# A Developer Diary

{about:"code learn and share"}

Home      Data Science      Java      JavaScript      jBPM      Tools      Tips      About

June 3, 2015 By Abhisek Jana  —  Leave a Comment (Edit)

## How to write Java WebService Client using SAAJ API?
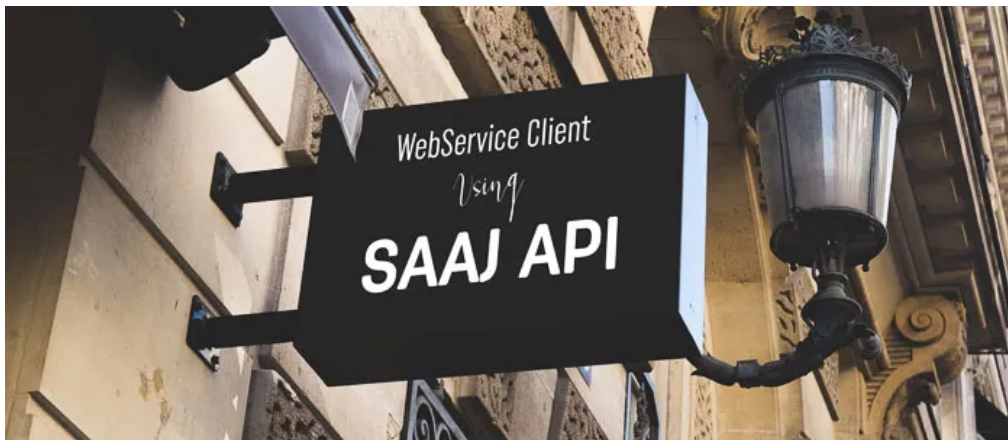


We often need a SOAP WebService Client and we might have just a sample request in hand with the endpoint or in case you need a generic webservice client then writing WebService Client using SAAJ API is a very easy way to invoke a WebService.

The method takes the URL endpoint and the request xml as the input and the return string would be the response from the WebService. In case of any errro the exception will be thrown from the SAAJ API.

```
public String invokeWebService(String destination,
                    String strSOAPRequest)
                    throws SOAPException, IOException {
        SOAPConnection connection;
        String response;
        connection = null;
```

```
                    response = "";
                    try {
                            SOAPMessage request = createSOAPMessage(strSOAPRequest);
                            SOAPConnectionFactory soapConnFactory =
SOAPConnectionFactory
                                            .newInstance();
                            connection = soapConnFactory.createConnection();
                            SOAPMessage reply = connection.call(request, destination);
                            response = getSOAPResponse(reply);
                    } catch (javax.xml.soap.SOAPException e) {
                            e.printStackTrace();
                            throw e;
                    } catch (IOException e) {
                            e.printStackTrace();
                            throw e;
                    } finally {
                            if (connection != null)
                                    try {
                                            connection.close();
                                    } catch (javax.xml.soap.SOAPException e) {
                                            e.printStackTrace();
                                    }
                    }
                    return response;
            }
```

We are creating a SOAPMessage here by passing the String request. You can add custom HTTPHeader here like HTTP Basic Authentication etc.

```
private SOAPMessage createSOAPMessage(
                    String strRequestXML)
                    throws SOAPException, IOException {
            ByteArrayInputStream is = new ByteArrayInputStream(
                            strRequestXML.getBytes());
            MessageFactory messageFactory = MessageFactory
                            .newInstance();
            MimeHeaders mimeHeaders = new MimeHeaders();
            mimeHeaders.addHeader("SOAPAction", "");
            mimeHeaders.addHeader("Content-Type", "text/xml; charset=UTF-8");
            SOAPMessage requestMsg = messageFactory.createMessage(
                            mimeHeaders, is);
            return requestMsg;
    }
```

This method would write the response to a ByteArrayOutputStream will be be returned back as a String.

```java
private java.lang.String getSOAPResponse(SOAPMessage reply)
                throws SOAPException, IOException {
        ByteArrayOutputStream os = new ByteArrayOutputStream();
        String strSOAPResponse = "";
        reply.writeTo(os);
        strSOAPResponse = new String(os.toByteArray());
        return strSOAPResponse;
    }
```

In case the Web Servivce is hosted on secured HTTP (HTTPS), you need to add the certificate to the Application Server. If you are using stand alone program the add the following lines:

```java
System.setProperty("com.ibm.ssl.performURLHostNameVerification", "true");
System.setProperty("javax.net.ssl.trustStore","key.jks");
System.setProperty("javax.net.ssl.trustStorePassword", "password");
System.setProperty("javax.net.ssl.trustStoreType","jks");
```

If you are using IBM JDK, add the following

```java
System.setProperty("java.protocol.handler.pkgs","com.ibm.net.ssl.internal.www.protocol");
Security.addProvider(new com.ibm.jsse.IBMJSSEProvider());
```
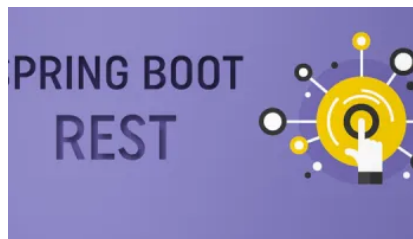
If you are using SUN/Oracle JDK, add the following

```java
System.setProperty("java.protocol.handler.pkgs","com.sun.net.ssl.internal.www.protocol");
Security.addProvider(new com.sun.net.ssl.internal.ssl.Provider());
```
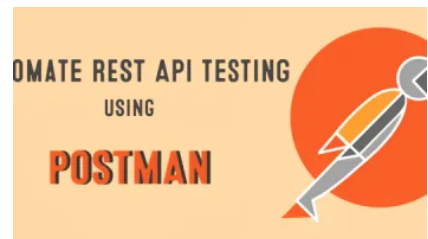
---

## Related

How to configure REST Service (JAX-RS) in IBM Liberty Profile
In "REST"

How to create RESTFul Webservices using Spring Boot
In "Spring Boot"

How to Automate REST API JSON Schema Validation testing using Postman
In "REST"

Filed Under: Java, SAAJ, WebService      Tagged With: Code, Java, Programming, SAAJ, SOAP, WebService

## Subscribe to stay in loop

* indicates required

Email Address *

Subscribe

## Leave a Reply

Logged in as Abhisek Jana. Edit your profile. Log out? Required fields are marked *

Comment *

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.