PhilHealth

Implementation Guide

Date 20240604
Version 3.0

APPROVAL SHEET

This document along with its attachments have been approved as the Official Implementation Guide for the Electronic Claims (eClaims) Project.

Following approval of this document, the documents will be distributed to Health Facilities and Health Information System / Electronic Medical Record (HIS/EMR) Service Providers for compliance to the said project

Prepared by:

(Sgd.) SENECA L. BELCHEZ
System Analyst

(Sgd.) MANUEL S. MANLANGIT
System Developer

(Sgd.) ARTHUR M. MARAMAG
System Developer

Recommended for Release by:

(Sgd.) JOCELYN S. PABLO
OIC-Division Chief
Information Systems Management Division
IT Management Department

Approved by:

(Sgd.) NELSON S. DE VERA
Acting Senior Manager
IT Management Department

(Sgd.) JOVITA V. ARAGONA
Senior Vice President/Chief Information Officer
Office of the Senior Vice President Chief Information Officer, Information Management Sector

Amendment History

Revision Number	Remarks
20240215	Initial Release for the SearchCaseRate, GenerateToken, ValidateEsoa and ValidateCF5 methods
20240216	Changed the parameter of SearchCaseRate method from object input parameters to comma separated list of input parameters
20240228	Added the requestQrAuthorization and inquireQrTrackingNo methods for the implementation of a use case for QR code of the eGov super app in the PhilHealth claims processing
20240306	Added the uploadeClaims method; Added Annex B, C, D, E, F
20240418	Added softwareCertId in the header of uploadeClaims method
20240423	Added the following methods: addRequiredDocument, eClaimsFileCheck, getClaimStatus, getDoctorPAN, getMemberPIN, getUploadedClaimsMap, getVoucherDetails, isDoctorAccredited, searchEmployer, getDBServerDateTime, getServerDateTime, getServerVersion and generatePBEFPDF

Table of Contents

INTRODUCTION

The Philippine Health Insurance Corporation (PhilHealth) has been implementing the Electronic Claims (eClaims) System since 2016 for the All Case Rates (ACR) provider payment mechanism. This system was developed to automate the claims process, enabling improved turnaround times for claims processing and reimbursement of providers.

The Universal Health Care Act of 2019 aims to address these issues by mandating PhilHealth to shift to paying providers prospectively using Diagnosis-Related Groups (DRG). A key component of this reform is an enhanced version of the current eClaims system which accommodates the collection of data required for DRG grouping.

This guide is designed to provide in-house hospital developers and service providers essential insights and instructions on how to successfully navigate the electronic claims submission process within the context of the PhilHealth DRG system.

Disclaimer:

All information and content in this material is provided in good faith by Philippine Health Insurance Corporation and is based on sources believed to be reliable and accurate at the time of development. The PhilHealth and their respective officers, employees and supervisors, do not accept legal liability or responsibility for the Material, or any consequences arising from its use.

Philippine Health Insurance Corporation (PhilHealth) is committed to ensuring ease of availability at the point of care for all its beneficiaries.

APPLICATION PROGRAM INTERFACE (API)

The Application Programming Interface (API) of the PhilHealth e-Claims Web Service (PECWS) serves as the primary interface for interactions with PhilHealth's electronic claims processing system. This existing facade API is the gateway through which healthcare institutions and service providers can connect to and communicate with PhilHealth's infrastructure for the purpose of submitting electronic claims.

It acts as the bridge, allowing these entities to send claim-related data and information to PhilHealth, enabling the processing and evaluation of healthcare claims. Through the PECWS API, users can submit data regarding patient information, medical procedures, diagnoses, and other relevant details necessary for the billing and reimbursement of healthcare services.

WEB SERVICE METHODS

1. Get Token Method

This method will generate a token as authorization key to access API methods using the encrypted Accreditation Number and Sofware Certification ID of the health facility.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getToken`

Method

POST

Inputs

Header

Key	Value/Remarks
`accreditationNo`	The PhilHealth Accreditation Number (PAN) of the Health Facility
`softwareCertificateId`	The software certification ID of the Health Facility

Output

JSON object containing the following key-value pairs:

Key	Value/Remarks
`success`	A value of 'true' that data has been retrieved successfully
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	The generated Token

Sample Output Payload

```
{
  "message":"",
  "result":"eyJhbGciOiJIUzI1Ni……    ",
  "success":true
}
```

2. Validate eSOA Method

A method that will validate encrypted eSOA XML format against Document Type Definition (DTD), data format and valid values set by PhilHealth.

Important Note:
After successful validation, eSOA XML should be encrypted using PhilHealth Public Key and submitted as attachment to electronic claims using the **EclaimsUpload Method.**

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/validateeSOA`

Method

POST

Inputs

Header

Key	Values
`token`	PECWS authentication token

Body

The eSOA XML must be encrypted using the cipher key issued by PhilHealth to the Health Facility. The encrypted text contains the following key-value pairs.

Key	Value/Remarks
`docMimeType`	`"text/xml"`
`hash`	The computed hash value of the unencrypted text.
`key1`	“” (Empty string)
`key2`	“” (Empty string)
`iv`	The initialization vector used in the AES encryption
`doc`	The result of the encryption of eSOA XML text using the cipher key of the HF

Sample Input Payload

```
{
   "docMimeType": "text/xml",
   "hash": "dfe701c0c9bbca0678300540591fac928f7dc8f4d74d9771a3841317e7a99aec",
   "key1": "",
   "key2": "",
   "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
   "doc": "PMs1FWFZT+odAp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geSEN=="
}
```

Sample eSOA XML

```
<eSOA
    pHciPan=""
    pHciTransmittalId="">
    <SummaryOfFees>
        <RoomAndBoard>
            <SummaryOfFee
                pActualCharges=""
                pSeniorCitizenDiscount=""
                pPWDDiscount=""
                pPCSO=""
                pDSWD=""
                pDOHMAP="0"
                pHMO="0"
            />
            <OtherFundSource
                pDescription="Henry Sy Foundation"
                pAmount="3000.00"
            />
        </RoomAndBoard>
        <DrugsAndMedicine>
            <SummaryOfFee
                pActualCharges="3500"
                pSeniorCitizenDiscount="0"
                pPWDDiscount="0"
                pPCSO="0"
                pDSWD="1000.00"
                pDOHMAP="0"
                pHMO="0"
            />
        </DrugsAndMedicine>
        <LaboratoryAndDiagnostic>
            <SummaryOfFee
                pActualCharges="4000.00"
                pSeniorCitizenDiscount="0"
                pPWDDiscount="0"
                pPCSO="0"
                pDSWD="0"
                pDOHMAP="0"
                pHMO="0"
            />
        </LaboratoryAndDiagnostic>
        <OperatingRoomFees>
            <SummaryOfFee
                pActualCharges="12000.00"
                pSeniorCitizenDiscount="0"
                pPWDDiscount="0"
                pPCSO="0"
                pDSWD="0"
                pDOHMAP="0"
                pHMO="0"
            />
        </OperatingRoomFees>
        <MedicalSupplies>
            <SummaryOfFee
                pActualCharges="2000.00"
                pSeniorCitizenDiscount="1000.00"
                pPWDDiscount="0"
                pPCSO="0"
                pDSWD="0"
                pDOHMAP="0"
                pHMO="0"
            />
```

```
    </MedicalSupplies>
    <PhilHealth
        pTotalCaseRateAmount="15000.00"
    />
    <Balance
        pAmount="3500.00"
    />
</SummaryOfFees>
<ProfessionalFees>
    <ProfessionalFee>
        <ProfessionalInfo
            pPAN="1234-4567890-1"
            pFirstName="JUAN"
            pMiddleName="TAMAD"
            pLastName="DELA CRUZ"
            pSuffixName=""
        />
        <SummaryOfFee
            pActualCharges="0"
            pSeniorCitizenDiscount="0"
            pPWDDiscount="0"
            pPCSO="0"
            pDSWD="0"
            pDOHMAP="0"
            pHMO="0"
        />
    </ProfessionalFee>
    <PhilHealth
        pTotalCaseRateAmount="2000.00"
    />
    <Balance
        pAmount="27180.00"
    />
</ProfessionalFees>
<ItemizedBillingItems>
    <ItemizedBillingItem
        pServiceDate="06-30-2021"
        pItemCode="DM-1353"
        pItemName="Paracetamol"
        pUnitOfMeasurement="Box"
        pUnitPrice="700.00"
        pQuantity="5"
        pTotalAmount="3500.00"
    />
    <ItemizedBillingItem
        pServiceDate="06-30-2021"
        pItemCode="LD-13434"
        pItemName="CT Scan"
        pUnitOfMeasurement="Box"
        pUnitPrice="500.00"
        pQuantity="3"
        pTotalAmount="4000.00"
    />
    <ItemizedBillingItem
        pServiceDate="06-30-2021"
        pItemCode="OR-13452"
        pItemName="Operating Room"
        pUnitOfMeasurement="Item"
        pUnitPrice="12000"
        pQuantity="1"
        pTotalAmount="12000"
    />
    <ItemizedBillingItem
        pServiceDate="06-30-2021"
        pItemCode="MS-2521"
```

```
            pItemName="Gloves"
            pUnitOfMeasurement="Box"
            pUnitPrice="500.00"
            pQuantity="1"
            pTotalAmount="500"
        />
        <ItemizedBillingItem
            pServiceDate="06-30-2021"
            pItemCode="MS-2522"
            pItemName="Face Mask"
            pUnitOfMeasurement="Box"
            pUnitPrice="500.00"
            pQuantity="3"
            pTotalAmount="1500.00"
        />
    </ItemizedBillingItems>
</eSOA>
```

eSOA XML DTD

```
<?xml version="1.0" encoding="utf-8"?>

<!ELEMENT eSOA (SummaryOfFees, ProfessionalFees, ItemizedBillingItems)>

<!ATTLIST eSOA
    pHciPan CDATA #REQUIRED
    pHciTransmittalId CDATA #REQUIRED>

<!ELEMENT    SummaryOfFees    (RoomAndBoard,    DrugsAndMedicine,    LaboratoryAndDiagnostic,
OperatingRoomFees, MedicalSupplies, PhilHealth, Balance)>

<!ELEMENT ProfessionalFees (ProfessionalFee*, PhilHealth, Balance)>

<!ELEMENT ItemizedBillingItems (ItemizedBillingItem+)>

<!ELEMENT PhilHealth EMPTY>
<!ATTLIST PhilHealth
    pTotalCaseRateAmount CDATA #REQUIRED
>

<!ELEMENT Balance EMPTY>
<!ATTLIST Balance
    pAmount CDATA #REQUIRED
>


<!ELEMENT OtherFundSource EMPTY>
<!ATTLIST OtherFundSource
    pDescription CDATA #REQUIRED
    pAmount CDATA #REQUIRED>

<!ELEMENT SummaryOfFee EMPTY>
<!ATTLIST SummaryOfFee
    pChargesNetOfApplicableVat CDATA #REQUIRED
    pSeniorCitizenDiscount CDATA #REQUIRED
    pPWDDiscount CDATA #REQUIRED
    pPCSO CDATA #REQUIRED
    pDSWD CDATA #REQUIRED
    pDOHMAP CDATA #REQUIRED
    pHMO CDATA #REQUIRED
  >

<!ELEMENT RoomAndBoard (SummaryOfFee, OtherFundSource*)>

<!ELEMENT DrugsAndMedicine (SummaryOfFee, OtherFundSource*)>
```

```
<!ELEMENT LaboratoryAndDiagnostic (SummaryOfFee, OtherFundSource*)>

<!ELEMENT OperatingRoomFees (SummaryOfFee, OtherFundSource*)>

<!ELEMENT MedicalSupplies (SummaryOfFee, OtherFundSource*)>

<!ELEMENT ProfessionalFee (ProfessionalInfo, SummaryOfFee)>

<!ELEMENT ProfessionalInfo EMPTY>
<!ATTLIST ProfessionalInfo
    pPAN  CDATA #REQUIRED
    pFirstName CDATA #REQUIRED
    pMiddleName CDATA #REQUIRED
    pLastName CDATA #REQUIRED
    pSuffixName CDATA #REQUIRED
>

<!ELEMENT ItemizedBillingItem EMPTY>
<!ATTLIST ItemizedBillingItem
    pServiceDate CDATA #REQUIRED
    pItemCode CDATA #REQUIRED
    pItemName CDATA #REQUIRED
    pUnitOfMeasurement CDATA #REQUIRED
    pUnitPrice CDATA #REQUIRED
    pQuantity CDATA #REQUIRED
    pTotalAmount CDATA #REQUIRED
>
```

3. Validate CF5 Method

A method that will validate encrypted CF5 XML format against Document Type Definition (DTD), data format and values set by PhilHealth.

