Communicating with the device

by SuperBonBon

1. Messaging

Once you have discovered a device you can start to play with it and send messages. An UPNP# root device is made of several UPNPDevice objects (child devices) which are containing UPNPService. The services object contains all the available operations and state variables names :

```
UPNPRootDevice rootDevice = net.sbbi.upnp.Discovery.discover()[0];
//let's look at the root device child devices
List childDevices = rootDevice.getChildDevices();
if ( childDevices != null ) {
  for ( Iterator i = childDevices.iterator(); i.hasNext(); ) {
    UPNPDevice child = (UPNPDevice)i.next();
    System.out.println( "Reaching child device " + child.getUDN() );
    // now let's look a the child device services
    for ( Iterator i2 = child.getServices().iterator(); i2.hasNext(); ) {
      UPNPService service = (UPNPService)i2.next();
      System.out.println( "Reaching child device service id " + service.getServiceId(
    }
  }
}
```

You can also look for child devices directly using a specific device URI for example let's lookup an IGD WANDevice:

```
String dvURN = "upnp:schemas-upnp-org:device:WANDevice:1";
UPNPDevice myIGDWANDevice = IGDRootDevice.getChildDevice( dvURN );
if ( myIGDWANDevice != null ) {
  System.out.println( "IGD WAN device " + myIGDWANDevice.getUDN() );
}
```

Finally you need to lookup a specific device service to start to interact with the device :

```
String dvURN = "upnp:schemas-upnp-org:device:WANDevice:1";
UPNPDevice myIGDWANDevice = IGDRootDevice.getChildDevice( dvURN );
if ( myIGDWANDevice != null ) {
  System.out.println( "IGD WAN device " + myIGDWANDevice.getUDN() );
  String srvURN = "upnp:schemas-upnp-org:service:WANIpConnection:1";
  UPNPService WANIpConnectionSrv = myIGDWANDevice.getService( srvURN );
  if ( WANIpConnectionSrv != null ) {
    System.out.println( "IGD WAN device WANIpConnection service " +
                        WANIpConnectionSrv.getServiceId() );
  }
}
```

The last step to start to give orders to the UPNP device is to create an ActionMessage object for a given service action name using an UPNPMessageFactory. The action message names, input and output argument are specified in the UPNP service specs or can be retreived with the UPNPService object :

```
String srvURN = "upnp:schemas-upnp-org:service:WANIpConnection:1";
UPNPService WANIpConnectionSrv = myIGDWANDevice.getService( srvURN );
if ( WANIpConnectionSrv != null ) {
  System.out.println( "IGD WAN device WANIpConnection service " +
                      WANIpConnectionSrv.getServiceId() );

  UPNPMessageFactory factory = UPNPMessageFactory.getNewInstance( WANIpConnectionSrv
  // let's try to retreive information concerning a
  // mapping entry on the UPNP router device. All the action names and arguments
  // used here are taken from the IGD specs available at http://www.upnp.org
  ActionMessage action = factory.getMessage( "GetSpecificPortMappingEntry" );
  // can return null if the action does not exists for the given service
  // or is not mandatory in the UPNP device specs.
  if ( action != null ) {
    // setting the input params
    action.setInputParameter( "NewRemoteHost", "" )
          .setInputParameter( "NewExternalPort", 21 )
          .setInputParameter( "NewProtocol", "TCP" );
    try {
      // let's invoke the action on the device
      // and retreive a response
      ActionResponse resp = action.service();
      System.out.println( "Mapping found" );
      System.out.println( resp.getOutActionArgumentValue( "NewInternalPort" ) );
      System.out.println( resp.getOutActionArgumentValue( "NewInternalClient" ) );
      System.out.println( resp.getOutActionArgumentValue( "NewEnabled" ) );
      System.out.println( resp.getOutActionArgumentValue( "NewPortMappingDescription"
      System.out.println( resp.getOutActionArgumentValue( "NewLeaseDuration" ) );
    } catch ( UPNPResponseException respEx ) {
      // the device responded with an error for example no mapping
      // existing the code returned are defined in the device specs
      if ( respEx.getDetailErrorCode().equals( "714" ) ) {
        // no entry matching according to the specs when code 714
        // is returned by the device
      } else {
        // DOH ! looks like a real error !
      }
    } catch ( IOException ioEx ) {
      // look like a communication problem occured with the device
    }
  }
}
```

2. State variable

You can also retreive the content of a state variable on the device using one more time a message factory to create an StateVariableMessage object for the desired state variable query :

```
String srvURN = "upnp:schemas-upnp-org:service:WANIpConnection:1";
UPNPService WANIpConnectionSrv = myIGDWANDevice.getService( srvURN );
if ( WANIpConnectionSrv != null ) {
  System.out.println( "IGD WAN device WANIpConnection service " +
                      WANIpConnectionSrv.getServiceId() );

  UPNPMessageFactory factory = UPNPMessageFactory.getNewInstance( WANIpConnectionSrv
  // let's try to retreive the content of the LastConnectionError WANIpConnection
  // service state variable
  StateVariableMessage stateVarMsg = factory.getStateVariableMessage( "LastConnection
  try {
    StateVariableResponse resp = stateVarMsg.service();
    System.out.println( "LastConnectionError device state variable value is :" +
                        resp.getStateVariableValue() );
  } catch ( UPNPResponseException respEx ) {
    if ( respEx.getDetailErrorCode().equals( "404" ) ) {
      // devices that do not implement state variables query
      // respond with a 404 error code
    } else {
      // DOH ! we have some unknown error
    }
  } catch ( IOException ioEx ) {
    // look like a communication problem occured with the device
  }
}
```