NYeC

New York eHealth  
Collaborative

**SHIN-NY API “*Hack-a-thon”*  
Developer’s Guide  
  
14-15 June 2013**  
  
**DOC-0267**

This document is provided by the New York eHealth Collaborative, (NYeC), in draft form solely for informational purposes. “NYeC” means New York eHealth Collaborative, Inc., a New York not-for-profit corporation with its principal place of business at 40 Worth St., Fifth floor, New York, NY 10013, and its officers, directors, employees, subcontractors and agents. This document is confidential, for your private use only, and may not be shared with others. This document is designed to serve as a general summary of the services NYeC may offer from time-to-time. The descriptions in this document are not a guarantee that any particular service will be available for your use, nor are the descriptions a guarantee that any service will perform in accordance with such descriptions or help you to achieve particular results. NYeC makes no representation, warranty or guarantee, expressed or implied, concerning this document or its contents. The information in this document is subject to change at any time, and NYeC has no duty to provide you with notice of such changes. In addition, NYeC will not be responsible or liable for any losses or damages, including without limitation direct, indirect, or consequential damages relating to or arising from your reliance upon any part of this document. Moreover, this document does not constitute an offer or agreement, or a solicitation of an offer or agreement, to enter into any particular relationship with NYeC. Before determining to use any service offered by NYeC, you should consult with your independent advisors to review and consider any associated risks and consequences. © 2013, New York eHealth Collaborative, Inc.

Table of Contents

[SHIN-NY API *Quick Reference Guide* 3](#_Toc358974273)

[Important CCD Concepts 4](#_Toc358974274)

[Key Terms 4](#_Toc358974275)

[1 Example of SOAP Message (ExecuteProprietaryDoc) 5](#_Toc358974276)

[2 SHIN-NY API Calls (Clinical Data Methods) 6](#_Toc358974277)

[2.1 Method: ExecuteXCPD (Step #1) 6](#_Toc358974278)

[2.1.1 Endpoint 6](#_Toc358974279)

[2.1.2 Input: XCPD Request Message 6](#_Toc358974280)

[2.1.3 Output #1: XCPD Request Response (Containing Patient EMPI) 7](#_Toc358974281)

[2.1.4 Output #2: Response to Unsuccessful XCPD Request: Internal Error 8](#_Toc358974282)

[2.1.5 Output #3: Response to Unsuccessful XCPD Request: No Exact Match Found 9](#_Toc358974283)

[2.1.6 Output #4: Response to Unsuccessful XCPD Request: No Match Found 10](#_Toc358974284)

[2.2 Method: ExecuteXCAQuery (Step #2) 11](#_Toc358974285)

[2.2.1 Endpoint 11](#_Toc358974286)

[2.2.2 Input: EMPI 11](#_Toc358974287)

[2.2.3 Output: ExecuteXCAQuery Response XML: XDSDocument Entry.uniqueId 11](#_Toc358974288)

[2.2.4 Output: ExecuteXCAQuery Response XML: RepositoryUniqueId 11](#_Toc358974289)

[2.3 Method: ExecuteXCARetrieve (Step #3) 12](#_Toc358974290)

[2.3.1 Endpoint 12](#_Toc358974291)

[2.3.2 Input: ExecuteXCARetrieve request for a known patient 12](#_Toc358974292)

[2.3.3 Output: ExecuteXCARetrieve Response XML 12](#_Toc358974293)

[2.4 Method: ExecuteProprietaryDoc (Alternative to Steps 1-3) 12](#_Toc358974294)

[2.4.1 Endpoint 12](#_Toc358974295)

[2.4.2 Output: ExecuteProprietaryDoc XML 12](#_Toc358974296)