Important Note:
After successful validation, CF5 XML should be encrypted using PhilHealth Public Key and submitted as attachment to electronic claims using the **eclaimsUpload Method**.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/validateCF5`

Method

POST

Inputs

Headers

Key	Values
`token`	PECWS authentication token

Body

The JSON object contains the following key-value pairs:

Name	Values

cf5	The CF5 XML text encrypted using cipher key of the HF
	
	{
	"docMimeType": "text/xml",
	"hash": "dc8f4d74d977dfe701c0c9bbca000540591fac928f71a3841317e7a99aec",
	"key1": "",
	"key2": "",
	"iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
	"doc": "PMs1FWFZT+odAp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
	}

eclaims	The e-Claims XML text encrypted using cipher key of the HF
	
	{
	"docMimeType": "text/xml",
	"hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fa28f71a3841317e7a99aec",
	"key1": "",
	"key2": "",
	"iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
	"doc": "PMs1FWFZT+ag2dbj3gs123fsfq3ahqasfwnosfaqwrgssrjlsrqarewr=="
	}

Sample Input Payload

```
{
  "cf5": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
  "eclaims": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  }
}
```

Sample Unencrypted CF5 XML

```
<CF5
    pHospitalCode="300806">
  <DRGCLAIM
    ClaimNumber="300806-20221216-1-1"
    PrimaryCode="A00.0"
    NewBornTimeOfBirth=""
    NewBornAdmWeight="2"
```

```
    Remarks="">
    <SECONDARYDIAGS>
      <SECONDARYDIAG
        SecondaryCode="A00.1"
        Remarks=""/>
    </SECONDARYDIAGS>
    <PROCEDURES>
      <PROCEDURE
        RvsCode=""
        Laterality=""
        Ext1=""
        Ext2=""
        Remarks=""/>
    </PROCEDURES>
  </DRGCLAIM>
</CF5>
```

4. Request QR Authorization Method

This method shall be used to consume the eGovPH API that will allow PhilHealth Member to authorize sharing of their data with Health Facility, after their eGovPH Super App QR code has been scanned.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/requestQRAuthorization`

Method

POST

Inputs

Header

Key	Values
`token`	PECWS authentication token

Body

A JSON object containing the following key-value pairs:

Name	Values
`qr_value`	The text value of the scanned QR code

Sample value of the JSON object:

```
{
```

"qr_value":"eyJpdiI6IlRHSThkV2lPSmZnSkJnakREN2h0Mmc9PSIsInZhbHVlIjoiazkzejdkenRjUmtxVTY2Um10Q1JHMCt6UHZxUUo2eVpJQkt2V1NkNHBUQT0iLCJtYWMiOiJjZGVkMDI4ZmYxZjlmODcwY2E5NjVlZDE0NDRjYWVhNzEyODQ5ZTEyM2Y4M2JkMzk2OWQxODJhMjJmMzZlZDQ1In1"
}

Output

JSON object contains the following name value pairs:

Key	Value/Remarks
success	A value of 'true' indicates that data has been retrieved successfully
message	If an error was encountered during the execution of this method, this will contain the error message
trackingNo	The tracking number that will be used as input to the inquireQrTrackingNo method

Sample output value:


```
{
      "success": true
      "message": "",
      "trackingNo": "645D7CE4182A1"
 }
```


5. Inquire QR Tracking No Method

This method is called to validate authorization using the QR Code and provide the data of the member after the user permitted the sharing of their data using eGovPH Super App.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/inquireQrTrackingNo?trackingNo=`

Method

GET

Inputs

Headers

Key	Value
token	PECWS authentication token

Path Params

Parameter	Remarks
`trackingNo`	The value of the tracking number returned by the `requestQrAuthorization` method

Output

Type

JSON Object

Description

The object contains the following name-value pairs:

The result of the encryption of the JSON object of the PAN of the HF and the software certification ID of the system of the HF. The encrypted text contains the following key-value pairs

Name	Values/Remarks
`success`	A value of 'true' indicates that data has been retrieved successfully
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	Encrypted JSON object using cipher key of the HF

Name	Value/Remarks
`memberInfo`	Contains the following member data

Name	Value/Remarks
`address`	Address
`contactNoLocal`	Local contact number
`dateOfBirth`	Date of birth
`emailAddress`	Email address
`firstName`	First name
`middleName`	Middle name
`lastName`	Last name
`extName`	Ext/Suffix name
`memCat`	Member category
`mobileNo`	Cellphone number
`pin`	Philhealth Identification Number

		sex: M=Male; F=Female
	employmentInfo	Contains the following data of the employer of the member empName: Name empTelNo: Telephone number empNo: PhilHealth Employer Number
	dependentInfo	Array of the data of the dependents of the member

Sample output value:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```


Sample unencrypted result value:


```
      {
        "dependentInfo": [],
        "employmentInfo": {
            "empName": "ACME",
            "empNo": "410474000002",
            "empTelNo": "85555555"
        },
        "memberInfo": {
            "address": "METRO MANILA",
            "contactNoLocal": "555 55555555",
            "dateOfBirth": "1995-01-15",
            "emailAddress": "test@acme.com",
            "extName": "",
            "firstName": "TEST",
            "lastName": "TEST",
            "memCat": "DIRECT CONTRIBUTOR - EMPLOYED GOVERNMENT",
            "middleName": "TEST",
            "mobileNo": "0555555555",
            "pin": "310254165195",
            "sex": "F"
        }
    }
```

6. Upload eClaims Method

This method is for submitting electronic claims.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/uploadeClaims`

Method

POST

Inputs

Header

Key	Value
`token`	PECWS authentication token

Body

The body is the eclaims XML in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value/Remarks
`docMimeType`	`"text/xml"`
`hash`	The computed hash value of the unencrypted text.
`key1`	`""` (Empty string)
`key2`	`""` (Empty string)
`iv`	The initialization vector used in the AES encryption
`doc`	The encrypted e-claim XML text.

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message

<table>
<tr><td>result</td><td>The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.

<table>
<tr><th>Key</th><th>Value/Remarks</th></tr>
<tr><td>docMimeType</td><td>"text/xml"</td></tr>
<tr><td>hash</td><td>The computed hash value of the unencrypted text.</td></tr>
<tr><td>key1</td><td>"" (Empty string)</td></tr>
<tr><td>key2</td><td>"" (Empty string)</td></tr>
<tr><td>iv</td><td>The initialization vector used in the AES encryption</td></tr>
<tr><td>doc</td><td>The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.</td></tr>
</table>
</td></tr>
</table>

Sample encrypted output value:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```


Sample Decrypted Output XML Texts

A. Succesfully Received

```
<eRECEIPT
    pUserName=""
    pUserPassword=""
    pHospitalCode="123456"
    pHospitalTransmittalNo="001"
    pTotalClaims="1"
    pTransmissionControlNumber="1234-5601-1234-1253"
    pTransmissionDate="08-26-2009"
    pTransmissionTime="00:00:00AM"
    pReceiptTicketNumber="1234-5601-1234"
</eRECEIPT>
```

B. Unsuccessfully Received

```
<eRECEIPT
    pUserName=""
    pUserPassword=""
    pHospitalCode="123456"
```

```
    pHospitalTransmittalNo="001"
    pTotalClaims="1"
    pTransmissionControlNumber=""
    pTransmissionDate="08-26-2009"
    pTransmissionTime="00:00:00AM" >
    <REMARKS pErrCode="T01" pErrDescription="Invalid parameter value: pAmtActual" />
    <REMARKS pErrCode="T02" pErrDescription="Invalid parameter value: pOperationDate" />
</eRECEIPT>
```

7. Search Case Rates Method

A method that allows the caller to search for the case rate amounts with different applicable periods for target benefit packages.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/searchCaseRates`

Method

POST

Inputs

Header

Key	Value
`token`	PECWS authentication token

Body

The JSON object contains the following key-value pairs:

Name	Values
`icdcode`	The ICD code of the target benefit package
`rvscode`	The RVS code of the target benefit package
`description`	Substring or full text of the name or description of the target benefit packages
`targetdate`	Format: mm-dd-yyyy. If this parameter has a valid date value (which, normally the admission date of a target claim), this method will return only the record for the applicable

	period that covers the target date. If an empty string is passed as the value is passed of this parameter, all the applicable periods of the target benefit packages will be returned

Input Payload

```
{
  "icdcode": "",
  "rvscode": "",
  "description":"DENGUE",
  "targetdate":"02-14-2024"
}
```

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	The JSON object of the result of the encryption of the XML text containing the records of the matching benefit packages using the cipher key of the health facility. The JSON object contains the following key value pairs:

Key	Value/Remarks
docMimeType	"text/xml"
hash	The computed hash value of the unencrypted text.
key1	"" (Empty string)
key2	"" (Empty string)
iv	The initialization vector used in the AES encryption
doc	The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.

Sample Encrypted Output Payload

```
{
  "success": true,
  "message": "",
```

```
  "result": {
      "docMimeType": "text/xml",
      "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7c",
      "key1": "",
      "key2": "",
      "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
      "doc":"PMs1FWFZT+odArTp0qf2zMmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
    }
}
```


Sample Decrypted XML Text of the Output Payload


```
{"eCASERATES": {
       "CASERATES": [
              {"AMOUNT": {
                            "pPrimaryProfFee": "350.0",
                            "pPrimaryHCIFee": "2250.0",
                            "pCheckFacilityMAT": "F",
                            "pPrimaryCaseRate": "2600.0",
                            "pSecondaryHCIFee": "0.00",
                            "pSecondaryProfFee": "0.00",
                            "pSecondaryCaseRate": "0.00",
                            "pCheckFacilityH1": "F",
                            "pCheckFacilityH2": "F",
                            "pCheckFacilityFSDC": "F",
                            "pCheckFacilityH3": "F",
                            "pCheckFacilityASC": "F",
                            "pCheckFacilityPCF": "T",
                            "pCheckFacilityTSEKAP": "F",
                            "pCheckFacilityABTC": "F",
                            "pCheckFacilityTBDOTSC": "F",
                            "pCheckFacilityOPMC": "F",
                            "pCheckFacilityRHU": "F",
                            "pCheckFacilityDATRC": "F",
                            "pCheckFacilityHIVTH": "F",
                            "pCheckFacilityFPC": "F",
                            "pCheckFacilityCIU": "F",
                            "pCheckFacilityDSP": "F",
                            "pCheckFacilityPCB": "F”
                     },
                     "pCaseRateCode": "CR0389",
                     "pCaseRateDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G.
PERITONEAL, HEMOFILTRATION)",
                     "pItemCode": "90945",
                     "pItemDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G. PERITONEAL,
HEMOFILTRATION)",
                     "pEffectivityDate": "09-15-2015",
                     "pEffectivityEndDate": "02-13-2024"
              },
              {"AMOUNT": {
                            "pPrimaryProfFee": "500.0",
                            "pPrimaryHCIFee": "3500.0",
                            "pCheckFacilityMAT": "F",
                            "pPrimaryCaseRate": "4000.0",
                            "pSecondaryHCIFee": "0.00",
                            "pSecondaryProfFee": "0.00",
                            "pSecondaryCaseRate": "0.00",
                            "pCheckFacilityH1": "F",
                            "pCheckFacilityH2": "F",
                            "pCheckFacilityFSDC": "F",
                            "pCheckFacilityH3": "F",
```

```
                        "pCheckFacilityASC": "F",
                        "pCheckFacilityPCF": "T",
                        "pCheckFacilityTSEKAP": "F",
                        "pCheckFacilityABTC": "F",
                        "pCheckFacilityTBDOTSC": "F",
                        "pCheckFacilityOPMC": "F",
                        "pCheckFacilityRHU": "F",
                        "pCheckFacilityDATRC": "F",
                        "pCheckFacilityHIVTH": "F",
                        "pCheckFacilityFPC": "F",
                        "pCheckFacilityCIU": "F",
                        "pCheckFacilityDSP": "F",
                        "pCheckFacilityPCB": "F"
                },
                "pCaseRateCode": "CR0389",
                "pCaseRateDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G.
PERITONEAL, HEMOFILTRATION)",
                "pItemCode": "90945",
                "pItemDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G. PERITONEAL,
HEMOFILTRATION)",
                "pEffectivityDate": "12-02-2014",
                "pEffectivityEndDate": "09-14-2015"
                },
                {"AMOUNT": {
                        "pPrimaryProfFee": "500.0",
                        "pPrimaryHCIFee": "3500.0",
                        "pCheckFacilityMAT": "F",
                        "pPrimaryCaseRate": "4000.0",
                        "pSecondaryHCIFee": "0.00",
                        "pSecondaryProfFee": "0.00",
                        "pSecondaryCaseRate": "0.00",
                        "pCheckFacilityH1": "F",
                        "pCheckFacilityH2": "F",
                        "pCheckFacilityFSDC": "F",
                        "pCheckFacilityH3": "F",
                        "pCheckFacilityASC": "F",
                        "pCheckFacilityPCF": "T",
                        "pCheckFacilityTSEKAP": "F",
                        "pCheckFacilityABTC": "F",
                        "pCheckFacilityTBDOTSC": "F",
                        "pCheckFacilityOPMC": "F",
                        "pCheckFacilityRHU": "F",
                        "pCheckFacilityDATRC": "F",
                        "pCheckFacilityHIVTH": "F",
                        "pCheckFacilityFPC": "F",
                        "pCheckFacilityCIU": "F",
                        "pCheckFacilityDSP": "F",
                        "pCheckFacilityPCB": "F"
                },
                "pCaseRateCode": "CR0389",
                "pCaseRateDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G.
PERITONEAL, HEMOFILTRATION)",
                "pItemCode": "90945",
                "pItemDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G. PERITONEAL,
HEMOFILTRATION)",
                "pEffectivityDate": "01-01-2014",
                "pEffectivityEndDate": "12-01-2014"
                },
                {"AMOUNT": {
                        "pPrimaryProfFee": "455.0",
                        "pPrimaryHCIFee": "2925.0",
                        "pCheckFacilityMAT": "F",
                        "pPrimaryCaseRate": "3380.0",
```

```
                        "pSecondaryHCIFee": "0.00",
                        "pSecondaryProfFee": "0.00",
                        "pSecondaryCaseRate": "0.00",
                        "pCheckFacilityH1": "F",
                        "pCheckFacilityH2": "F",
                        "pCheckFacilityFSDC": "F",
                        "pCheckFacilityH3": "F",
                        "pCheckFacilityASC": "F",
                        "pCheckFacilityPCF": "T",
                        "pCheckFacilityTSEKAP": "F",
                        "pCheckFacilityABTC": "F",
                        "pCheckFacilityTBDOTSC": "F",
                        "pCheckFacilityOPMC": "F",
                        "pCheckFacilityRHU": "F",
                        "pCheckFacilityDATRC": "F",
                        "pCheckFacilityHIVTH": "F",
                        "pCheckFacilityFPC": "F",
                        "pCheckFacilityCIU": "F",
                        "pCheckFacilityDSP": "F",
                        "pCheckFacilityPCB": "F"
                    },
                    "pCaseRateCode": "CR0389",
                    "pCaseRateDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G. PERITONEAL, HEMOFILTRATION)",
                    "pItemCode": "90945",
                    "pItemDescription": "DIALYSIS PROCEDURE OTHER THAN HEMODIALYSIS (E.G. PERITONEAL, HEMOFILTRATION)",
                    "pEffectivityDate": "02-14-2024",
                    "pEffectivityEndDate": "12-31-9999"
                    }
            ]
        }
}
```


8. Add Required Document Method

A method that allows the caller to add required documents in compliance to RTH Claims.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/addRequiredDocument`

Method

POST

Header

Key	Value
token	PECWS authentication token

JSON Payload
JSON object containing the following key-value pairs

Key	Value/Remarks
`serieslhiono`	Series Lhio Number
`documents`	JSON Object list of documents to be added to the claim submitted **Key** / **Value/Remarks** `pDocumentType` — Document Type `pDocumentURL` — Document URL

```
{"serieslhiono": "",
 "documents":[
                    {"documenttype": "CF1", "documenturl": ""},
                    {"documenttype": "CF2", "documenturl": ""}
             ]
}
```

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	Possible results: "Claims has already been paid" "Claims has already been denied"

9. EClaims File Check Method

A method that allows the caller to validate the eClaims XML File.

