Equator REST Methods

1 Equator
2 Overview of Available Calls
2.1 IsProviderOnline
2.2 Provider Search by Name
2.3 Provider Search by ID
2.4 Provider Search by Organisation
2.5 Practice Search via NASH HPIO
2.6 Get Local Providers List
2.7 Send Message
3 HTTP Request, Method POST

Equator

Supports HTTP 1.1

Uses POST Messages.

The Server accepts encrypted and non-encryted messages but all external access will be PGP/PKI /GNUPG/HTTPS encrypted and signed. The signing key must be enabled on the Server or post will Fail. The server Public key can be obtained directly from the Server. PKI keys are available on LDAP.

The HL7 Messages can be "Classic" HL7 Encoded.

Note:

⚠ Only the ASCII displayable character set (hexadecimal values between 20 and 7E, inclusive) and the AS CII Carriage Return character is permitted for the content of your communications at this time.

Overview of Available Calls

See the following article for understanding the responses: Recipient Lookup Response Explanation

Method	URL	Formats Supported	Formats Required Sending	HTTP Method
IsProvider Online	/rest/routing/isRoutable/byIDNumber?ID=[ProviderNo] This method indicates if the provider communicates via Medical Objects. It does not indicate that they are online at the exact time the request is made. [ProviderNo] represents the Provider number that you wish to search for. Returns are True, False (Newer server builds will also return INVALID if the provider ID does not pass its respective validation scheme checks). NOTE: Stems of a provider number will return True. e.g. 296163 only on older Equator builds. 296163 is the stem of 296163HY, 296163KX etc and will return True as we have those number listed as online.	Basic Text		GET
Provider Search by Name	/rest/provider/lookup/byname?FAMILYNAME=[Surname]&GIVENNAME=[GivenName]&FORMAT=[Format] This performs a partial search against the Surname and Given names provided and returns a list of the potential matches.	JSON or XML	n/a	GET
Provider Search by ID	/rest/provider/lookup/byid?ID=[MedicareProviderNo or MedicalObjectsAssignedProviderNo]&FORMAT=[Format] This performs a search for a particular provider and returns the details of the provider if found. Format is either JSON (default) or XML NOTE: If only the first portion of a provider number is supplied multiple possible results may be returned. E.g. Lookup of value 296163 returns 296163HY, 296163EL, 296163KX, 296163MW and 296163JJ.	JSON or XML	n/a	GET

Provider Search by Organisation	/rest/provider/lookup/byorganisation?ORGANISATION=[OrganisationName]&FORMAT=[Format] &onlineProvidersOnly=True Returns all providers for an organisation name(Practice name). If your patient doesn't know the practitioner they will be seeing at the practice use the first valid returned result. For JSON: Always use "&onlineProvidersOnly=True". You don't want providers you can't deliver to! If an idNumber is available for the namespaceID "AUSHICPR" please use as the primary. idNumber for namespaceID "Medical-Objects" are only to be used if an AUSHICPR is not available. For XML: Only routable providers will be listed. There should be no need to filter. Output limited to 1000 records Results are returned alphabetically based upon Surname then Firstname.	JSON or XML	n/a	GET
Practice Search via NASH HPIO	/rest/routing/isRoutable/byHPIO?HPIO=[HPIO] Returns True if practice is available via HPIO otherwise returns False. Note that only a small portion of our network uses this delivery method currently.	Basic Text		GET
Get Local Providers List	/rest/practice/providers/asJSON This returns a JSON list of the health professionals at your location that are registered on the local Equator (Your providers that are installed with Medical Objects on this server). The format is consistent with HL7 v2.3.1 XCN segment	JSON	n/a	GET

<table>
<tr>
<td>Send Message</td>
<td>/hl7/
Accept: application/hl7

Content-type: application/hl7

Note the trailing / in /hl7/ must not be missed.

Allows sending a HL7 message through the Medical-Objects network. The POST returns a HL7 Ack message(which you will have to check for a positive or negative response) or an error description in the HTTP response message-body (this should not be confused with "200 OK" in the HTTP Response header Status Code).

Please note that MedicareProviderNo or MedicalObjectsAssignedProviderNo are rejected if invalid. This can be disabled for the "from" provider if necessary. Equator Build 5136 and above. Please note certificate authentication is required. Please contact Medical Objects for more details.
<table>
<tr><th>Message In</th><th>Status Code</th><th>Status</th><th>Expected Response</th><th>Valid Ack Codes</th></tr>
<tr><td>Valid HL7</td><td>200</td><td>OK</td><td>HL7 Acknowledgement</td><td>AA, AE, AR, CA, CE, CR</td></tr>
<tr><td>Flawed HL7</td><td>200</td><td>OK</td><td>HL7 Acknowledgement with ERR segment</td><td>AE, AR, CE, CR</td></tr>
<tr><td>Critically Flawed HL7</td><td>400</td><td>Bad Request</td><td>Error message. HL7 response not possible with invalid input</td><td>N/A</td></tr>
<tr><td>Authorisation Failure</td><td>401</td><td>Unauthorised</td><td>Error message. HL7 response not possible without input HL7</td><td>N/A</td></tr>
<tr><td>Authorisation Failure</td><td>None</td><td>None</td><td>No response from server. An example of this is when an expired certificate is used to connect as the connection is completely dropped. HL7 response not possible without input HL7</td><td>N/A</td></tr>
</table>
</td>
<td>HL7</td>
<td>HL7</td>
<td>POST</td>
</tr>
</table>

HTTP Request, Method POST

1. **Encrypted Message Headers**
   Connection: keep-alive (this may not be respected :-))
   Accept: "application/pgp-encrypted" or "application/pki-encrypted or "application/gnupg-encrypted"
   Username: The full PGP/GNUPG/PKI Keyname of the signing Key. No Password.