|  |  |  |
| --- | --- | --- |
| SHIN-NY API *Quick Reference Guide* | **What is the SHIN-NY API?**   * A set of web-based functions that developers can use to integrate SHIN-NY clinical data into their applications. | **What are the prerequisites?**   * Know about SOAP * Know how to sign a request using an X.509 certificate to ID your application to the SHIN-NY. |
| **Security for Hack-a-thon—***For access to the SHIN-NY during the Hack-a-thon and, to satisfy the security requirements, you will be given:*   * **User ID** (i.e., clinical practitioner’s User ID) * **Facility OID** (i.e., clinical practitioner’s OID) * **X.509 Certificate** * **IDPToken** (i.e., JSON web token)   **Note:** No user password is needed because we are not authenticating users during the Hack-a-thon. The IDPToken is provided for this purpose.  *Security requirements are:*   1. A signature for the entire document.  Signature is generated using X.509.   **Application Verification***—The SOAP message must be digitally signed to verify the application. Following is an outline of the digital signature:*  <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  <soapenv:Header>  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"  xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"">  <wsse:BinarySecurityToken/>  <dsig:Signature>  <dsig:SignedInfo>  <dsig:DigestValue>6Zk3egHGsjN…<dsig:DigestValue>  </dsig:SignedInfo>  <dsig:SignatureValue>  mvTODcbUzFWrrfdwlyfHYagIRWx8Y....  </dsig:SignatureValue>  <dsig:KeyInfo>  <wsse:SecurityTokenReference>  <wsse:Reference />  </wsse:SecurityTokenReference>  </dsig:KeyInfo>   </dsig:Signature>  </wsse:Security>  </soapenv:Header>  <soapenv:Body/> </soapenv:Envelope>  For details, see *Example of SOAP Message (ExecuteProprietaryDoc)*  on page 5. | **Step #1—Call ExecuteXCPD—***First step in the retrieval of clinical data, determine EMPI number.*   1. Format an XCPD request from the provided [*template*](https://www.dropbox.com/sh/qxy1d4dpfwkolr3/oEBKxArpe9)replacing any @tokens including:  * livingSubjectName (name—first, last) * livingSubjectBirthTime (DOB—YYYYMMDD) * [livingSubjectAdministrativeGender] (M/F—optional value)  1. Submit the XCPD request to the **ExecuteXCPD** method along with these three parameters:  * **User ID** –SHIN-NY ID of the end user * **Facility OID** – Facility for the end user * **IDPToken**  1. From the response, extract the EMPI using the following XPATH: //patientid/id/@extension   **Template URL:** <https://www.dropbox.com/sh/qxy1d4dpfwkolr3/oEBKxArpe9>  **Step #2 Call ExecuteXCAQ—***Use the EMPI number to get a list of documents for the patient.*   1. Format an XCAQ request from the provided [*template*](https://www.dropbox.com/sh/qxy1d4dpfwkolr3/oEBKxArpe9)*.* 2. Set the XPATH: //AdhocQuery/rim:ValueList/rim:Value  to the following:   '<EMPI>^^^&amp;2.16.840.1.113883.3.2591.400.1&amp;ISO'  …replacing <*tokens*>  …So that the request will resemble:     1. Submit the XCAQ request to the **ExecuteXCAQ** method along with these three parameters:  * **User ID** –SHIN-NY ID of the end user * **Facility OID** – Facility for the end user * **IDPToken**  1. Extract the response document ID @ the XPATH:   //rim:ExternalIdentifier[@id='urn:uuid:172976ae-84f2-42d2-a1db-f86efc0ea3e4']/@value | **Step #3 ExecuteXCARetrieve—***Once you have the document ID you can retrieve the CCD.*   1. Format an XCAR request from the provided [*template*](https://www.dropbox.com/sh/qxy1d4dpfwkolr3/oEBKxArpe9)*.* 2. Set the following XPATHs:   //RetrieveDocumentSetRequest/DocumentRequest/HomeCommunityId =2.16.840.1.113883.3.2591.400.1  //RetrieveDocumentSetRequest/DocumentRequest/RepositoryUniqueId =2.16.840.1.113883.3.2591.400.1.1  //RetrieveDocumentSetRequest/DocumentRequest/DocumentUniqueId =<*Document ID from XCAQ*>   1. Submit the XCAR request to the **ExecuteXCAR** method along with these three parameters:  * **User ID** –SHIN-NY ID of the end user * **Facility OID** – Facility for the end user * **IDPToken**  1. Extract the BASE64 encoded string from the result using the XPATH:   //xsi:RetrieveDocumentSetResponse/xsi:DocumentResponse/xsi:Document   1. Convert the BASE64 string into a string for your platform. 2. Load the string into an XML parser and process the clinical data.   **Alternative to steps 1-3: call ExecuteProprietaryDoc—***If you are using only an MRN and Facility OID, you can request documents directly without all the IHE steps. For details, see Method: ExecuteProprietaryDoc on page 12.* |
| **Questions?**  Hackers can post questions via email ([shinnyapi-hackathon@googlegroups.com](mailto:shinnyapi-hackathon@googlegroups.com)) or by visiting the Google group (<https://groups.google.com/d/forum/shinnyapi-hackathon> ). | | |
| Important CCD Concepts **XPath to Demographics:**  /ClinicalDocument/recordTarget/patient Role/addr  **XPaths to document sections are formatted according to the following pattern:**  //ClinicalDocument/component/structuredBody/component/section/text/table[../../templateId/@root=’**<*SectionTemplateID*>**’]  **Use the following SectionTemplateIDs for each section:**  **Allergies**: 2.16.840.1.113883.3.88.11.83.102  **Medications**: 2.16.840.1.113883.3.88.11.83.112  **Vital Signs**: 2.16.840.1.113883.3.88.11.83.119  **Encounters**: 2.16.840.1.113883.3.88.11.83.127  **Results**: 2.16.840.1.113883.3.88.11.83.122  The element provided at the ‘text/table’ XPath is a valid HTML table and can be processed by row or presented directly to users.  Following is an example of the Vital Signs section rendered as HTML:    For a more detailed review of the data, configure your XPath to use ‘entry’ element, which is at the same level of ‘text’, as follows:  //ClinicalDocument/component/structuredBody/component/section/entry[../templateId/@root='@SectionTemplateID'] Key Terms **CCD—**Continuity of care document  **IHE—**Integrating the Healthcare Environment. This is a set of health information exchange standards, which includes the following:   * **XCPD**—Cross community patient discovery standard * **XCAQ**—Cross community document lookup standard * **XCAR**—Cross community document retrieval standard   **X.509 Certificate—**Used in software to: (1) verify the identity of a user and (2) send secure clinical data via encryption to the owner of the certificate.  **IDPToken—**Unique identifier of the application user that persists for the duration of a session with the SHIN-NY API.  **OID—**Object Identifier in the API context used to identify the healthcare facility, (e.g., Facility OID for patients registered is 1.2.3.4.99999.4) | | |

