
POST /ecr-manager/ECR, 18
POST /ecr-manager/trigger, 26

/pacer-index-api

GET /pacer-index-api/1.0.0/manage/{id}, 33
GET /pacer-index-api/1.0.0/search, 38
POST /pacer-index-api/1.0.0/manage, 35
PUT /pacer-index-api/1.0.0/manage/{id}, 36
DELETE /pacer-index-api/1.0.0/manage/{id}, 37