Endpoint

`https://`***{pecws.domain}***`/PHIC/Claims3.0/eClaimsFileCheck`

Method

POST

Header

Key	Value
`token`	PECWS authentication token

Body

The body is the eclaims XML in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value/Remarks
`docMimeType`	`"text/xml"`
`hash`	The computed hash value of the unencrypted text.
`key1`	`""` (Empty string)
`key2`	`""` (Empty string)
`iv`	The initialization vector used in the AES encryption
`doc`	The encrypted e-claim XML text.

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.

Key	Value/Remarks
`docMimeType`	`"text/xml"`
`hash`	The computed hash value of the unencrypted text.
`key1`	`""` (Empty string)
`key2`	`""` (Empty string)
`iv`	The initialization vector used in the AES encryption
`doc`	The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.

Sample encrypted output value:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```


10. Get Claim Status Method

A method for getting the processing status of claims.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getClaimStatus`

Method

GET

Header

Key	Value
`token`	PECWS authentication token

Body

The body is the eclaims XML in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value
`serieslhionos`	JSON Object list of Strings containing multiple claims Series Nos

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation

<table>
<tr><td>message</td><td>If an error was encountered during the execution of this method, this will contain the error message</td></tr>
<tr><td>result</td><td>The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.
<table>
<tr><th>Key</th><th>Value/Remarks</th></tr>
<tr><td>docMimeType</td><td>"text/xml"</td></tr>
<tr><td>hash</td><td>The computed hash value of the unencrypted text.</td></tr>
<tr><td>key1</td><td>"" (Empty string)</td></tr>
<tr><td>key2</td><td>"" (Empty string)</td></tr>
<tr><td>iv</td><td>The initialization vector used in the AES encryption</td></tr>
<tr><td>doc</td><td>The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.</td></tr>
</table>
</td></tr>
</table>

Sample JSON input payload:

```
{"serieslhionos": ["1","2"]}
```

Sample encrypted output value:

```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```

Sample decrypted result value:

```
{
      "CLAIMS": [
            {
                  "STATUS": {
                        "CLAIM": {
                              "TRAIL": {
                                    "PROCESS": [
                                          {
                                                "pProcessStage": "VALIDATION",
                                                "pProcessDate": "07-25-2012"
                                          },
                                          {
                                                "pProcessStage": "EDITING",
                                                "pProcessDate": "07-25-2012"
                                          },
                                          {
                                                "pProcessStage": "VALIDATION",
```

```
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "EDITING (RECEIVING)",
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "ENCODING",
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "RECEIVING",
                                        "pProcessDate": "07-23-2012"
                                    }
                                ]
                            },
                            "pClaimSeriesLhio": "120723190000119",
                            "pPin": "190892937993",
                            "pPatientLastName": "LASTNAME",
                            "pPatientFirstName": "FIRSTNAME",
                            "pPatientMiddleName": "MIDDLENAME",
                            "pPatientSuffix": "",
                            "pAdmissionDate": "05-02-2012",
                            "pDischargeDate": "05-06-2012",
                            "pClaimDateReceived": "05-15-2012",
                            "pClaimDateRefile": "",
                            "pStatus": "IN PROCESS"
                        },
                        "pAsOf": "07-25-2012",
                        "pAsOfTime": "04:46:23PM"
                    }
                },
                {
                    "STATUS": {
                        "CLAIM": {
                            "TRAIL": {
                                "PROCESS": [
                                    {
                                        "pProcessStage": "VALIDATION",
                                        "pProcessDate": "07-25-2012"
                                    },
                                    {
                                        "pProcessStage": "EDITING",
                                        "pProcessDate": "07-25-2012"
                                    },
                                    {
                                        "pProcessStage": "VALIDATION",
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "EDITING (RECEIVING)",
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "ENCODING",
                                        "pProcessDate": "07-23-2012"
                                    },
                                    {
                                        "pProcessStage": "RECEIVING",
                                        "pProcessDate": "07-23-2012"
                                    }
                                ]
                            },
                            "pClaimSeriesLhio": "120723190000119",
                            "pPin": "190592937994",
```

```
                        "pPatientLastName": "LASTNAME",
                        "pPatientFirstName": "FIRSTNAME",
                        "pPatientMiddleName": "MIDDLENAME",
                        "pPatientSuffix": "III",
                        "pAdmissionDate": "05-02-2012",
                        "pDischargeDate": "05-06-2012",
                        "pClaimDateReceived": "05-15-2012",
                        "pClaimDateRefile": "",
                        "pStatus": "IN PROCESS"
                    },
                    "pAsOf": "07-25-2012",
                    "pAsOfTime": "04:46:23PM"
                }
            }
        ]
}
```


11. Get Doctor PAN Method

A method for getting the PhilHealth accreditation number (PAN) of a health care professional (HCP) using the information of the professional such as the complete name and the date of birth.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getDoctorPAN`

Method

POST

Header

Key	Value
token	PECWS authentication token

JSON Payload

The body is in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value
tin	Health Care Professional TIN
lastname	Health Care Professional Last Name
firstname	Health Care Professional First Name
middlename	Health Care Professional Middle Name
suffix	Health Care Professional Suffix Name
birthdate	Health Care Professional Birth Date (Date Format should be : 'MM-DD-YYYY')

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	Possible result: If ECWS finds a matching record for the passed data of the HCP, the method will return the 12-digit PAN in the format *"0000-0000000-0"* If ECWS does not find a matching record, ECWS will return a blank string ("") If ECWS encountered an exception/error while querying for the records of the HCP, ECWS will return the exception/error message

Sample encrypted JSON input payload:

```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  }
```

Sample decrypted JSON input payload:

```
{"tin":"1234567890",
 "lastname":"LASTNAME",
 "firstname":"FIRSTNAME",
 "middlename":"MIDDLENAME",
 "suffix":"III",
 "birthdate":"01-01-1990"}
```

12. Get Member PIN Method

This method is to be used to query the PhilHealth Identification Number (PIN) of a PhilHealth member using the complete name of the date of birth of the member.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getMemberPIN`

Method

POST

Header

Key	Value
token	PECWS authentication token

JSON Payload

The body is in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value
lastname	Member Last Name
firstname	Member First Name
middlename	Member Middle Name
suffix	Member Suffix Name
birthdate	Member Birth Date (Date Format should be : 'MM-DD-YYYY')

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	Possible result: If method finds a matching record for the passed data of the Member, the method will return the 12-digit PIN in the format *"00-0000000000"* If method does not find a matching record, method will return a blank string ("") If method encountered an exception/error while querying for the records of the Member method will return the exception/error message

Sample encrypted JSON input payload:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  }
```

Sample decrypted JSON input payload:

```
{"lastname":"LASTNAME",
 "firstname":"FIRSTNAME",
 "middlename":"MIDDLENAME",
 "suffix":"III",
 "birthdate":"01-01-1990"}
```

13. Get Uploaded Claims Map Method

To facilitate claim reconciliation and verification, the health facility and the PhilHealth must have a common reference for a given claim. The PhilHealth Claim Serier Number is used for that purpose. The `getUploadedClaimsMap` method will map the ID generated by the system of the health facility to the Claim Series Number generated by PhilHealth for each of the claims included in the e-claim XML file that was submitted through the eClaimsUpload method. The `receiptTicketNo` parameter of the `getUploadedClaimsMap` method should be the value of the `pReceiptTicketNumber` attribute returned by the eClaimsUpload method for those set of claims.

Endpoint

`https://`***{pecws.domain}***`/PHIC/Claims3.0/getUploadedClaimsMap`

Method

GET

Header

Key	Value
`token`	PECWS authentication token

Parameter

Key	Value
`receiptTicketNo`	The receipt ticket number is generated after calling the eClaimsUpload method

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of

	the submitted e-claim data. **Key** / **Value/Remarks** `docMimeType` — `"text/xml"` `hash` — The computed hash value of the unencrypted text. `key1` — "" (Empty string) `key2` — "" (Empty string) `iv` — The initialization vector used in the AES encryption `doc` — The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.

Sample encrypted result:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```


Sample decrypted result:


```
{
      "eCONFIRMATION": {
            "MAPPING": {
                  "pClaimNumber": "09-08-01-006",
                  "pPatientLastName": "LASTNAME",
                  "pPatientFirstName": "MARIA",
                  "pPatientMiddleName": "C",
                  "pPatientSuffix": "",
                  "pAdmissionDate": "08-25-2009",
                  "pDischargeDate": "08-25-2009",
                  "pClaimSeriesLhio": "090801990000199"
            },
            "pReceiptTicketNumber": "071311000005",
            "pHospitalCode": "300832",
            "pHospitalTransmittalNo": "3008321107000008",
            "pTotalClaims": "1",
            "pReceivedDate": "09-13-2009"
      }
}
```

14. Get Voucher Details Method

This method can be used to facilitate reconciliation of paid claims This method returns the voucher and other payment details of claims. The value of `voucherNo` parameter can be set to the value of the `pVoucherNo` attribute returned by the GetClaimStatus method if a voucher has already been prepared for the payment of a claim. Take note that there can be more than one claim in voucher.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getVoucherDetails`

Method

GET

Header

Key	Value
`token`	PECWS authentication token

Parameter

Key	Value
`voucherNo`	Valid voucher number

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.

Key	Value/Remarks
`docMimeType`	`"text/xml"`
`hash`	The computed hash value of the unencrypted text.

key1	"" (Empty string)
key2	"" (Empty string)
iv	The initialization vector used in the AES encryption
doc	The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.

