



## NextGen Connect Code Library Report

Prepared for: matthy@dehoog.xyz  
Prepared by: X Connections Ltd  
<https://documentor.email>  
support@documentor.email  
Creation date report: 20 mrt 2019 09:17:50

X Connections has no business relationship whatsoever with and are not employed by Mirth Corporation or NextGen Healthcare

## 1 Code Libraries

The following Code Libraries are present in this Library Export file. First the libraries present are listed, then every library is specified separately.

Library Name	Documentor
Version:	3.7.0
Revision:	12
id:	a2475505-cc05-4566-8194-1167faaec63
Last Modified:	1551004947745
Include in new channels:	false
Description:	
This library contains function for the documentor 2019 copyright FS Consulting BV	

Library Name	Library 1
Version:	3.7.0
Revision:	2
id:	b5c1e6cc-b0de-47ee-bf19-92aa18bb3e20
Last Modified:	1550587297964
Include in new channels:	false
Description:	

## 2 Code Templates

Library Name: Documentor	
Code Template	<xsl:choose>
Version:	3.7.0
Revision:	2
id:	b039559e-b329-4cb8-ae65-36a513ff2bf8
Last Modified:	1550830216973
Type:	DRAG_AND_DROP_CODE
Code:	<pre>/**  &lt;xsl:choose&gt; code snipped &lt;xsl:choose&gt; */ &lt;xsl:choose&gt;   &lt;xsl:when test=""&gt;      &lt;/xsl:when&gt;     &lt;xsl:otherwise&gt;      &lt;/xsl:otherwise&gt;   &lt;/xsl:choose&gt;</pre>

Library Name: Documentor	
Code Template	<xsl:for-each>
Version:	3.7.0
Revision:	1
id:	bdbbcb1a-d063-4baa-85a8-58c509c831f2
Last Modified:	1549287569223
Type:	DRAG_AND_DROP_CODE
Code:	<pre>/**   &lt;xsl:for-each&gt; code snipped   &lt;xsl:for-each select=""&gt;   &lt;/xsl:for-each&gt; */ &lt;xsl:for-each select=""&gt;  &lt;/xsl:for-each&gt;</pre>

Library Name: Documentor	
Code Template	<xsl:if/>
Version:	3.7.0
Revision:	1
id:	fe8acaaf-ff8f-433f-aab4-090ed94f00a3
Last Modified:	1549287568878
Type:	DRAG_AND_DROP_CODE
Code:	<pre>/**   &lt;xsl:if test="@show_comments = 'true'"&gt; */ &lt;xsl:if test="@show_comments = 'true'"&gt;   ...some output if true... &lt;/xsl:if&gt;</pre>

Library Name: Documentor	
Code Template	<xsl:variable name>
Version:	3.7.0
Revision:	1
id:	15e25032-73b3-4ec7-8f53-a618eebea23d
Last Modified:	1549287568978
Type:	DRAG_AND_DROP_CODE
Code:	
<pre>/**   &lt;xsl:variable name/&gt; code snipped */ &lt;xsl:variable name="string1" select="string[1]"/&gt;</pre>	

Library Name: Documentor	
Code Template	addDays2Date
Version:	3.7.0
Revision:	3
id:	a7233613-782a-48cf-8137-ab80586bf3e9
Last Modified:	1550830458963
Type:	FUNCTION
Code:	<pre>/**  * add days to date  *  * @param {Any} indate -  * @param {Any} days -  * @return {String} return description  */ function addDays2Date(indate,days) {   var date = new Date(indate);   date.setDate(date.getDate() + days);   datestr = DateUtil.formatDate('yyyy-MM-dd', date);   return datestr.toString(); }</pre>

Library Name: Documentor	
Code Template	convertJavaTime
Version:	3.7.0
Revision:	3
id:	d01da9a5-1365-432c-9e77-545a2ac50ad4
Last Modified:	1550831845165
Type:	FUNCTION
Code:	<pre> /**  * Documentor: This function converts java time (javaTime) like '1338341400000' into  * an other format (timeFormat)  *  * @param {Any} javaTime -  * @param {Any} timeFormat -  * @return {String} return description  */ function convertJavaTime(javaTime, timeFormat) {     var javaTime = javaTime.toString();     javaTime = parseInt(javaTime)     var theDateTime =new org.joda.time.DateTime(javaTime)      var strConvertedTime = org.joda.time.format.DateTimeFormat.forPattern(timeFormat).print(theDateTime);     //logger.info("strConvertedTime is " + strConvertedTime);      return strConvertedTime } </pre>