# Example of SOAP Message (ExecuteProprietaryDoc)

The following example of a complete SOAP request is a proprietary call that shows a:

* **Digital Signature**—comprising both a **signature value** and a **security key identifier**, which, together, authenticate the calling application, (i.e., sender of the message).   
    
  The application is authenticated every time it calls the API. After the application is authenticated, the API “believes” everything the application tells it.
* **IDP Token**—which identifies the application user, (i.e., clinician). Users are authenticated only once per session.

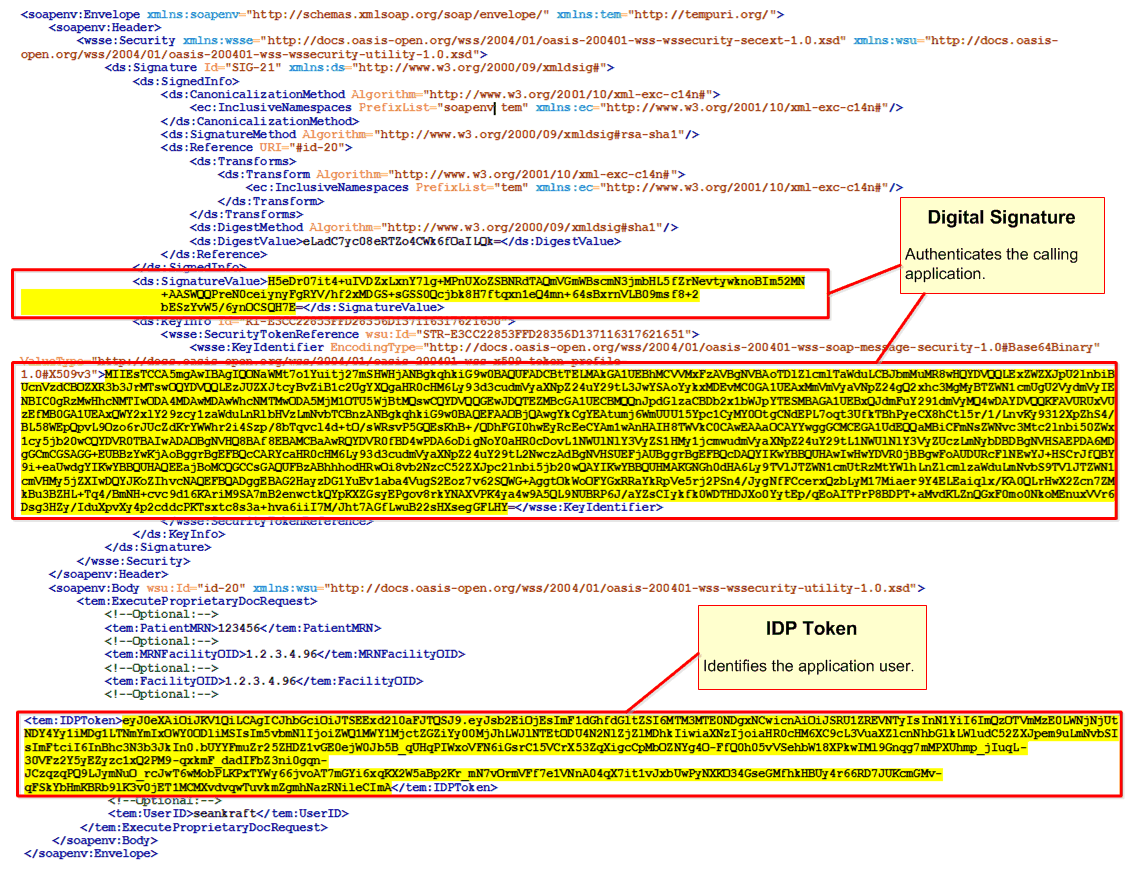


Figure 1. Soap Message Call: ExecuteProprietaryDoc

# SHIN-NY API Calls (Clinical Data Methods)

## Method: ExecuteXCPD (Step #1)

Request EMPI for a known patient.

### Endpoint

[http://hackathon.shinnyapi.org/messagegateway/ExecuteXCPD](http://hackathon.shinnyconnect.org/messagegateway/ExecuteXCPD)