Sample encrypted result:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```


Sample decrypted result:


```
{
      "VOUCHER": {
            "CLAIM": [
                  {
                        "CHARGE": [
                              {
                                    "pPayeeType": "C",
                                    "pPayeeCode": "2XX25",
                                    "pPayeeName": "DR. DJXXXXX XXX X. SXXX",
                                    "pRMBD": "0.00",
                                    "pDRUGS": "0.00",
                                    "pXRAY": "0.00",
                                    "pOPRM": "0.00",
                                    "pSPFee": "0.00",
                                    "pGPFee": "0.00",
                                    "pSURFee": "4000.00",
                                    "pANESFee": "0.00",
                                    "pGrossAmount": "4000.00",
                                    "pTaxAmount": "0.00",
                                    "pNetAmount": "4000.00"
                              },
                              {
                                    "pPayeeType": "C",
                                    "pPayeeCode": "3XX25",
                                    "pPayeeName": "DR. IXX OLXXXX A. CANXXXX",
                                    "pRMBD": "0.00",
                                    "pDRUGS": "0.00",
                                    "pXRAY": "0.00",
                                    "pOPRM": "0.00",
                                    "pSPFee": "0.00",
```

```
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "1200.00",
                        "pGrossAmount": "1200.00",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "1200.00"
                    },
                    {
                        "pPayeeType": "H",
                        "pPayeeCode": "30XX04",
                        "pPayeeName": "XXXX CITY XXXXXX HOSPITAL",
                        "pRMBD": "800.00",
                        "pDRUGS": "507.50",
                        "pXRAY": "994.40",
                        "pOPRM": "3490.00",
                        "pSPFee": "0.00",
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "0.00",
                        "pGrossAmount": "5791.90",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "5791.90"
                    },
                    {
                        "pPayeeType": "M",
                        "pPayeeCode": "P192003617072",
                        "pPayeeName": "ALOXXX , BERNAXXX X",
                        "pRMBD": "0.00",
                        "pDRUGS": "2544.00",
                        "pXRAY": "141.00",
                        "pOPRM": "0.00",
                        "pSPFee": "0.00",
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "0.00",
                        "pGrossAmount": "2685.00",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "2685.00"
                    }
                ],
                "pClaimSeriesLhio": "060516030019903",
                "pPin": "192003610605",
                "pPatientLastName": "ALOXXX",
                "pPatientFirstName": "BERNAXXX",
                "pPatientMiddleName": "B",
                "pPatientSuffix": "",
                "pAdmissionDate": "04-18-2006",
                "pDischargeDate": "04-20-2006",
                "pClaimDateReceived": "05-12-2006",
                "pClaimDateRefile": "",
                "pIsAdjustment": "F"
            },
            {
                "CHARGE": [
                    {
                        "pPayeeType": "C",
                        "pPayeeCode": "27XX2",
                        "pPayeeName": "DR. VIRXXX XX. X. DXXXO",
                        "pRMBD": "0.00",
                        "pDRUGS": "0.00",
                        "pXRAY": "0.00",
                        "pOPRM": "0.00",
                        "pSPFee": "0.00",
                        "pGPFee": "600.00",
                        "pSURFee": "0.00",
```

```
                                    "pANESFee": "0.00",
                                    "pGrossAmount": "600.00",
                                    "pTaxAmount": "0.00",
                                    "pNetAmount": "600.00"
                              },
                              {
                                    "pPayeeType": "H",
                                    "pPayeeCode": "3XXX04",
                                    "pPayeeName": "XXXX CITY XXXXX HOSPITAL",
                                    "pRMBD": "2400.00",
                                    "pDRUGS": "240.00",
                                    "pXRAY": "671.25",
                                    "pOPRM": "0.00",
                                    "pSPFee": "0.00",
                                    "pGPFee": "0.00",
                                    "pSURFee": "0.00",
                                    "pANESFee": "0.00",
                                    "pGrossAmount": "3311.25",
                                    "pTaxAmount": "0.00",
                                    "pNetAmount": "3311.25"
                              },
                              {
                                    "pPayeeType": "M",
                                    "pPayeeCode": "P19200XX31034",
                                    "pPayeeName": "ANDXXX , CONCHXXX X",
                                    "pRMBD": "0.00",
                                    "pDRUGS": "1157.50",
                                    "pXRAY": "863.00",
                                    "pOPRM": "0.00",
                                    "pSPFee": "0.00",
                                    "pGPFee": "0.00",
                                    "pSURFee": "0.00",
                                    "pANESFee": "0.00",
                                    "pGrossAmount": "2020.50",
                                    "pTaxAmount": "0.00",
                                    "pNetAmount": "2020.50"
                              }
                        ],
                        "pClaimSeriesLhio": "060516030031234",
                        "pPin": "192005981034",
                        "pPatientLastName": "ANDXXX",
                        "pPatientFirstName": "CONCHXXX",
                        "pPatientMiddleName": "XXXX",
                        "pPatientSuffix": "",
                        "pAdmissionDate": "04-21-2006",
                        "pDischargeDate": "04-27-2006",
                        "pClaimDateReceived": "05-12-2006",
                        "pClaimDateRefile": "",
                        "pIsAdjustment": "F"
                  }
            ],
            "SUMMARY": {
                  "PAYEE": [
                        {
                              "pPayeeType": "C",
                              "pPayeeCode": "30XX04",
                              "pPayeeName": "HC- XXXX CITY XXXXX HOSPITAL ",
                              "pRMBD": "0.00",
                              "pDRUGS": "0.00",
                              "pXRAY": "0.00",
                              "pOPRM": "0.00",
                              "pSPFee": "0.00",
                              "pGPFee": "600.00",
                              "pSURFee": "4000.00",
                              "pANESFee": "1200.00",
```

```
                        "pGrossAmount": "5800.00",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "5800.00",
                        "pCheckNo": "0000XXX429",
                        "pCheckDate": "06-19-2006"
                    },
                    {
                        "pPayeeType": "H",
                        "pPayeeCode": "30XX04",
                        "pPayeeName": "XXXX CITY XXXXX HOSPITAL",
                        "pRMBD": "3200.00",
                        "pDRUGS": "747.50",
                        "pXRAY": "1665.65",
                        "pOPRM": "3490.00",
                        "pSPFee": "0.00",
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "0.00",
                        "pGrossAmount": "9103.15",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "9103.15",
                        "pCheckNo": "0000XXX430",
                        "pCheckDate": "06-19-2006"
                    },
                    {
                        "pPayeeType": "M",
                        "pPayeeCode": "P1920XX987072",
                        "pPayeeName": "ALOXXX , BERNXXXX X",
                        "pRMBD": "0.00",
                        "pDRUGS": "2544.00",
                        "pXRAY": "141.00",
                        "pOPRM": "0.00",
                        "pSPFee": "0.00",
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "0.00",
                        "pGrossAmount": "2685.00",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "2685.00",
                        "pCheckNo": "0000XXX431",
                        "pCheckDate": "06-19-2006"
                    },
                    {
                        "pPayeeType": "M",
                        "pPayeeCode": "P1920XX731034",
                        "pPayeeName": "ANXXXX , CONXXXXX X",
                        "pRMBD": "0.00",
                        "pDRUGS": "1157.50",
                        "pXRAY": "863.00",
                        "pOPRM": "0.00",
                        "pSPFee": "0.00",
                        "pGPFee": "0.00",
                        "pSURFee": "0.00",
                        "pANESFee": "0.00",
                        "pGrossAmount": "2020.50",
                        "pTaxAmount": "0.00",
                        "pNetAmount": "2020.50",
                        "pCheckNo": "0000XXX432",
                        "pCheckDate": "06-19-2006"
                    }
                ],
                "pTotalAmount": "19608.65",
                "pNumberOfClaims": "2"
            },
            "pVoucherNo": "201-062001-06I03",
```

```
            "pVoucherDate": "06-14-2006"
        }
}
```

15. Is Doctor Accredited Method

A method that allows the caller to verify if the Health Care Professional is accredited based on the given encrypted JSON payload containing the Accreditation number, Admission date and discharge date.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/isDoctorAccredited`

Method

POST

Header

Key	Value
`token`	PECWS authentication token

JSON Payload

JSON object containing the following key-value pairs

Key	Value
`accrecode`	Health Care Professional Accreditation Number
`admissiondate`	Date of Admission (Date Format should be : 'MM-DD-YYYY')
`dischargedate`	Date of Discharge (Date Format should be : 'MM-DD-YYYY')

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.

Key	Value/Remarks
`isaccredited`	
`accrecode`	Health Care Professional Accreditation Number
`admissiondate`	Date of Admission (Date Format : 'MM-DD-YYYY')

`dischargedate`	Date of Discharge (Date Format : 'MM-DD-YYYY')
`accreditationstart`	Date of Start of Accreditation (Date Format : 'MM-DD-YYYY')
`accreditationend`	Date of End of Accreditation (Date Format : 'MM-DD-YYYY')

Sample JSON input payload:

```
{"accreCode":"1234567890",
 "admissionDate":"01-01-1990",
 "dischargeDate":"01-01-1990"}
```

16. Search Employer Method

A method that allows the caller to search for the employer an employed member based on PhilHealth Employer Number and/or the employer name.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/searchEmployer`

Method

POST

Header

Key	Value
`token`	PECWS authentication token

JSON Payload

JSON object containing the following key-value pairs

Key	Value
`philhealthno`	PhilHealth Employer Number.
`employername`	The full name or a search pattern for the name of an mployer. Wildcard characters are supported. The percent (%) wildcard matches one or more characters. For example, passing "PHILIPPINE%" for this key matches names of employers starting with "PHILIPPINE"

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks

success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	The JSON object as the result of the encryption (using the cipher key of the health facility) of the XML text containing the Receipt Ticket Number and other data about the processing of the submitted e-claim data.

Key	Value/Remarks
docMimeType	"text/xml"
hash	The computed hash value of the unencrypted text.
key1	"" (Empty string)
key2	"" (Empty string)
iv	The initialization vector used in the AES encryption
doc	The encrypted text of the JSON object containing the records of the matching benefit packages. Sample XML text and the DTD of the XML text is shown below.

Sample JSON input:

```
{

    "philhealthno": "123456789012,
    "employername": "EMP1"
}
```

Sample JSON output:

```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  },
    "success": true,
    "message": ""
}
```

Sample decrypted result:

```
{
      "eEMPLOYERS": {
            "employer": [{
                  "pPEN": "123456789012",
                  "pEmployerName": "EMP1",
                  "pEmployerAddress": "EMP1"
            },
        {
                  "pPEN": "123456789013",
```

```
                "pEmployerName": "EMP2",
                "pEmployerAddress": ""
            }],
            "ASOF": "01-01-2024"
        }
}
```


17. Get DB Server Date Time Method

A method that allows the caller to get the current DB Server Date and Time. This method can be used by the caller to check if one of the databases of PhilHeath is down or not available.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getDBServerDateTime`

Method

GET

Header

Key	Value
token	PECWS authentication token

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	The JSON Array containing the following key-value pairs

Key	Value/Remarks
server	Database Server
dateTime	Database current date and time
remarks	

Sample JSON output:


```
{
    "result": [{
                "server": "DB Server 1",
```

```
                "datetime": "01-01-2024 01:20:20 PM",
                "remarks": ""
            },
            {
                "server": "DB Server 2",
                "datetime": "01-01-2024 01:20:20 PM",
                "remarks": ""
            }],
    "success": true,
    "message": ""
}
```


18. Get Server Date Time Method

A method that allows the caller to get the current server date and time. This method can be used by the caller to determine the current status of the API or its server --whether it is active or down.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getServerDateTime`

Method

GET

Header

Key	Value
token	PECWS authentication token

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	The JSON Object containing the following key-value pairs

Key	Value/Remarks
server	Database Server
dateTime	Database current date and time
remarks	

Sample JSON output:


```
{
    "result": {
                    "server": "DB Server 1",
                    "datetime": "01-01-2024 01:20:20 PM",
                    "remarks": ""
                }
    "success": true,
    "message": ""
}
```


19. Get Server Version Method

A method that allows the caller to get the current Server version.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/getServerVersion`

Method

GET

Header

Key	Value
`token`	PECWS authentication token

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
`success`	A value of 'true' indicates a successful operation
`message`	If an error was encountered during the execution of this method, this will contain the error message
`result`	PECWS Version

Sample JSON output:


```
{
    "result": "PECWS 3.0",
    "success": true,
    "message": ""
}
```

20. Is Claim Eligible Method

A method that allows the caller to check the eligibility of the claim based on the encrypted JSON object containing the JSON Payload.

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/isClaimEligible`

Method

POST

Header

Key	Value
`token`	PECWS authentication token

JSON Payload

The body is in JSON format as encrypted using the cipher key of the Health Facility. The JSON object contains the following key-value pairs:

Key	Value
`isForOPDHemodialysisClaim`	If the claim is for OPD Hemodialysis Claim (Y/N)
`memberpPIN`	Member PIN
`memberBasicInformation`	Basic Information
`patientIs`	Patient Type (MM - Member /DD - Dependent)
`admissionDate`	Admission Date
`patientPIN`	Valid PIN of the patient
`patientBasicInformation`	Basic Information
`membershipType`	Member Category
`pEN`	Employer PEN
`employerName`	Employer Name
`isFinal`	If the Claim is final (0/1)

Basic Information

Key	Value
`lastname`	Last Name
`firstname`	First Name
`middlename`	Middle Name
`maidenname`	Maiden Name
`suffix`	Suffix Name

sex	Sex
dateOfBirth	Birth Date (Date Format should be : 'MM-DD-YYYY')

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks
success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	The JSON Object containing the following key-value pairs **Key** / **Value/Remarks** isOk pBEFReferenceNo trackingNumber asOf

Sample encrypted JSON input payload:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  }
```


Sample decrypted JSON input payload:


```
{"hospitalCode":"",
  "isForOPDHemodialysisClaim":"",
  "memberpPIN":"",
  "memberBasicInformation":{"lastname":"",
  "firstname":"",
  "middlename":"",
  "maidenname":"",
  "suffix":"",
  "sex":"",
  "dateOfBirth":""},
  "patientIs":"",
  "admissionDate":"",
  "patientPIN":"",
  "patientBasicInformation":{"lastname":"",
```

```
  "firstname":"",
  "middlename":"",
  "maidenname":"",
  "suffix":"",
  "sex":"",
  "dateOfBirth":""},
  "membershipType":"",
  "pEN":"",
  "employerName":"",
  "isFinal":""}
```

Sample JSON output:

```
{
    "result": {
                    "isOk": "YES",
                    "pBEFReferenceNo": "",
                    "trackingNumber": "",
                    "asOf": ""
                }
    "success": true,
    "message": ""
}
```

21. Generate PBEF PDF Method

A method that allows the caller to generate the PDF file of the PhilHealth Benefit Eligibiity Form (PBEF).

Endpoint

`https://{pecws.domain}/PHIC/Claims3.0/generatePBEFPDF?pBEFReferenceNo=`

Method

GET

Header

Key	Value
`token`	PECWS authentication token

Parameter

Key	Value
`pBEFReferenceNo`	Result from isClaimEligible method

Output

This method returns a JSON object with the key-value pairs described below:

Key	Value/Remarks

success	A value of 'true' indicates a successful operation
message	If an error was encountered during the execution of this method, this will contain the error message
result	Byte value of the PBEF PDF

Sample encrypted JSON input payload:


```
{
    "result": {
    "docMimeType": "text/xml",
    "hash": "dc8f4d74d977dfe701c0c9bbca0678300540591fac928f71a3841317e7a99aec",
    "key1": "",
    "key2": "",
    "iv": "y1jPMxvQE2aJPVnqqn1pDQ==",
    "doc": "PMs1FWFZT+odAp0qf2zManmroSUr3lYgDFnhYeJqBkuhJNMJU5geEN=="
  }
```


Sample decrypted JSON input payload:


```
{"hospitalCode":"",
  "isForOPDHemodialysisClaim":"",
  "memberpPIN":"",
  "memberBasicInformation":{"lastname":"",
  "firstname":"",
  "middlename":"",
  "maidenname":"",
  "suffix":"",
  "sex":"",
  "dateOfBirth":""},
  "patientIs":"",
  "admissionDate":"",
  "patientPIN":"",
  "patientBasicInformation":{"lastname":"",
  "firstname":"",
  "middlename":"",
  "maidenname":"",
  "suffix":"",
  "sex":"",
  "dateOfBirth":""},
  "membershipType":"",
  "pEN":"",
  "employerName":"",
  "isFinal":""}
```


Sample JSON output:


```
{
    "result": {
                    "isOk": "YES",
                    "pBEFReferenceNo": "",
                    "trackingNumber": "",
                    "asOf": ""
                }
```

```
    "success": true,
    "message": ""
}
```

Annex A

Guidelines for the Data Encryption Using the Cipher Key of the Health Facility

Data will be encrypted using AES-256-CBC algorithm. PhilHealth issues a *cipher key* to the health facility for each certified software. The cipher key is to be used in generating the secret key for the AES encryption. System developlers of service providers/health facilities may use their preferred programming language and tools for the data encryption/decryption.

