

Les Services WEB

TP N°3: Développement d'un Handler pour tracer les requêtes/réponses SOAP (suite du TP n°1)

SOMMAIRE

I- Pré-requis :.....	3
I- Objectifs :.....	3
II- Développement du Module	3
a-Création du Module	3
b-Déploiement du Module.....	10
c-Test du Module.....	11

I- Pré-requis :

- Le TP n°1 doit être réalisé avant.

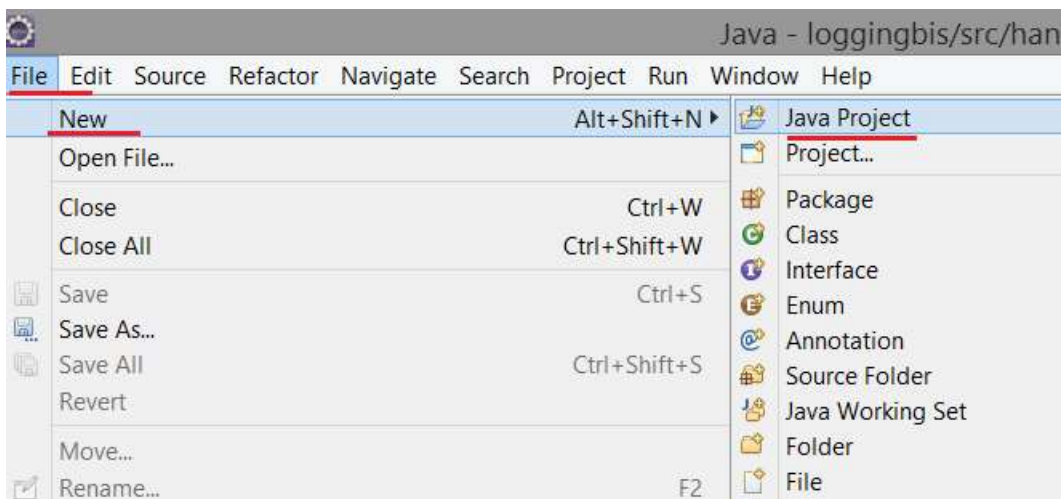
I- Objectifs :

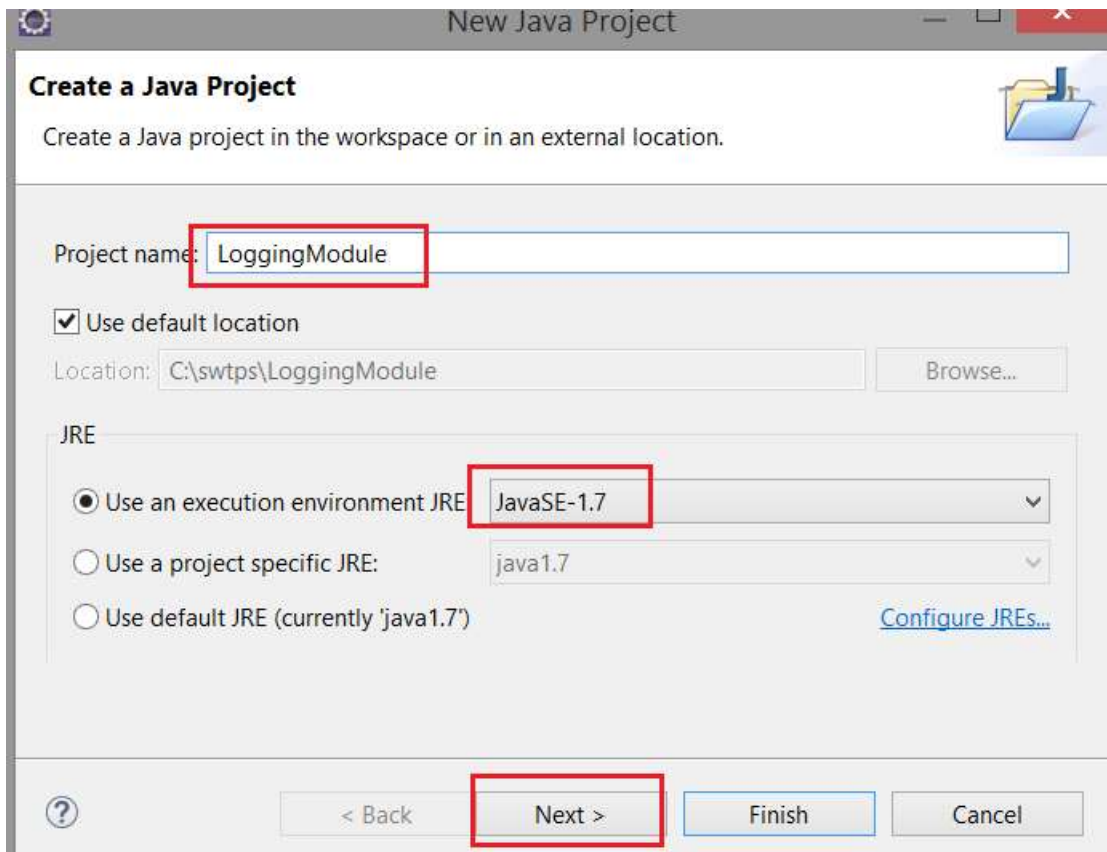
- Créer un nouveau module ;
- Créer un Handler pour tracer toutes les requêtes et les réponses SOAP ;
- Configurer log4J ;
- Intégrer le module au niveau de AXIS2 ;
- Tester et consulter les requêtes et les réponses SOAP.