2. **Plain Text Headers (HTTPS or Local LAN connections only)**
   Connection: keep-alive
   Accept: application/hl7
   Set username and password as per basic Authentification

HTTPS Notes:

Clients must support TLS >=1.2.

Clients connecting to secure ports running on 443 generally will be required to support Server Name Indication to access secure server ports (see rfc3546). If this is not supported then, the TLS connection request may be actively reset.

Depending on the secure server port instance, a trusted x509 client certificate may be required to authenticate to establish a connection. Additional username/password authentication is available to provide extra authorisation and identification for HTTPS connections.

PKI/PGP Notes:

The post data is the PGP/PKI/GNUPG encrypted message or plain text message as appropriate.

Will accept Single message or a batch with FHS/BHS/BTS/FTS, no extra characters before or after the message allowed.

PKI Encrypted messages are to be base64 encoded. Will be base 64 encoded on the way out.

As PKI API does not support signed/encryted messages in one ASN.1 wrapper the message is first encypted ('Encoding:Enveloped') and then signed ('Encoding:Signed'). The signer must have a valid PKI Certificate and the message should be encrypted with the target servers encryption site certificate.

For new PKI users the encrytion certificate must be available on HIC LDAP server. Rights are granted on the basis of the signer.

PGP messages to be radix encoded rather than binary.

GNUPG messages should be radix encoded (ie --armor).

The Server response

Will be PGP/PKI encrypted or plain text depending on the zone. PGP in means PGP out.

The message will be encrypted with the requesting public key and signed by the server key.

For Encrypted messages
ContentEncoding: "pgp-encrypted" or "base64-pki-encrypted" or "gnupg-encrypted"

If the message is not parsable then

ContentEncoding: application/error

In case of an error then the response will be a plain text error message. This will occur if the message fails parsing or the public key of the signer is not registered. If the message is OK but what is requested is not then standard HL7 error handling (Mainly ERR segments) will be used. ie 'application/error' means a connection error. It can be displayed to the user via eg an exception handler and has a multiline nature which Name=Value pairs to give more info, assuming that its not an unauthorised PGP/GNUPG/PKI Key.

The Server URL
Takes the form of

HTTP:

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/'

e.g. http://203.42.156.38:2000/hl7/

HTTPS with SNI:

'https://' + HL7HostName + '/hl7/'

e.g. https://hd-0ae5c60c-a510-43b3-a509-c57f29b2d368-guid.moc.test.medical-objects.com.au/hl7

or

'https://' + HL7HostName + ':' + inttostr(Port) + '/hl7/'

e.g. https://hd-0ae5c60c-a510-43b3-a509-c57f29b2d368-guid.moc.test.medical-objects.com.au:1300/hl7

If the message is in PIT format (not desirable) then

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/pit/'

e.g. http://203.42.156.38:2000/hl7/pit/

In the case of a PIT message a HL7 ACK will be returned.

Utility Methods
1. The PGP/GNUPG Key Name for the Server
HTTP GET with

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?METHOD=SERVER_KEY_NAME'

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?METHOD=SERVER_GNUPG_KEY_NAME'

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?METHOD=SERVER_PKI_KEY_NAME'

2. The PGP/GNUPG Public Key of the Server.
HTTP GET with

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?METHOD=SERVER_PGP_KEY'

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?METHOD=SERVER_GNUPG_KEY'

3. To upload a PGP/GNUPG Public Key for a potential user
HTTP POST with

'http://' + HL7HostName + ':' + inttostr(Port) + '/hl7/admin?' +

'METHOD=CLIENT_KEY_UPLOAD' + (or 'GNUPG_KEY_UPLOAD')

'&KEYNAME=<PGP Key Name>' +

'&CLIENTNAME=<Text Name of Potential User>' +

'&PHONENUMBER=<Contact Number of Potential User>'

nb: Don't include <>

The Client Public Key is Uploaded as the POST Data.

The key has to be manually validated and a user account created.

The user must know the Key FingerPrint of the Key they are planning to use.

The server will return, a error response if not permitted or some help text if the key has been accepted.
The ability to accept keys in this way is an option which can be turned off.

This text gives helpful info which can be displayed to the user.

Server rights:
All users will have a userlevel. The most basic one just allows uploading HL7 ORU Messages (ORU^R01). This is the base right that a valid PKI key permits. In the STF server a valid PKI key permits provider lookup (MFQ^M02). A user can edit their own STF records using MFN^M02 messages.