### Input: XCPD Request Message

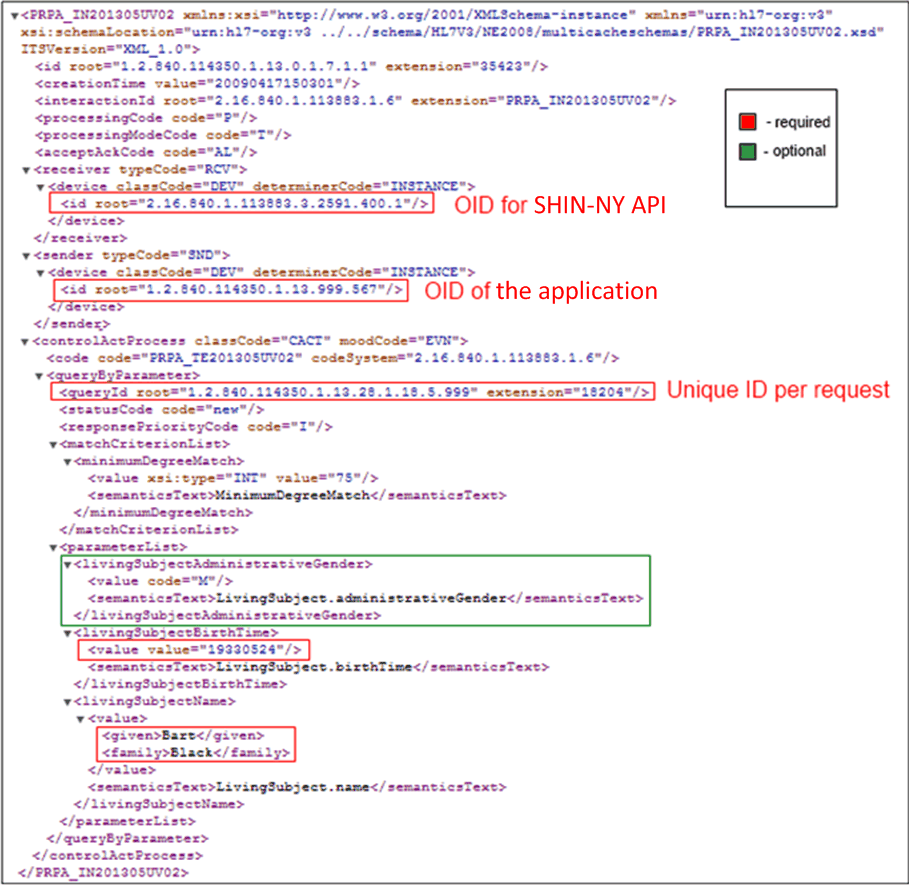


Figure 2. XCPD Request

### Output #1: XCPD Request Response (Containing Patient EMPI)

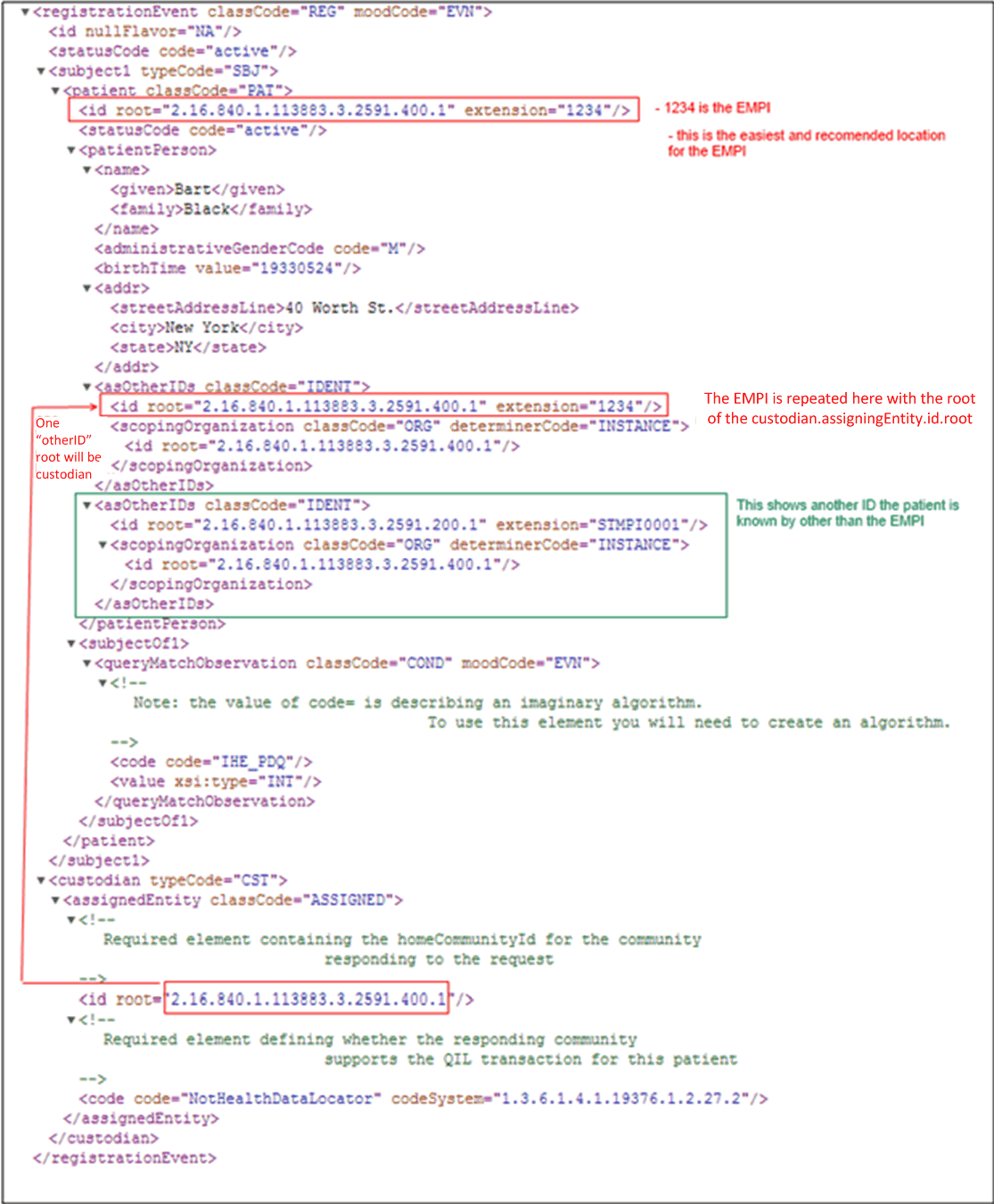


Figure 3. XCPD Response

### Output #2: Response to Unsuccessful XCPD Request: Internal Error

Internal errors indicate that something that was sent in the request to the SHIN-NY API was rejected.

**Note:** This is an IHE profile and IHE profiles do not provide transparency into the causes of errors.

Carefully examine what was sent and try to determine what might be wrong, (e.g., malformed parameter).