II- Développement du Module

a-Création du Module

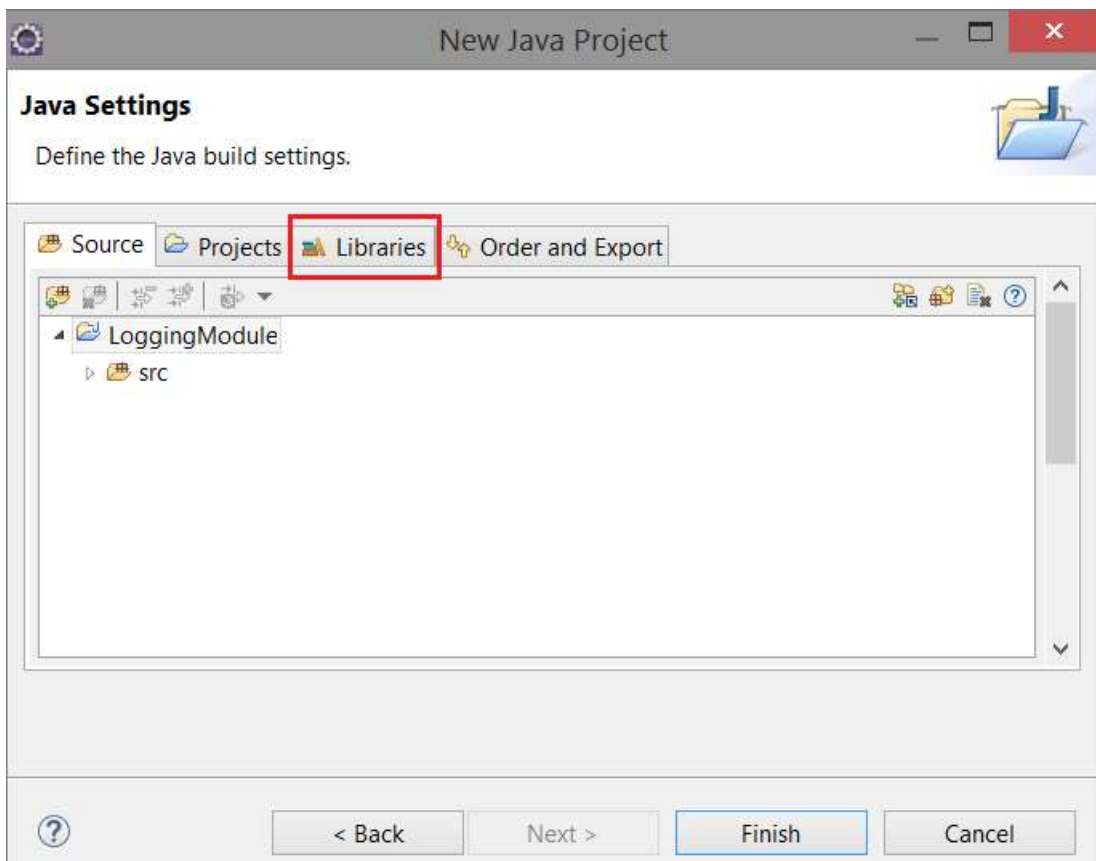
- Créer un nouveau projet java comme suit :