```
{
        "docMimeType": "{MIME type of the data}",
        "hash": "{SHA-256 hash of the data before encryption}",
        "key1": "",
        "key2": "",
        "iv": "{The initialization vector encoded as base-64 string}",
        "doc": "{The encrypted data encoded as base-64 string}"
}
```

Figure 1: Sample Format/Layout of the Resulting Encrypted Data

1. Get the hash total of the data to be decrypted using SHA-256 hash algorithm. After decrypting the data, a new hash total can be computed using the same algorithm, which can then be compared to the original hash total generated before decryption. If the two hash totals match, the data integrity is preserved. However, if the hash totals do not match, the data may have been tampered with or corrupted during decryption.

2. Set the value of the secret key to be used for the AES encryption.

    a. Hash the given cipher key using SHA-256 algorithm.

    b. Use the first 32 bytes of the resulting hash value as the passphrase for the AES encryption.

    c. If the length of the resulting hash value is less than 32 bytes, pad the hash value with null character (with hexadecimal value of '0x00') to have a value having 32 bytes.

3. Prepare the initialization vector (IV) to be used

    a. Generate an array of random 16 bytes (for a total of 128 bits).

    b. Encode the array of bytes as base 64 string

4. Encrypt the data

    a. Convert the data as array of bytes

    b. Encrypt the data using AES encryption using the values the array of bytes of the password, the IV and the data.

    c. Encode the resulting encrypted array of bytes as base-64 string.

5. Build the JSON string of the encrypted data using the following names and their corresponding values:

    a. `docMimeType`: The value for this key will be the MIME type of the target data. For example, if the target data is a string in JSON format, the value for this key should be "`application/json`".

    b. `hash`: Use the resulting value of Step #1 as the value of this key

    c. `key1`: Set the value of this key as empty string.

    d. `key2`: Set the value as empty string.

    e. `iv`: Use the resulting value of Step #3.b as the value of this key

    f. `doc`: Use the resulting value of Step #4.c as the value of this key

Annex B

Document Type Code and Description

Document Code	Description
CAB	Clinical Abstract
CAE	Certification of Approval/Agreement from the Employer
CF1	Claim Form 1
CF2	Claim Form 2
CF3	Claim Form 3
CF4	Claim Form 4
COE	Certificate of Eligibility
CSF	Claim Signature Form
CTR	Confirmatory Test Results by SACCL or RITM
DTR	Diagnostic Test Result
ITB	Itemized Billing (PDF format)
ITX	Itemized Billing (MS Excel Format)
MBC	Member's Birth Certificate
MDR	Proof of MDR with Payment Details
MEF	Member Empowerment Form
MMC	Member's Marriage Contract
MSR	Malarial Smear Results
NHC	Newborn Hearing Registry Card (Blue Card)
NHT	Newborn Hearing Screening Test Result
MWV	Waiver for Consent for Release of Confidential Patient Health Information
NTP	NTP Registry Card
OPR	Operative Record
ORS	Official Receipts
PAC	Pre-Authorization Clearance
PBC	Patient's Birth Certificate
PIC	Valid Philhealth Indigent ID
POR	PhilHealth Official Receipts
SOA	Statement of Account
STR	HIV Screening Test Result

TCC	TB-Diagnostic Committee Certification (-) Sputum
TYP	Three Years Payment of (2400 x 3 years of proof of payment)
MRF	PhilHealth Member Registration Form
ANR	Anesthesia Record
HDR	Hemodialysis Record
CF4	Claim Form 4
CF5	Electronic Claim Form 5 (for DRG Shadow Billing)
ESA	Electronic Statement of Account (eSOA)
NIR	Neuroimaging Result
RGR	Radiographic Result
BCR	Blood ? Result
CBC	Complete Blood Count
CPR	? ? Result
OTH	Other documents

Annex C
Data Dictionary eClaimsUpload

Name	Length	Description	Valid Values
pUserName	String(20)	Provider user id	To be provided by PhilHealth
pUserPassword	String(20)	Provider user password	To be provided by PhilHealth
pHospitalCode	String(12)	Facility Accreditation Number	For now PMCC number should be used
pHospitalEmail	String(150)	Hospital Email Address where communication will be sent	Must not be blank
pHospitalTransmittalNo		Hospital Transmittal Number	Generated by the Hospital own batching system. This should be unique per hospital.
pTotalClaims	String(3)	Claims counter	Integer format
pClaimNumber	String(12)	Hospital Claim Number	Hospital Generated Claim Case #, this should be unique per hospital
pTrackingNumber	String(20)	The Claims Eligibility Tracking number assigned if undergone the Online Eligibility Checking	Formatted as: '####-####-####-####' Can be blank
pCataractPreAuth	String(20)	Cataract Pre-Authorization Application Number	
pPhilhealthClaimType	String(20)	Flag whether Claims Payment Mechanism	● 'ALL-CASE-RATE' ● 'Z-BENEFIT'
pPatientType	String(1)	Patient Type	● 'I' – Inpatient ● 'O' – Outpatient
pIsEmergency	String(1)	Flag if Emergency Case	• 'Y' – Yes ● 'N' – No
pClaimSeriesLhio	String(15)	Philhealth Generated and Assigned Unique Number per Claim	Can be used by the hospital to reconcile their records with Philhealth This will be returned after the claim are uploaded to Philhealth
pMemberPIN	String(12)	PhilHealth Identification Number – a unique 12 digit number assigned to a member.	The last character in the PIN is a modulus 11 check digit.
pMemberLastName	String(60)	Member's Complete Surname	Any value consisting of : 'A' to 'Z', 'Ñ'. Can include a space in between characters
pMemberFirstname	String(60)	Member's Complete First name	
pMemberMiddleName	String(60)	Member's Complete Middle name	
pMemberSuffix	String(5)	Member's Suffix name	'JR', 'SR', 'III', …etc Suffixes can be blank
pMemberBirthDate	String(10)	Member's Birth Date	Date Format should be : 'MM-DD-YYYY'
pMemberShipType	String(2)	Philhealth membership type of the member	(Not limited to the following :) ● 'S' – Employed Private ● 'G' – Employer Government ● 'I' – Indigent ● 'NS' – Individually Paying ● 'NO' – OFW ● 'PS' – Non Paying Private ● 'PG' – Non Paying Government ● 'P' – Lifetime Member
pMailingAddress	String(150)	Mailing Address (address where the benefit payment notice will be sent)	This is where the notices will be mailed.
pZipCode	String(4)	Philippine Zip Code of the municipality	4 digit Philippine zip code value of the municipality/city
pMemberSex	String(1)	Member Sex	● 'M' – Male ● 'F' – Female

pLandlineNo	String(20)	Members Landline Number	Can be blank
pMobileNo	String(20)	Members Cell Number	Can be blank
pEmailAddress	String(150)	Email Address	Can be blank
pPatientIs	String(1)	Flag whether patient is the member or if dependent the relationship of patient with the member.	● 'M' – patient is member(Self) ● 'S' – patient is spouse ● 'C' – patient is child ● 'P' – patient is parent
pPatientPIN	String(12)	PhilHealth Identification Number – a unique 12 digit number assigned to a patient.	The last character in the PIN is a modulus 11 check digit.
pPatientLastName	String(60)	Patient's Complete Surname	Same as for the member. These can be blank since these are disregarded if the value of pPatientIs is 'M'
pPatientFirstName	String(60)	Patient's Complete First name	
pPatientMiddleName	String(60)	Patient's Complete Middle name	
pPatientSuffix	String(5)	Patient's Suffix name	'JR', 'SR', 'III', …etc Suffixes can be blank
pPatientBirthDate	String(10)	Patient's Birth Date	Date Format should be : 'MM-DD-YYYY'
pPatientSex	String(1)	Member Sex	● 'M' – Male ● 'F' – Female
pPEN	String(12)	Philhealth Employer Number – a unique 12 digit number assigned to an employer	These are disregarded if pMemberShipType is not ('S' or 'G')
pEmployerName	String(100)	The Registered name of the employer	
pPatientReferred	String(1)	Referred Patient	● 'Y' – Yes ● 'N' – No
pReferredIHCPAccreCode	String(12)	Referring Facility Accreditation Code	Required if the patient is referred by another IHCP
pAdmissionDate	String(10)	Admission Date	Date Format should be : 'MM-DD-YYYY'
pDischargeDate	String(10)	Discharge Date	
pAdmissionTime	String(10)	Admission Time	Time Format should be : 'HH:MM:SSAM/PM'
pDischargeTime	String(10)	Discharge Time	
pDisposition	String(1)	Patient's Disposition	● 'I' – Improved ● 'R' – Recovered ● 'H' – Home/Discharged Against Medical Advise ● 'A' – Absconded ● 'E' – Expired ● 'T' – Transferred/Referred
pExpiredDate	String(10)	Date of Death of patient	Date Format should be : 'MM-DD-YYYY' Required when pDisposition = 'E'
pExpiredTime	String(10)	Time of Death of patient	Time Format should be : 'HH:MM:SSAM/PM' Required when pDisposition = 'E'
pReferralIHCPAccreCode	String(12)	Referral Facility Accreditation Code	Required when pDisposition = 'T'
pReferralReasons	String(150)	Reason/s for referral/transfer	
pAccommodationType	String(1)	Type of Accommodation	● 'P' – Private ● 'N' – Non-Private (Charity/Service)
pAdmissionDiagnosis	String(500)	Admission Diagnosis	Can be multiple lines
pDischargeDiagnosis	String(500)	Discharge Diagnosis	
pICDCode	String(15)	ICD 10 Code of the illness	*Refer to ICD10 library*
pRelatedProcedure	String(150)	Related Procedure	Any value consisting of : 'A' to 'Z', 'Ñ'. Can include a space in between characters
pRVSCode	String(6)	Relative Value Scale Code of the procedure/operation performed	*See RVS Library* Required for Operating Room, Surgeons and Anesthesiologist claims only
pProcedureDate	String(10)	Date of Procedure	Date Format should be : 'MM-DD-YYYY'
pLaterality	String(1)	Laterality	● 'L' – Left ● 'R' – Right ● 'B' – Both

			● 'N' – N/A
pSessionDate	String(10)	Inclusive Date of Session	Date Format should be : 'MM-DD-YYYY'
pCheckUpDate1	String(10)	1st Checkup Date for MCP Package	Date Format should be : 'MM-DD-YYYY' Required for prenatal claims under non-hospital facility
pCheckUpDate2	String(10)	2nd Checkup Date for MCP Package	
pCheckUpDate3	String(10)	3rd Checkup Date for MCP Package	
pCheckUpDate4	String(10)	4th Checkup Date for MCP Package	
pTBType	String(1)	Type of TB-Dots claim	● 'I' – Intensive Phase ● 'M' – Maintenance NOTE For 'I', admission and discharge dates should be the first and last days of treatment in the intensive phase respectively. For 'M', admission and discharge dates should be the first and last days of treatment in the maintenance phase respectively. Required for TB-DOTS claims only
pNTPCardNo	String(10)	NTP Card No	Required for TB-DOTS claims only
pDay0ARV	String(10)	Day 0 ARV (Anti Rabies Vaccine)	Date Format should be : 'MM-DD-YYYY' Required for Animal Bite Package claims only
pDay3ARV	String(10)	Day 3 ARV (Anti Rabies Vaccine)	
pDay7ARV	String(10)	Day 7 ARV (Anti Rabies Vaccine)	
pRIG	String(10)	RIG (Rabies Immunoglobulin)	
pABPOthers	String(10)	Other Date	
pABPSpecify	String(50)	Others (Specify)	Required for Animal Bite Package claims only
pEssentialNewbornCare	String(1)	Essential Newborn Care	Flag whether Yes (Y) or No (N) Required for Newborn Care Package
pNewbornHearingScreeningTest	String(1)	Newborn Hearing Screening Test	
pNewbornScreeningTest	String(1)	Newborn Screening Test	
pFilterCardNo	String(20)	Filter Card Number	Required when the pNewbornScreeningTest = 'Y'
pDrying	String(1)	Immediate drying of newborn	Flag whether Yes (Y) or No (Y) Required when the pEssentialNewbornCare = 'Y'
pSkinToSkin	String(1)	Early skin-to-skin contact	
pCordClamping	String(1)	Timely cord clamping	
pProphylaxis	String(1)	Eye prophylaxis	
pWeighing	String(1)	Weighing of the newborn	
pVitaminK	String(1)	Vitamin K administration	
pBCG	String(1)	BCG vaccination	
pNonSeparation	String(1)	Non-separation of mother/baby for early breastfeeding initiation	
pHepatitisB	String(1)	Hepatitis B vaccination	
pLaboratoryNumber	String(20)	Laboratory Number	Required for Outpatient HIV/AIDS Treatment Package
pDoctorAccreCode	String(12)	Doctor's Accreditation Number	Formatted as: '####-######-##'
pDoctorLastName	String(60)	Doctor's Complete Surname	Same as for the member.
pDoctorFirstName	String(60)	Doctor's Complete First name	
pDoctorMiddleName	String(60)	Doctor's Complete Middle name	
pDoctorSuffix	String(5)	Doctor's Suffix name	'JR', 'SR', 'III', …etc Suffixes can be blank
pWithCoPay	String(1)	Flag whether the professional fee is no co-pay or with co-pay	● 'Y' – With co-pay ● 'N' –Noco-pay
pDoctorCoPay	String(12)	Amount of co-pay	Formatted as: '#######.##' Required when pWithCoPay='Y'
pEnoughBenefits	String(1)	(Y) – if the PhilHealth benefit is enough to cover HCI and PF charges. No purchases of drugs/medicines, supplies, diagnostics, and co-pay for professional fees by the member/patient. (N) –if the benefit of the member/patient was completely consumed prior to co-pay OR the benefit of the member/patient is not completely consumed BUT with purchases/expenses for drugs/medicines, supplies, diagnostics and others.	Flag whether Yes(Y) or No(N)
pTotalHCIFees	String(12)	Total Health Care Institution Fees	Formatted as: '#######.##' Required whenpEnoughBenefits = 'Y'
pTotalProfFees	String(12)	Total Professional Fees	
pGrandTotal	String(12)	Grand Total is equal to the HCI and Prof Fees	
pTotalActualCharges	String(12)	Total Actual Charges	Formatted as: '#######.##'

pDiscount	String(12)	Amount after Application of Discount	Required whenpEnoughBenefits = 'N'
pPhilhealthBenefit	String(12)	PhilHealth Benefit	
pTotalAmount	String(12)	Amount after PhilHealth Deduction	
pMemberPatient	String(1)	Member/Patient	Flag whether Yes(Y) or No(N) if applicable
pHMO	String(1)	HMO	
pOthers	String(1)	Others (i.e., PCSO, Promissory note, etc.)	
pDrugsMedicinesSupplies	String(1)	(Y) –if there is a purchase/s for drugs/medicines and/or medical supplies bought by the patient/member within/outside the HCI during confinement (N) – None	Flag whether Yes(Y) or No(N)
pDMSTotalAmount	String(12)	Total Amount for Drugs, Medicines and Supplies	Formatted as: '#######.##' Required when pDMSTotalAmount = 'Y'
pExaminations	String(1)	(Y) – if there is a diagnostic/laboratory examinations paid for by the patient/member done within/outside the HCI during confinement (N) – None	Flag whether Yes(Y) or No(N)
pExamTotalAmount	String(12)	Total Amount for Diagnostic	Formatted as: '#######.##' Required when pExamTotalAmount = 'Y'
pCaseRateCode	String(6)	Case Rate Codes for All Case Rates	*See Case Rate Library*
pZBenefitCode	String(7)	Z-Benefit Codes	Standard Risk Acute Lymphocytic (lymphoblastic) Leukemia for Children ● Z0011 – 1st tranche ● Z0012 – 2nd tranche ● Z0013 – 3rd tranche Early Stage Breast Cancer (Stage 0 to III-A) ● Z0021 –1st tranche ● Z0022 –2nd tranche Low to Intermediate Risk Prostate Cancer Requiring Prostatectomy ● Z003 – full payment End Stage Renal Disease Eligible for Kidney Transplant (Low Risk) ● Z0041 –1st tranche ● Z0042 –2nd tranche Elective Surgery for Standard Risk Coronary Artery Bypass Graft (CABG) ● Z0051 – 1st tranche ● Z0052 – 2nd tranche Tetralogy of Fallot (TOF) ● Z0061 – 1st tranche ● Z0062 – 2nd tranche Ventricular Septal Defect(VSD) ● Z0071 – 1st tranche ● Z0072 – 2nd tranche Cervical Cancer Chemoradiation with Cobalt & Brachytherapy (Low

			Dose) or Primary Surgery for Stage IA1, IA2 – IIA1 ● Z0081 – 1st tranche ● Z0082 – 2nd tranche Cervical Cancer Chemoradiation with Linear Accelerator & Brachytherapy (High Dose) ● Z0091 – 1st tranche ● Z0092 – 2nd tranche
pChiefComplaint	String(200)	Chief Complaint or Reason for Admission	Text/Memo
pBriefHistory	String(2500)	Brief History of Present Illness (OB Score/OB History)	Text/Memo
pCourseWard	String(500)	Course in the Wards	Text/Memo
pPertinentFindings	String(500)	Pertinent Laboratory and Diagnostic Findings	Text/Memo
pPrenatalConsultation	String(10)	Initial Prenatal Consultation	Date Format should be : 'MM-DD-YYYY'
pMCPOrientation	String(1)	Orientation to MCP/Availment of Benefits	Flag whether Yes(Y) or No(N)
pExpectedDeliveryDate	String(10)	Expected data of delivery	Date Format should be : 'MM-DD-YYYY'
pVitalSigns	String(1)	Vital Signs are Normal	Flag whether Yes(Y) or No(N)
pPregnancyLowRisk	String(1)	Ascertain the Present. Pregnacy is low-risk	
pLMP	String(10)	Last Menstrual Period	Date Format should be : 'MM-DD-YYYY'
pMenarcheAge	String(2)	Age of Menarche	Must be an integer
pObstetricG	String(10)	Obstetric History	Text
pObstetricP	String(10)		Text
pObstetric_T	String(10)		Text
pObstetric_P	String(10)		Text
pObstetric_A	String(10)		Text
pObstetric_L	String(10)		Text
pMultiplePregnancy	String(1)	Multiple Pregnancy	Obstetric Risk Factors Flag whether Yes(Y) or No(N)
pOvarianCyst	String(1)	Ovarian Cyst	
pMyomaUteri	String(1)	Myoma Uteri	
pPlacentaPrevia	String(1)	Placenta Previa	
pMiscarriages	String(1)	History of 3 miscarriages	
pStillBirth	String(1)	History of stillbirth	
pPreEclampsia	String(1)	History of pre-eclampsia	
pEclampsia	String(1)	History of eclampsia	
pPrematureContraction	String(1)	Premature contraction	
pHypertension	String(1)	Hypertension	Medical/Surgcal risk factors Flag whether Yes(Y) or No(N)
pHeartDisease	String(1)	Heart Disease	
pDiabetes	String(1)	Diabetes	
pThyroidDisaster	String(1)	Thyroid Disorder	
pObesity	String(1)	Obesity	
pAsthma	String(1)	Moderate to severe asthma	
pEpilepsy	String(1)	Epilepsy	
pRenalDisease	String(1)	Renal Disease	
pBleedingDisorders	String(1)	Bleeding disorders	
pPreviousCS	String(1)	History of previous caesarian section	
pUrineMyomectomy	String(1)	History of uterine myomectomy	
pVisitDate	String(10)	Date of visit	Date Format should be : 'MM-DD-YYYY'
pAOGWeeks	String(3)	AOG in weeks	Text
pWeight	String(10)	Weight	Weight & Vital Signs
pCardiacRate	String(10)	Cardiac Rate	
pRespiratoryRate	String(10)	Respiratory Rate	
pBloodPressure	String(10)	Blood Pressure	
pTemperature	String(10)	Temperature	
pDeliveryDate	String(10)	Date of Delivery	Date Format should be : 'MM-DD-YYYY'
pDeliveryTime	String(10)	Time of Delivery	Time Format should be : 'HH:MM:SSAM/PM'
pObstetricIndex	String(50)	Obstetric Index	Text
pAOGLMP	String(50)	AOG by LMP	Text
pDeliveryManner	String(50)	Manner of Delivery	Text
pPresentation	String(50)	Presentation	Text
pFetalOutcome	String(50)	Fetal Outcome	Text

pSex	String(1)	Sex	● 'M' – Male ● 'F' –Female
pBirthWeight	String(10)	Birth Weight (gram)	Must be an integer
pAPGARScore	String(10)	APGAR Score	
pPostpartum	String(10)	Scheduled Posparum follow-up consultation 1 week after delivery	Date Format should be : 'MM-DD-YYYY'
pPerinealWoundCare	String(1)	Perineal wound care	Flag whether Yes(Y) or No(N)
pPerinealRemarks	String(100)	Remarks	Text/Memo
pMaternalComplications	String(1)	Signs of Maternal Postparum Complications	Flag whether Yes(Y) or No(N)
pMaternalRemarks	String(100)	Remarks	Text/Memo
pBreastFeeding	String(1)	Breastfeeding and Nutrition	Flag whether Yes(Y) or No(N)
pBreastFeedingRemarks	String(100)	Remarks	Text/Memo
pFamilyPlanning	String(1)	Family Planning	Flag whether Yes(Y) or No(N)
pFamilyPlanningRemarks	String(100)	Remarks	Text/Memo
pPlanningService	String(1)	Provided family planning service to patient (as requested by patient)	Flag whether Yes(Y) or No(N)
pPlanningServiceRemarks	String(100)	Remarks	Text/Memo
pSurgicalSterilization	String(1)	Referred to partner physician for Voluntary Surgical Sterilization (as requested by patient)	Flag whether Yes(Y) or No(N)
pSterilizationRemarks	String(100)	Remarks	Text/Memo
pFollowupSchedule	String(1)	Schedule the next postpartum follow-up	Flag whether Yes(Y) or No(N)
pFollowupScheduleRemarks		Remarks	Text/Memo
pReferredReason	String(500)	Reason for referral from other HCI	Text/Memo
pIntensive	String(1)	Completed Intensive Phase	Flag whether Yes(Y) or No(N)
pMaintenance	String(1)	Completed Maintenance Phase	
pCriteria	String(200)	Clinical Criteria and/or Laboraties/Diagnositics Criteria	Text/Memo
pBP	String(20)	Blood Pressure	Text
pCR	String(20)		Text
pRR	String(20)	Respiratory Rate	Text
pTemp	String(20)	Temperature	Text
pHEENT	String(20)	Head, Ears, Eyes, Nose and Throat	Text
pChestLungs	String(20)	Chest/Lungs	Text
pCVS	String(20)		Text
pAbdomen	String(20)	Abdomen	Text
pGUIE	String(20)		Text
pSkinExtremities	String(20)	Skin Extremities	Text
pNeuroExam	String(20)	Neuro Examination	Text
pCourseDate	String(10)	Date in the Course in the Ward	Date Format should be : 'MM-DD-YYYY'
pFindings	String(200)	Pertinent PE/Lab Findings	Text/Memo
pAction	String(200)	Doctors Order/Action	Text/Memo
pPurchaseDate	String(10)	Date of Purchased	Date Format should be : 'MM-DD-YYYY'
pDrugCode	String(20)	Hospital Assigned Drug Code	Can be any format
pPNDFCode	String(20)	PNDF Code (Blank until PNDF lib is available)	Can be Blank for now
pGenericName	String(50)	Generic Name of Medicines/Drugs taken	Must not be blank
pBrandName	String(50)	Brand Name of Medicines/Drugs taken	Must not be blank
pPreparation	String(30)	Dose/ Cap/ Syrup/ Injectible/ Tab with ml/mg/gm content	Must not be blank
pQuantity	String(10)	Unit quantity of item	Integer format
pDiagnosticDate	String(10)	Date of Diagnostic	Date Format should be : 'MM-DD-YYYY'
pDiagnosticType	String(20)	Type of diagnostic/test done	● 'IMAGING' ● 'LABORATORY' ● 'SUPPLIES' ● 'OTHERS'
pDiagnosticName	String(50)	Name of Imaging procedure for Imaging, Name of Laboratory procedure for Laboratory, Name of Supplies for Supplies or Others for Supplies and Others	Must not be blank
pCompanyName	String(100)	Company's Name	Text
pCompanyTIN	String(15)	Company's TIN	Formatted as: '###-###-###-###'

pBIRPermitNumber	String(20)	BIR Permit Number	
pReceiptNumber	String(20)	Official Receipt Number	
pReceiptDate	String(12)	Official Receipt Date	Date Format should be : 'MM-DD-YYYY'
pVATExemptSale	String(10)	VAT Exempt Sale	Formatted as: '#######.##'
pVAT	String(10)	VAT – 12%	
pTotal	String(10)	Total Amount in the Receipt	
pUnitPrice	String(10)	Unit price of item	Formatted as: '#######.##'
pDescription	String(100)	Item Description	Text
pAmount	String(10)	Total Amount of the Specific Item	Formatted as: '#######.##'
pDocumentType	String(3)	Document to support the claim	*See Document Library*
pDocumentURL	String(250)	URL of the document accessible via https.	The document must first be encrypted using philhealth public key before publishing online. Please see the Annex for the guidelines for encryption.
pHospitalTransmittalNo	String(20)	Hospital Generated Transmittal Number	
pTransmissionControlNumber	String(18)	Philhealth Generated Transmittal file control number.	Will be blank if the transmission is failed
pErrCode	String(3)	Claim file error Code	
pErrDescription	String(100)	Claim file error Description	
pReceiptTicketNumber	String(18)	Philhealth Generated Upload Comfirmation Receipt ticket number	
pTransmissionDate	String(10)		Date Format should be : 'MM-DD-YYYY'
pTransmissionTime	String(10)		TIME Format should be : 'HH:MM:SSAM/PM'
pReceivedDate	String(10)	Date when the transmitted file received by PhilHealth	Date Format should be : 'MM-DD-YYYY'
pHasAttachedSOA	String(1)	Type of Accommodation	● 'Y' – With attached SOA ● 'N' –Without attached SOA
pDateSigned	String(10)	Date signed	Date Format should be : 'MM-DD-YYYY'
pRelCode	String(1)	Code indicating the relation of the representative who signed the consent to access patient record	Valid values: ● S = Spouse ● C = Child ● P = Parent ● I = Sibling ● O = Others (pRelDesc should have a value)
pRelDesc	String(50)	Specified relation of the representative who signed the consent on behalf of the patient to access patient records. This include relation not included in the list of defined values of relation of pRelCode	
pReasonCode	String(1)	Code indicating the reason why a representative signed the consent to access patient record	Valid values: ● I = Patient is incapacitated ● O = Patient is incapacitated (pReasonDesc should have a value)
pReasonDesc	String(50)	Text indicating reason why a reason why a representative signed the consent to access patient record	
pPreAuthDate	String(10)	Z-Benefit pre-authorization date	Date Format should be : 'MM-DD-YYYY'
pThumbmarkedBy	String(1)	Indicates whether the thumbmark is from the member/patient or from the representative of the member	Valid values: ● P = of the patient/member ● R = of a representative
pServiceProvider	String(50)	The acronym or short name of the company/institution that provided the system used in submitting the e-claims XML file	
pCertificateId	String(50)	The software validation certificate number issued upon passing the software compliance testing	

Annex D
Data Dictionary ValidateEsoa

Element	Name	Length	Description	Valid Values
eSOA	pHciPan	varchar2 (9)	The accreditation number issued by PhilHeath to the health facility	
	pTransmittalId	varchar2 (50)	A UNIQUE reference number assigned to the claim by the submitting health facility	
Summary of Fees	pChargesNetOf ApplicableVat	Number(10,2)	Refed to the total charges net of applicable vat	0.00 to 99999999.99
	pSeniorCitizenD iscount	Number(10,2)	The amount of senior citizen (SC) discount, if applicable	0.00 to 99999999.99
	pPWDDIscount	Number(10,2)	The amount of Person with Disability (PWD) discount, if applicable	0.00 to 99999999.99
	pPCSO	Number(10,2)	The amount of Philippine Charity Sweepstakes Office (PCSO) discount, if applicable	0.00 to 99999999.99
	pDSWD	Number(10,2)	The amount of Department of Social Welfare and Development (DSWD) discount, if applicable	0.00 to 99999999.99
	pDOHMAP	Number(10,2)	The amount of Department of Health (DOHMAP) discount, if applicable	0.00 to 99999999.99
	pHMO	Number(10,2)	The amount of HMO discount, if applicable	0.00 to 99999999.99
	pDescription	Varchar (10,2)	The description of other funding source, if applicable	
	pAmount	Number (10,2)	The amount of other funding source discount, if applicable	0.00 to 99999999.99

Element	Name	Length	Description	Valid Values
eSOA	pHciPan	varchar2 (9)	The accreditation number issued by PhilHeath to the health facility	
	pTransmittalId	varchar2 (50)	A UNIQUE reference number assigned to the claim by the submitting health facility	
Other Funding Source	pDescription	300		
	pAmount			
PhilHealth	pTotalCaseRate Amount	Number (8,2)	The amount of case rate package being claimed	0.00 to 999999.99
Balance	pAmount	Number (10,2)	The difference between the total amount of charges, net of applicable vat, for the room and board, drugs and medicine, laboratory and diagnostic, operating room and medical supplies charges, less applicable discounts and the case rate amount.	0.00 to 99999999.99
Professional Information	pPAN	Varchar (14)	The corresponding PhilHealth Accreditation Number (PAN) of the attending Physician	
	pFirstName	Varchar (60)	The first name of the attending physician	
	pMiddleName	varchar	The middle name of the attending physician	
	pLastName	Varchar (60)	The last name of the attending physician	
	pSuffixName	Varchar (60)	The suffix or extension name of the attending physician (ex. Jr., Sr. II, IIII and so on)	
Itemized Billing	pServiceDate	Date	the date of service for room and board, laboratory and diagnostic service and operating room fees; the date of issuance for drugs and medicine and medical supplies	mm-dd-yyyy
	pItemCode	Varchar (25)	Refer to the code of the item in the electronic Statement of Account (eSOA)	as listed in eSOA Item Library
	pItemName	Varchar (300)	Refer to the description of an item in the electronic Statement of Account (eSOA)	as listed in eSOA and Item Library
	pUnitOfMeasurement	Varchar (50)	Refer to the unit of measurement used for an item in the electronic Statement of Account (eSOA)	as listed in eSOA and PhilHealth Drug Library
	pUnitPrice	Number (8,2)	Refer to the unit amount of an item in the electronic Statement of Account (eSOA)	0.00 to 999999.99
	pQuantity	Number (4,0)	Refer to the total number or quantity of an item in the electronic Statement of Account (eSOA)	0 to 9999
	pTotalAmount	Number (11,2)	Refer to the total amount of of an item in the electronic Statement of	0.00 to 999999999.99

Element	Name	Length	Description	Valid Values
eSOA	pHciPan	varchar2 (9)	The accreditation number issued by PhilHeath to the health facility	
	pTransmittalId	varchar2 (50)	A UNIQUE reference number assigned to the claim by the submitting health facility	
			Account (eSOA) i.e pTotalAmount = pQuantity x pUnitPrice)	
	pCategory	Varchar (50)	Refer to eSOA category	as listed in eSOA and Category Library

Annex E

Data Dictionary ValidateCF5

Name	Length	Description	Valid Values
Name	**Type (Length)**	**Description**	**Valid Values**
Table : DRG Information			
PDX_Code	Varchar2 (10)	Primary Diagnosis Code (ICD-10)	
Nb_TOB	Varchar2 (5)	Time of Birth of NewBorn(24-HR Format)	
Nb_Admweight	Numeric (2,2)	Admission of weight of New Born (24-hour format)	
Series	Varchar2 (13)	Claims series no.	
Lhio	Varchar2 (2)	Lhio code	
Table: DRG Procedure			
PROCID	Number(38,0)	DRG Procedure	
RVS	Varchar2(6)	Relative Scale Procedure Code	
Laterality	Varchar2(2)	Laterality	L,R,B
Ext1_Code	Numeric(1)	Extension Code 1	
Ext2_Code	Numeric(1)	Extension Code2	
ICD9_Code	Varchar2(6)	ICD9 CM Code	
Series	Varchar2 (13)	Claims series no.	
Lhio	Varchar2 (2)	Lhio code	

Annex F - Guidelines for the Encryption of Image Files of the supporting documents of claims Submitted through e-Claims web servce

PROCEDURES

1. The supporting documents will be scanned and converted to PDF/A format. The scanned image files should be saved temporarily to a directory and be deleted after they have been encrypted.
2. Each file will be encrypted individually. Encryption will be done on the side of HCI. PhilHealth will not include file encryption in the e-Claims web service. This will avoid overhead of web service calls both on the side of HCI and PhilHealth.
3. Before encrypting the file, compute for the hash of file using SHA-256 hash method. (This hash will be compared to the hash that will be computed later after the file has been extracted on from decrypted file on PhilHealth side. If the hash values match, it assures that the file has been encrypted and decrypted correctly.)
4. As recommended by online articles about encrypting files, two different encryption methods will be used. The image file itself will be encrypted using AES-256-CBC. To be able to decrypt the file, PhilHealth needs to have the password used in the encryption. The password will be sent together with the encrypted data. But the password will be themselves encrypted using public key encryption. PhilHealth will provide a separate digital certificate or file containing the public key that will be used in encrypting the password.
5. HCIs/SPs may use their preferred programming language or tools in doing the encryption.
6. The password to be used should be composed of 32 random bytes of data. Two arrays of random 16 bytes long will be used. On calling the AES encryption, these two arrays will be merged or concatenated but the two "keys" will be encrypted separately.
7. For the initialization vector (IV) parameter for the AES, an array of random 16 bytes will be used. This IV will also be encrypted using public key encryption and then sent to PhilHealth.
8. The resulting encrypted document will be saved using the original file name appended with ".enc". Other information about the file such as the MIME type of the image file will also be included in the

file. The encrypted data elements will be encoded in the file using Base64 encoding. All data elements will be included in a single file in JSON format.

FORMAT OF THE OUTPUT FILE