**XPath:** //detectedIssueManagement/code (Highlighted)

<?xml version="1.0" encoding="UTF-8"?>  
<PRPA\_IN201306UV02 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="urn:hl7-org:v3 ../../schema/HL7V3/NE2008/multicacheschemas/PRPA\_IN201306UV02.xsd"  
 xmlns="urn:hl7-org:v3" ITSVersion="XML\_1.0">  
 <id root="1.2.840.114350.1.13.999.238" extension="55789"/>  
 <creationTime value="20090417150302"/>  
 <interactionId root="2.16.840.1.113883.1.6" extension="PRPA\_IN201306UV02"/>  
 <processingCode code="P"/>  
 <processingModeCode code="T"/>  
 <acceptAckCode code="NE"/>  
 <receiver typeCode="RCV">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="2.16.840.1.113883.3.2591.100.1.1.6"/>  
 </device>  
 </receiver>  
 <sender typeCode="SND">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="2.16.840.1.113883.3.2591.400.1.1"/>  
 <telecom value="http://servicelocation/IHEXCPDRespondingGateway"/>  
 </device>  
 </sender>  
 <acknowledgement>  
 <typeCode code="AE"/>  
 <targetMessage>  
 <id root="1.2.840.114350.1.13.0.1.7.1.1" extension="35423"/>  
 </targetMessage>  
 </acknowledgement>  
 <controlActProcess classCode="CACT" moodCode="EVN">  
 <code code="PRPA\_TE201306UV02" codeSystem="2.16.840.1.113883.1.6"/>  
 <!-- Used to indicate that the responder is not able to process the request at this time -->  
 <reasonOf typeCode="RSON">  
 <detectedIssueEvent classCode="ALRT" moodCode="EVN">  
 <code code="ActAdministrativeDetectedIssueCode" codeSystem="2.16.840.1.113883.5.4"/>  
 <mitigatedBy typeCode="MITGT">  
 <detectedIssueManagement classCode="ACT" moodCode="EVN">  
 <code code="InternalError" codeSystem="1.3.6.1.4.1.19376.1.2.27.3"/>  
 </detectedIssueManagement>  
 </mitigatedBy>  
 </detectedIssueEvent>  
 </reasonOf>  
 <queryAck>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <queryResponseCode code="AE"/>  
 </queryAck>  
 <queryByParameter>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <statusCode code="new"/>  
 <parameterList>  
 <livingSubjectAdministrativeGender>  
 <value code="M"/>  
 <semanticsText>LivingSubject.administrativeGender</semanticsText>  
 </livingSubjectAdministrativeGender>  
 <livingSubjectBirthTime>  
 <value value="INVALIDDATE"/>  
 <semanticsText>LivingSubject.birthTime</semanticsText>  
 </livingSubjectBirthTime>  
 <livingSubjectName>  
 <value>  
 <given>Bart</given>  
 <family>Black</family>  
 </value>  
 <semanticsText>LivingSubject.name</semanticsText>  
 </livingSubjectName>   
 </parameterList>  
 </queryByParameter>  
 </controlActProcess>  
</PRPA\_IN201306UV02>

### Output #3: Response to Unsuccessful XCPD Request: No Exact Match Found

If the criteria sent in the XCPD request returns multiple matches, no patient results are returned because only one match is an acceptable response. Consider sending the request with the additional gender demographic: Male/Female.

