

XML を返す url から XML データを取得、解析し、Node のリストを作成する。

```
URLConnection conn = (new URL(url)).openConnection();
conn.connect();

InputStream in = new InputStream(new InputStreamReader(conn.getInputStream()));
XPathFactory xfactory = XPathFactory.newInstance();
XPath xpath = xfactory.newXPath();

NodeList result = (NodeList)xpath.evaluate("//*[local-name()='Parameter']/text()", in,
XPathConstants.NODESET);

for (int i=0; i<result.getLength(); i++) {
    System.out.println(result.item(i).toString());
}

package generate;

import java.util.Collections;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.Map;
import java.util.Set;

import javax.xml.XMLConstants;
import javax.xml.namespace.NamespaceContext;

public class NameSpaceContextImpl implements NamespaceContext {

    Map<String, String> map = new HashMap<String, String>();

    public NameSpaceContextImpl() {
        setNamespaceURI(XMLConstants.DEFAULT_NS_PREFIX, XMLConstants.NULL_NS_URI);
        setNamespaceURI(XMLConstants.XML_NS_PREFIX, XMLConstants.XML_NS_URI);
        setNamespaceURI(XMLConstants.XMLNS_ATTRIBUTE, XMLConstants.XMLNS_ATTRIBUTE_NS_URI);
    }

    public void setNamespaceURI(String prefix, String uri) {
        map.put(prefix, uri);
    }

    public String getNamespaceURI(String prefix) {
        return map.get(prefix);
    }

    public String getPrefix(String namespaceURI) {
        if (namespaceURI == null) {
            throw new IllegalArgumentException();
        }

        Set<Map.Entry<String, String>>set = map.entrySet();
        for (Map.Entry<String, String>item : set) {
            if (namespaceURI.equals(item.getValue())) {
                return item.getKey();
            }
        }
        return XMLConstants.NULL_NS_URI;
    }

    public Iterator getPrefixes(String namespaceURI) {
        Set<String> prefixes = new HashSet<String>();

        Set<Map.Entry<String, String>>set = map.entrySet();
        for (Map.Entry<String, String>item : set) {
            if (namespaceURI.equals(item.getValue())) {
                prefixes.add(item.getKey());
            }
        }

        return Collections.unmodifiableCollection(prefixes).iterator();
    }
}
```