```
{
"docMimeType": "<MIME type of the image file e.g. 'application/pdf >",
"hash": "<SHA-256 hash of the image file before encryption>",
"key1" : "<Encoded in Base64 format, the public key encrypted first 16 bytes of the password>",
"key2" : "<Encoded in Base64 format, the public key encrypted second 16 bytes of the password>",
"iv" : "<Encoded in Base64 format, the public key encrypted initialization vector>",
"doc" : "<Encoded in Base64 format, AES-256-CBC encrypted image file data>
}
```

Figure 1: Output File Format/Layout

The figure above shows the format of the output file. As it can be observed, the output file is basically a representation in JSON format of the four encrypted data elements and 2 unencrypted data elements. The figure shows the description and format of the data elements.

The soft copy of this guide comes with two demo *kits* with each demo kit packaged in one zip file. The file *Demo Kit for PHP.zip* contains the demo kit for PHP. It contains PHP source codes which can be used to setup a demo PHP web application. The file *Demo Kit for C#.zip* contains the demo kit for kit for C#. It contains Visual Studio solution and project files which can be used to generate a demo Windows Forms application. Although the demo kits uses PHP and C#, HCI/SPs can use their preferred programming languages. The demo kits are provided to serve just as a guide.
Both demo kits contains a folder named "@Files". It has the following folders and files:

@Files	
Input	
pnpki_philhealth_eclaims_auth_cert.pem	The public key file generated from the sample digital certificate. This file should replace the previous public key file name `sample_public_key.pem` which was generated by another certificate authority.
SAMPLE_BIRTH_CERTIFICATE.pdf	Can be passed as the target PDF file for encryption. Note the demo app will delete the target PDF file after encryption.
SAMPLE_BIRTH_CERTIFICATE.orig.pdf	Original copy of the sample PDF file
Recreate file to be encrypted.bat	A batch file that can be used to recreate the `SAMPLE_BIRTH_CERTIFICATE.pdf` from `SAMPLE_BIRTH_CERTIFICATE.orig.pdf` after the demo app has deleted it
Output	
SAMPLE_BIRTH_CERTIFICATE-usingCSharp.pdf.enc	Sample generated encrypted e-claims document

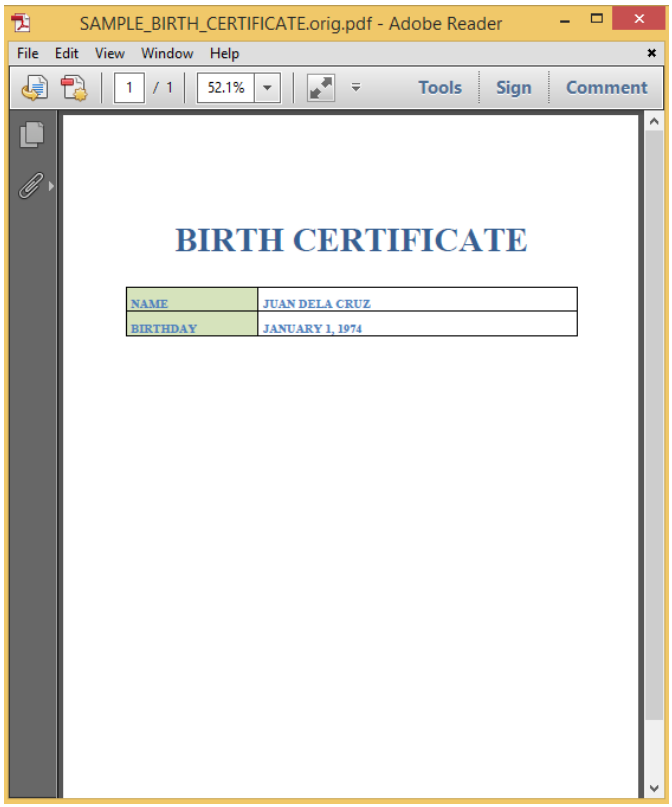

Figure 2 Sample PDF file (SAMPLE_BIRTH_CERTIFICATE.orig.pdf)