<b>Library Name: Documentor</b>	
<b>Code Template</b>	<b>executeRuntimeCommand</b>
Version:	3.7.0
Revision:	1
id:	c2125719-ee67-47dc-a021-cd29037e6760
Last Modified:	1549795676120
Type:	FUNCTION
Code:	

```

/**
  Modify the description here. Modify the function name and parameters as needed. One function per
  template is recommended; create a new code template for each new function.

  @param {Any} args -
  @param {Any} charset -
  @return {Any}
*/
function executeRuntimeCommand(args, charset) {
  var process = java.lang.Runtime.getRuntime().exec(args);
  var stdoutConsumer = new StreamConsumer(process.getInputStream(), charset);
  var stderrConsumer = new StreamConsumer(process.getErrorStream(), charset);
  return {
    exitValue : process.waitFor(),
    stdout : stdoutConsumer.getOutput(),
    stderr : stderrConsumer.getOutput()
  };
}

function StreamConsumer(is, charset) {
  var output = "";

  var thread = new java.lang.Thread({
    run: function() {
      if (typeof charset !==
'undefined') {
        output =
org.apache.commons.io.IOUtils.toString(is, charset);

      } else {
        output = org.apache.commons.io.IOUtils.toString(is);
      }
    }
  });

  this.interrupt = function() {
    thread.interrupt();
  }

  this.getOutput = function() {
    thread.join();
    return output;
  };

  thread.start();
}

```

Library Name: Documentor	
Code Template	executeSharedDBStatementDoc
Version:	3.7.0
Revision:	1
id:	97676940-ab7f-4e09-91ff-b125e68dfb84
Last Modified:	1549287569033
Type:	FUNCTION
Code:	<pre> /**  * Documentor: Executes a SQL statement using a shared database connection stored in the Global Map. If  * the connection does not exist, creates it.  *  * @param {String} statement - The SQL statement to execute  * @param {Boolean} isQuery - true if the statement is a SELECT, otherwise false  * @param {List} paramList - A Java List object containing the parameters for the statement  * (optional)  * @return {Any}  */ function executeSharedDBStatementDoc(statement, isQuery, paramList) {     var result;      try     {         var dbConnDoc = globalMap.get("dbConnDoc");          if (dbConnDoc == null    dbConnDoc.getConnection().isClosed())         {             var dbConnDoc = DatabaseConnectionFactory.createDatabaseConnection(\$('doc_srv'), \$('doc_db'), \$('doc_us'), \$('doc_pw'));              globalMap.put("dbConnDoc", dbConnDoc);         }          if (paramList == undefined)         {             paramList = new Packages.java.util.ArrayList();         }          if (isQuery)         {             result = dbConnDoc.executeCachedQuery(statement, paramList);         }          else         {             //logger.info(statement); </pre>

```
        result = dbConnDoc.executeUpdate(statement, paramList);
    }
}
catch (e)
{
    logger.error("Error in Execute Statement with Shared DB Conn dbConnDoc: " + e + "-" + statement);
}
finally
{
    return result;
}
}
```

Library Name: Documentor	
Code Template	get_sub_daystogo
Version:	3.7.0
Revision:	1
id:	01cb9cd4-7575-411a-9a5b-c079015462e6
Last Modified:	1549287569089
Type:	FUNCTION
Code:	<pre> /**   Documentor: Get the amount of subscription days left. The amount of days left is stored into channel   map sub_daytogo. Channel map sub_enddate contains the date when the subscription ends.    @param {Any} order_id - the string to distinguish file types   @param {Any} cust_id - identification of the customer   @return {String} return description */ function get_sub_daystogo(order_id, cust_id) {   //Check if the subscription end date is not yet passed. } </pre>

Library Name: Documentor	
Code Template	write2log
Version:	3.7.0
Revision:	6
id:	6af2c114-cb60-4aff-9735-d11a1554940e
Last Modified:	1550139157690
Type:	FUNCTION
Code:	<pre> /**  * Documentor: Deze functie schrijft naar een log file.  *  * @param {String} arg1 - arg1 description  * @return {String} return description  */ function write2log(loglevel, status, log_msg) {   if(\$('logIt') == 'true'){     if(loglevel &gt;= \$('logLevel')){       //Get the current month       var monthString = DateUtil.getCurrentDate('M');        //Get the current day       var dayString = DateUtil.getCurrentDate('dd');       logger.info(log_msg);       FileUtil.write(\$('logDir') + monthString + "\\" + dayString + "\\" + \$('logFileName'),true,Date() + " [" + channelName + "]" + status + "-" + log_msg + "\r\n");     }   } } </pre>

**Library Name: Library 1****Code Template**      **Template 1**

Version: 3.7.0

Revision: 1

id: 54d930b1-5fda-4ca9-8ac1-c16f34b7e94c

Last Modified: 1550584005643

Type: FUNCTION

Code:

/\*\*

Modify the description here. Modify the function name and parameters as needed. One function per template is recommended; create a new code template for each new function.

@param {String} arg1 - arg1 description

@return {String} return description

\*/

```
function new_function1(arg1) {
```

```
    // TODO: Enter code here
```

```
}
```