**XPath:** //controlActProcess/detectedIssueEvent (Highlighted)

**XPath:** //triggerFor/actOrderRequired/code/@code (Highlighted)

<?xml version="1.0" encoding="UTF-8"?>  
<PRPA\_IN201306UV02 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="urn:hl7-org:v3" xmlns="urn:hl7-org:v3" ITSVersion="XML\_1.0">  
 <id root="1.2.840.114350.1.13.999.238" extension="55789"/>  
 <creationTime value="20090417150302"/>  
 <interactionId root="2.16.840.1.113883.1.6" extension="PRPA\_IN201306UV02"/>  
 <processingCode code="P"/>  
 <processingModeCode code="T"/>  
 <acceptAckCode code="NE"/>  
 <receiver typeCode="RCV">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="1.2.840.114350.1.13.999.567"/>  
 </device>  
 </receiver>  
 <sender typeCode="SND">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="2.16.840.1.113883.3.2591.400.1.1"/>  
 <telecom value="http://servicelocation/IHEXCPDRespondingGateway"/>  
 </device>  
 </sender>  
 <acknowledgement>  
 <typeCode code="AA"/>  
 <targetMessage>  
 <id root="1.2.840.114350.1.13.0.1.7.1.1" extension="35423"/>  
 </targetMessage>  
 </acknowledgement>  
 <controlActProcess classCode="CACT" moodCode="EVN">  
 <code code="PRPA\_TE201306UV02" codeSystem="2.16.840.1.113883.1.6"/>  
 <detectedIssueEvent classCode="ALRT" moodCode="EVN">  
 <code code="ActAdministrativeDetectedIssueCode" codeSystem="2.16.840.1.113883.5.4"/>  
 <triggerFor typeCode="TRIG">  
 <actOrderRequired classCode="ACT" moodCode="RQO">  
 <code code="LivingSubjectAdministrativeGenderRequested" codeSystem="1.3.6.1.4.1.19376.1.2.27.1"/>  
 </actOrderRequired>  
 </triggerFor>  
 </detectedIssueEvent>   
 <subject typeCode="SUBJ">  
 <!--No RegistrationEvent is included -->  
 </subject>  
 <queryAck>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <queryResponseCode code="OK"/>  
 </queryAck>  
 <queryByParameter>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <statusCode code="new"/>  
 <parameterList>  
 <livingSubjectAdministrativeGender>  
 <value code=""/>  
 <semanticsText>LivingSubject.administrativeGender</semanticsText>  
 </livingSubjectAdministrativeGender>  
 <livingSubjectBirthTime>  
 <value value="19330514"/>  
 <semanticsText>LivingSubject.birthTime</semanticsText>  
 </livingSubjectBirthTime>  
 <livingSubjectName>  
 <value>  
 <given>Patrick</given>  
 <family>Ueckert</family>  
 </value>  
 <semanticsText>LivingSubject.name</semanticsText>  
 </livingSubjectName>  
 </parameterList>  
 </queryByParameter>  
 </controlActProcess>  
</PRPA\_IN201306UV02>

### Output #4: Response to Unsuccessful XCPD Request: No Match Found

**XPath:** //controlActProcess/queryAck/queryResponseCode/[@code = 'NF'] (Highlighted)

<PRPA\_IN201306UV02 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="urn:hl7-org:v3"  
 xmlns="urn:hl7-org:v3" ITSVersion="XML\_1.0">  
 <id root="1.2.840.114350.1.13.999.238" extension="55789"/>  
 <creationTime value="20090417150302"/>  
 <interactionId root="2.16.840.1.113883.1.6" extension="PRPA\_IN201306UV02"/>  
 <processingCode code="P"/>  
 <processingModeCode code="T"/>  
 <acceptAckCode code="NE"/>  
 <receiver typeCode="RCV">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="2.16.840.1.113883.3.2591.100.1.1.6"/>  
 </device>  
 </receiver>  
 <sender typeCode="SND">  
 <device classCode="DEV" determinerCode="INSTANCE">  
 <id root="2.16.840.1.113883.3.2591.400.1.1"/>  
 <telecom value="http://servicelocation/IHEXCPDRespondingGateway"/>  
 </device>  
 </sender>  
 <acknowledgement>  
 <typeCode code="AA"/>  
 <targetMessage>  
 <id root="1.2.840.114350.1.13.0.1.7.1.1" extension="35423"/>  
 </targetMessage>  
 </acknowledgement>  
 <controlActProcess classCode="CACT" moodCode="EVN">  
 <subject typeCode="SUBJ">  
 <!--No RegistrationEvent is included -->  
 </subject>  
 <queryAck>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <queryResponseCode code="NF"/>  
 </queryAck>  
 <queryByParameter>  
 <queryId root="1.2.840.114350.1.13.28.1.18.5.999" extension="18204"/>  
 <statusCode code="new"/>  
 <parameterList>  
 <livingSubjectAdministrativeGender>  
 <value code="M"/>  
 <semanticsText>LivingSubject.administrativeGender</semanticsText>  
 </livingSubjectAdministrativeGender>  
 <livingSubjectBirthTime>  
 <value value="19330514"/>  
 <semanticsText>LivingSubject.birthTime</semanticsText>  
 </livingSubjectBirthTime>  
 <livingSubjectName>  
 <value>  
 <given>Forest</given>  
 <family>Gump</family>  
 </value>  
 <semanticsText>LivingSubject.name</semanticsText>  
 </livingSubjectName>   
 </parameterList>  
 </queryByParameter>  
 </controlActProcess>  
</PRPA\_IN201306UV02>

## Method: ExecuteXCAQuery (Step #2)

Request to obtain a list of document identifiers using the EMPI for a patient.