```
Bag Attributes
    localKeyID: 01 00 00 00
subject=/CN=eclaims-test.philhealth.gov.ph/serialNumber=100000000A/OU=TMD/O=Philhealth/C=PH
issuer=/CN=Gov Authentication Test CA/O=DOST/C=PH
-----BEGIN CERTIFICATE-----
MIIFqjCCA5KgAwIBAgIIfGK6fQx9D6owDQYJKoZIhvcNAQELBQAwQTEjMCEGA1UE
AwwaR292IEF1dGhlbnRpY2F0aW9uIFR1c3QgQ0ExDTALBgNVBAoMBERPU1QxCzAJ
BgNVBAYTAlBIMB4XDTE0MDkzMDA4MjUwNloXDTE0MTIyOTA4MjUwNlowbjEnMCUG
A1UEAwweZWNsYWltcy10ZXN0LnBoaWxoZWFsdGguZ292LnBoMRMwEQYDVQQFEwox
MDAwMDAwMDBBMQwwCgYDVQQLDANUTUQxEzARBgNVBAoMClBoaWxoZWFsdGgxCzAJ
BgNVBAYTAlBIMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAk2a1LqSv
Tmz6/bdE/zxYjOIkzvlFsXtw4Km/Kcf+Cgym+br65MMms2GTff5PauiW0Z5+CvGE
ata06Ge4SW8KB5XexOo5E5gQH7iFKkD34Rj6P37HcfLE4kXLV5DNgfLnqBRE35rZ
ZqcRfz+2p8Ou3aEA7xq9irgA7F7Qf3iYuejBVjp0v8TLpOzV2XjhrozbMciGWdAj
4JG0KJpu5Kf3t54kGHQscs5c6ieDcq4hXTzIRqozE2hhqgPgWeCPBsWP9eU5+NQ8
Xv3Ad04PaaJ5/+SQH+bS4wkOn9m+1tFgn3v12XVRrEajgXMRBkzM87YbZVvagxjn
zDXWpbGBcaehHQIDAQABo4IBdzCCAXMwPQYIKwYBBQUHAQEEMTAvMC0GCCsGAQUF
BzABhiFodHRwOi8vZ292dGVzdGNhLm5wa2kuZ292LnBoL29jc3AwHQYDVR0OBBYE
FP0C6sET9CCfWrxhFMznC1dhlcg/MAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAU
bpWB/iCpw3sUJEyobjQeCtsU/7wwUAYDVR0gBEkwRzBFBgdghGABBAEBMDowOAYI
KwYBBQUHAgEWLGh0dHA6Ly9nb3Z0ZXN0Y2EubnBraS5nb3YucGgvQ1AvR292QXV0
aC5odG1sMD0GA1UdHwQ2MDQwMqAwoC6GLGh0dHA6Ly9nb3Z0ZXN0Y2EubnBraS5n
b3YucGgvY3JsL0dvdkF1dGguY3JsMA4GA1UdDwEB/wQEAwIEsDAdBgNVHSUEFjAU
BggrBgEFBQcDAgYIKwYBBQUHAwQwJAYDVR0RBB0wG4EZbmV0d29ya0BwaGlsaGVh
bHRoLmdvdi5waDANBgkqhkiG9w0BAQsFAAOCAgEAOrq5LdnUvmMTfBpKk2vT9VV0
jVUPVlGqMmkhhyucvaRdDevBqgbKoeUAuGotCBVavE/1m1bsytNVhDgX73AVobaF
rFzkfyX7x8vNQXqdG8GhgDYZOEnE84eYn24eAJw0A4NH2s62L1/SKf6pk01ouPXa
PQuyMPnBuUlv9I7hd1+9sMmkxujEMUw8h5SlUvIsrjpop0+Bdx69GfDgaIXYjLYT
47bvae/zrPRZVbZw6qhofUFLM0t5hHLqE3Bd6+uIoCxtqI9YBZ9ciiU/wjZhmlOV
G+dysOF5MfAKbj7ekPXVaQNLP1oJLB0/x4alpUUkfuDUnOoaW5mNCK36iTp1Mheb
0u6KBqbc0cNMbNWKV6Ed/woV5G57CbPLQbz5/Hb+pofdcOUISc33YZcM6c6tMdAW
dzvDEKMavNOqtHOyZGljy1UlyqaQAN0mNOYOc22vMiaN39opcjLKrVzjlaDXjxtU
4Pw7nCKBPOIXm4JSDIjE7YDK2n3dBTr6BJ+GVT283PGAY0ka8KvunlX9sL+1bSII
rgcCsK3aLyGJ6mfvFxsen8z3iCMy6AIbk8IcM9TizEXIjQx/i9KT4CVBlM5LzKYl
ycszRL27b5q5kIwrgQFYcsj86Zr7RcJX9B2cA546qlGHWgrIRUasYs7xeJKcHnmH
BUQRiOva/E6PLFfxpI0=
-----END CERTIFICATE-----
```

Figure 3 Content of the test public key file (pnpki_philhealth_eclaims_auth_cert.pem)

DEMO KIT FOR PHP

The contents of the file *Demo Kit for PHP.zip* are shown below. The folder `EclaimsDocEncryption` can just be copied to the root or any directory of a PHP installation to be able to setup the demo application. Please enable the openssl extension to run the demo application. The file `encryptEClaimsDoc.php` is the entry point of the demo application. Please see below the screenshot of the PHP demo application.

EclaimsEncryption	
@Files	The contents of this folder has been described on previous page
encryptEClaimsDoc.php	Provides a sample user interface for passing gathering inputs parameters in encrypting scanned image files of the supporting documents of an electronically-submitted claim
testEncryptAndDecryptXml.php	Provides a demo application for the encryption and decryption of the input and output parameters that are to be passed as XML text
PhilHealthEClaimsEncryptor.php	Contains the class implementing the encryption procedures

Figure 4: Screenshot of the PHP demo application

Figure 5: Output of PhilHealth's test application after decrypting the file encrypted by the PHP demo application

DEMO APPLICATION FOR C#

The contents of the file *Demo Kit for C#.zip* are shown below. It contains a Visual Studio solution/project files. Open the solution file and build the `PhilHealthEClaimsEncryptionDemoApp` project to generate the executable file of the demo Windows Forms application file. See the screenshot of the C# demo application.

PhilHealthEClaimsEncryptionDemoApp	

@Files	The contents of this folder has been described on previous section
bin	
Helpers	Contains one helper/utility class
Obj	
Properties	
App.config	
Form1.cs	
Form1.Designer.cs	
Form1.resx	
PhilHealthEClaimsDocEncryption.cs	A class that implements the guide for encrypting a e-claims document
PhilHealthEClaimsEncryptionDemoApp.csproj	The C# project file
Program.cs	
EClaimsDocEncryption.sln	Visual Studio solution file
EClaimsDocEncryption.v12.suo	

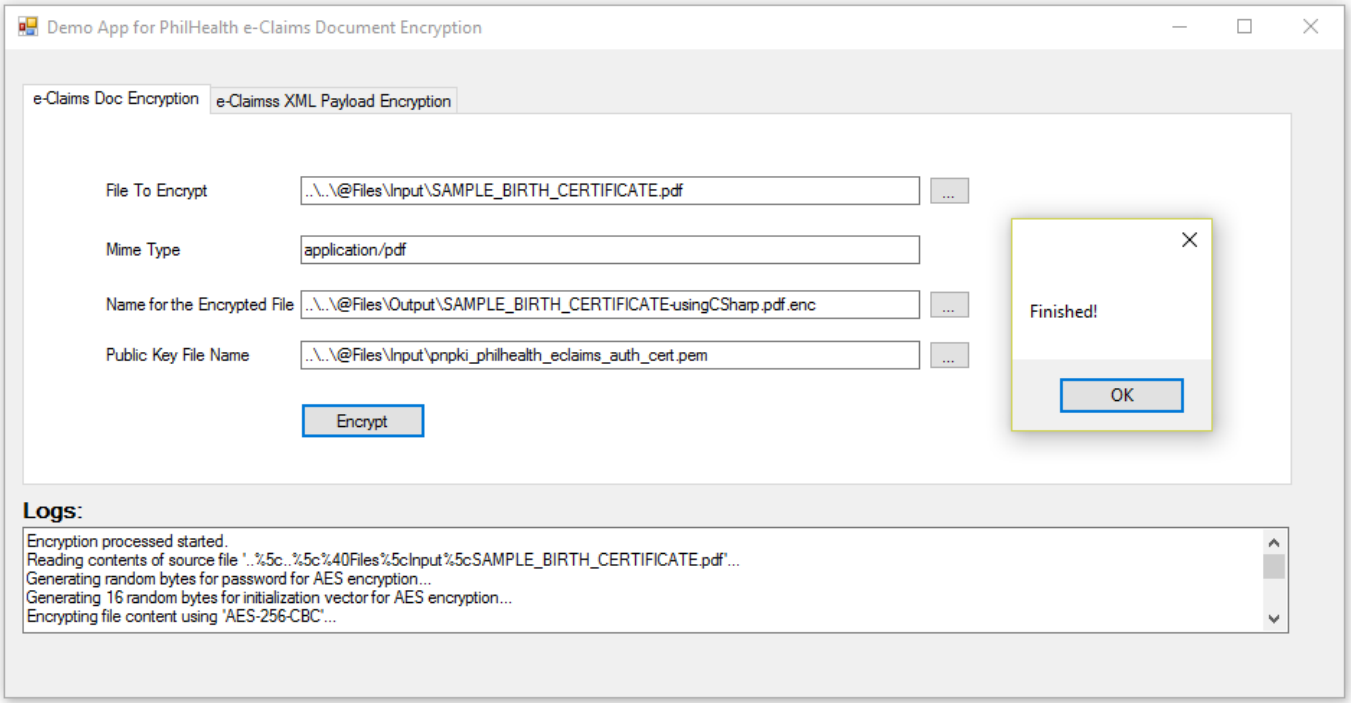

Figure 6 Screenshot of the C# demo application

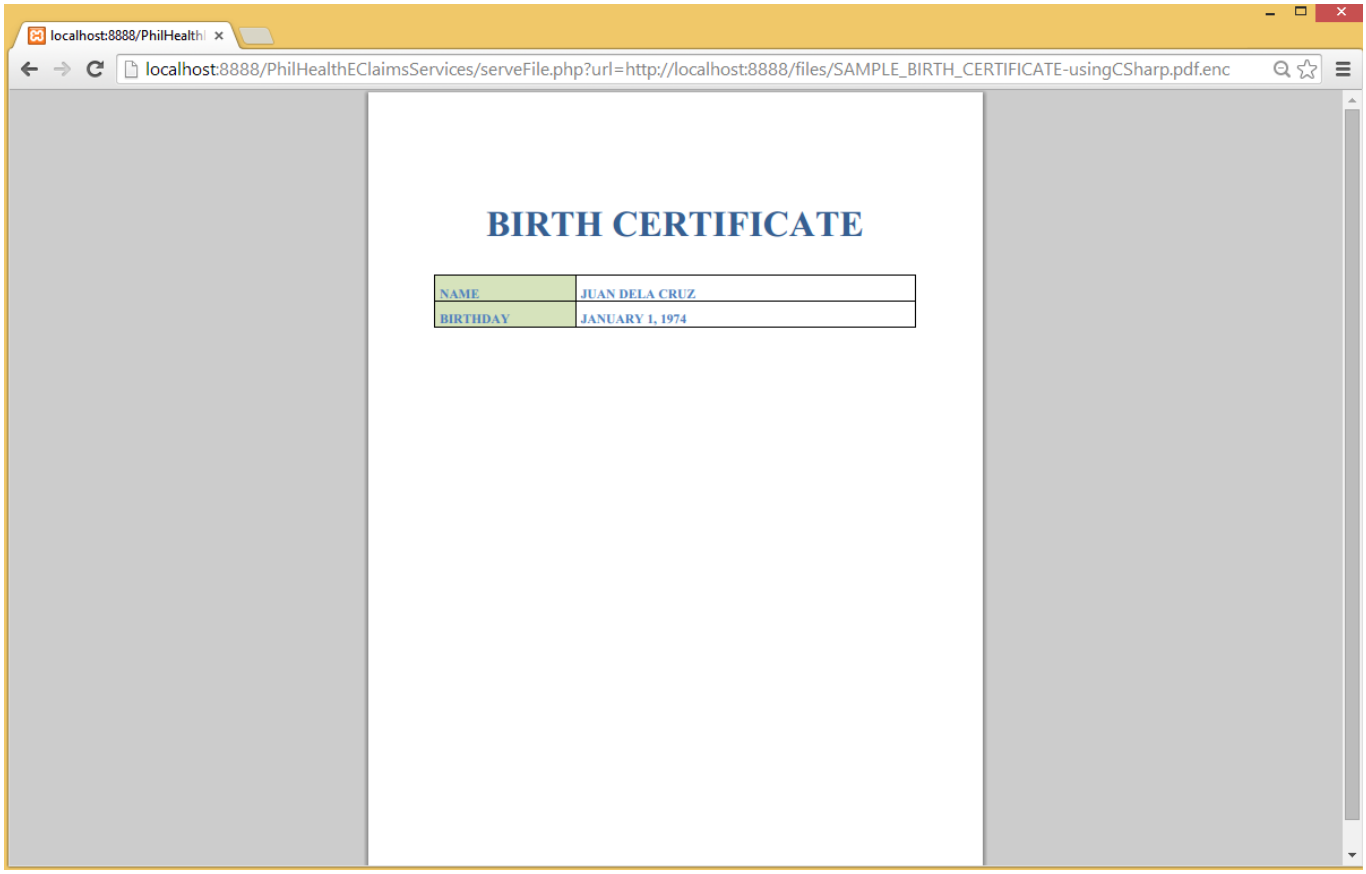

Figure 7 Output of PhilHealth's test application after decrypting the file encrypted by the C# demo application

Annex G: Guidelines for the Encryption and Decryption of Parameters that are to be passed as xml

GUIDELINES AND PROCEDURES

- The application of HCI should encrypt input parameters that are supposed to be passed as XML. HCI applications should also decrypt all the output parameters that are supposed to be returned by PECWS as XML.
- The input and output parameters that are to be passed as XML text will be referred in this section as XML payload data.

ENCRYPTION OF THE XML PAYLOAD DATA

1. The procedures for the encryption of the XML payload data will be similar to the procedures in the encryption of the image files of the supporting documents of electronically submitted claims. The difference is that encryption of the XML payload data will not use public-key encryption.
2. The XML payload data will be encrypted using "AES-256-CBC" algorithm.
3. PhilHealth will issue the passphrase/password/cipher key to be used by each HCI. The password should be hashed using SHA-256 algorithm. The first 32 bytes of the hashed value should be passed as the password for the AES encryption. If the resulting hashed value is less than 32-bytes, the value should be padded with '0x00' or null character.
4. An array of random 16 bytes should be used as the initialization vector.
5. Before encrypting the data, compute for the hash of file using SHA-256 hash method. (This hash will be compared to the hash that will be computed later after the file has been extracted on from decrypted file on PhilHealth side. If the hash values match, it assures that the file has been encrypted
6. The expected output of these encryption procedures is to be encoded in JSON. The figure below shows the expected key-value pairs that are to be included in the JSON text.
    a. The "docMimeType" key, should have fixed value of "<text/xml>".
    b. The value for "key1" should be an empty string.
    c. The value for "key2" should also be set to empty string.
    d. The value for "iv" key should be the base-64 encoded value of the initialization vector mentioned in Step#6.
    e. The value for the "doc" key should the base-64 encoded value of the result of the AES-256 encryption of the XML payload data.

```
{
"docMimeType": "<text/xml>",
"hash": "<SHA-256 hash of the image file before encryption>",
"key1" : "",
"key2" : "",
"iv" : "<Encoded in Base64 format, the public key encrypted initialization vector>",
"doc" : "<Encoded in Base64 format, AES-256-CBC encrypted image file data>
}
```

Figure 1: Format/Layout of encrypted data

7. SPs/HCIs may use their preferred programming language or tools in doing the encryption and decryption. The PECWS development kit includes sample codes in C# and PHP. For the sample source codes in PHP, the `encryptXmlPayloadData` method of the `PhilHealthEClaimsEncryptor` class contains an implementation of the encryption procedures outlined above.

DECRYPTION OF THE XML PAYLOAD DATA

1. The decryption process is just the reverse of the encryption process outlined in previous section.
2. For the service methods that are supposed to return an XML, PECWS will encrypt the supposed XML data and return a text encoded in JSON whose key-value pairs as described in the previous section.
3. After receiving the JSON text, the HCI application should parse it.
4. Get the value of the "iv" key. That is the base-64 encoded value of the initialization vector used in the AES-256 encryption of the XML payload data. Decode the value of the initialization vector from base-64 encoding to its corresponding raw array of bytes.
5. Get the password, passphrase or cipher key of the HCI. Get the SHA-256 hash value of that value. . If the length in bytes of the resulting hashed value is less than 32, pad the value with '\0" or null character.
6. Get the value of the "doc" key. That value is the base-64 encoding of the encrypted data. Decode the data from base-64 encoding to the raw array of bytes of the encrypted data, With values of the passphrase, initialization vector and actual encrypted data, decrypt the data using AES-256-CBC.
7. You may get the hash value of the decrypted data and compare it with the value of the "hash" key.
8. After the decryption, the application of HCI may now process and parse the target XML data.
9. SPs/HCIs may use their preferred programming language or tools in doing the encryption and decryption. The PECWS development kit includes sample codes in C# and PHP. For the sample source codes in PHP, the `decryptPayloadDataToXmlmethod` of the `PhilHealthEClaimsEncryptor` class contains an implementation of the decryption procedures outlined above.

Annex H -References

Reference	File Name
eSOA	eSOA Library.xlsx
eSOA	eSOA.dtd
CF5	CF5.dtd
SSVTF	SSVTF PeCWS S3.0(Revised 2024-05-30).docx