### Endpoint

[http://hackathon.shinnyapi.org/messagegateway/ExecuteXCAQ](http://hackathon.shinnyconnect.org/messagegateway/ExecuteXCAQ)

### Input: EMPI



Figure 4. XCAQ Request

### Output: ExecuteXCAQuery Response XML: XDSDocument Entry.uniqueId

The ExtrinsicObject sections of the response contain both repository and document IDs that are passed to the ExecuteXCARetrieve request to retrieve the documents pertaining to the patient.

This example shows the ExternalIdentifier element where the IDs reside. Select the appropriate ExternalIdentifier, based on the patient name.

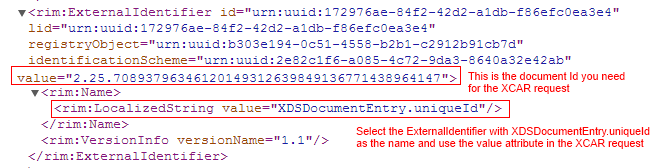


Figure 5. ExecuteXCAQ Response

### Output: ExecuteXCAQuery Response XML: RepositoryUniqueId

You will also need the repository ID value for insertion in the ExecuteXCARetrieve request. *Figure 6* shows the location of the ID value in the ExecuteXCAQuery response.

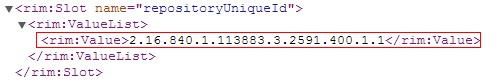


Figure 6. RepositoryUniqueId value in XML response.

## Method: ExecuteXCARetrieve (Step #3)

Request to get a consolidated CCD using a document identifier parameter, (i.e., DocumentUniqueId).

### Endpoint

[http://hackathon.shinnyapi.org/messagegateway/ExecuteXCAR](http://hackathon.shinnyconnect.org/messagegateway/ExecuteXCAR)

### Input: ExecuteXCARetrieve request for a known patient

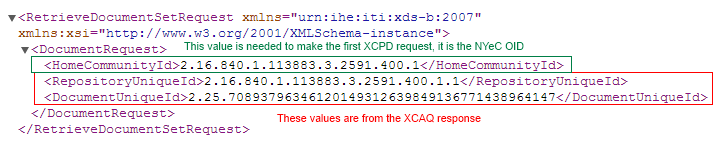


Figure 7. ExecuteXCARetrieve Request XML

### Output: ExecuteXCARetrieve Response XML

This API call returns the consolidated CCD.   
Failure is unlikely at this point because the request contains valid IDs for both a repository and document.

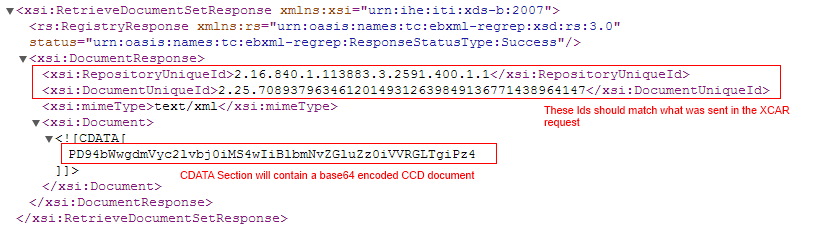


Figure 8. XML Response to ExecuteXCARetrieve Request

## Method: ExecuteProprietaryDoc (Alternative to Steps 1-3)

Request a patient’s CCD using patient’s MRN plus a facility OID.

### Endpoint

[http://hackathon.shinnyapi.org/messagegateway/ExecuteProprietary](http://hackathon.shinnyconnect.org/messagegateway/ExecuteProprietary)

### Output: ExecuteProprietaryDoc XML

This API call returns a consolidated CCD.   
Failure is unlikely at this point because the request contains valid IDs for both a repository and document.

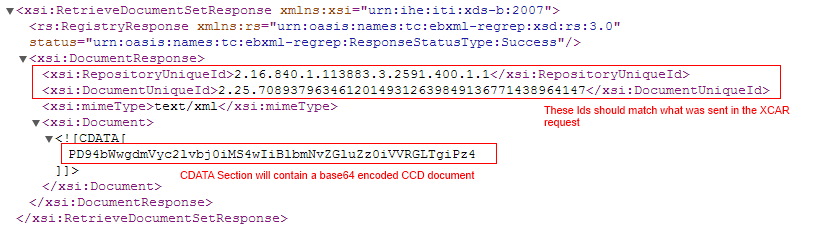


Figure 9. XML Response to ExecuteProprietaryDoc Request