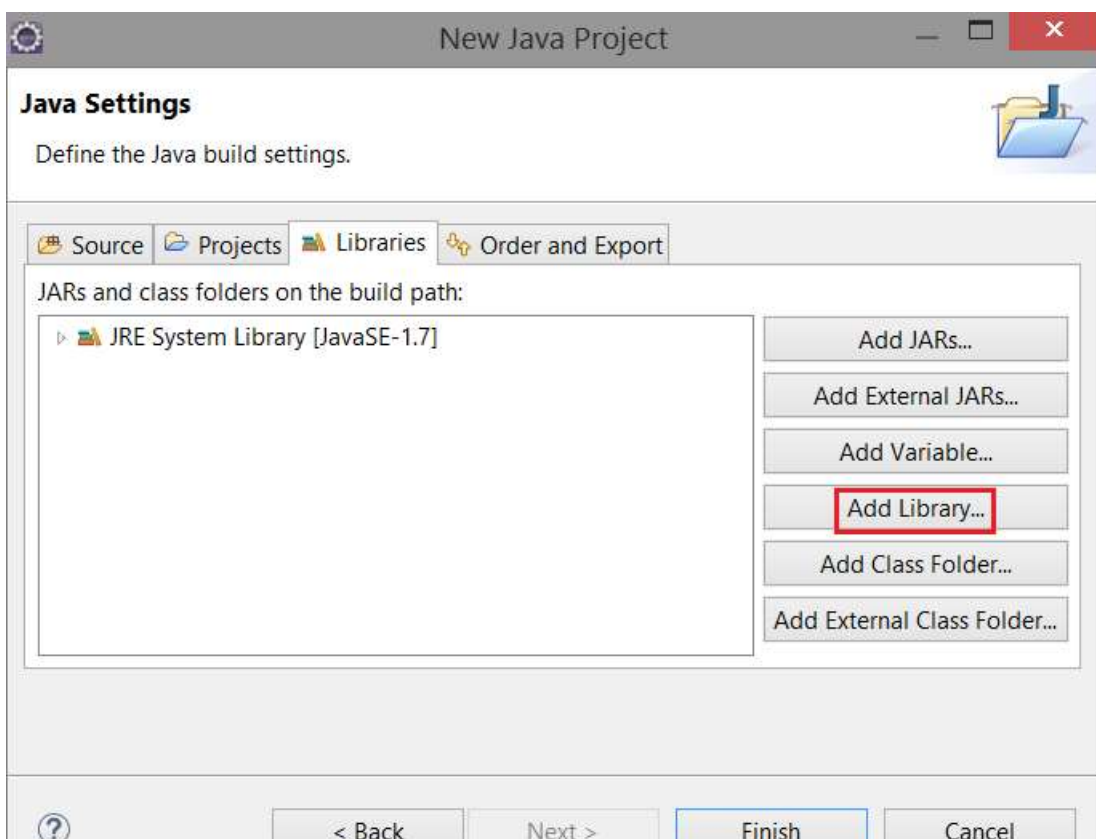


donner comme nom à votre premier projet : LoggingModule

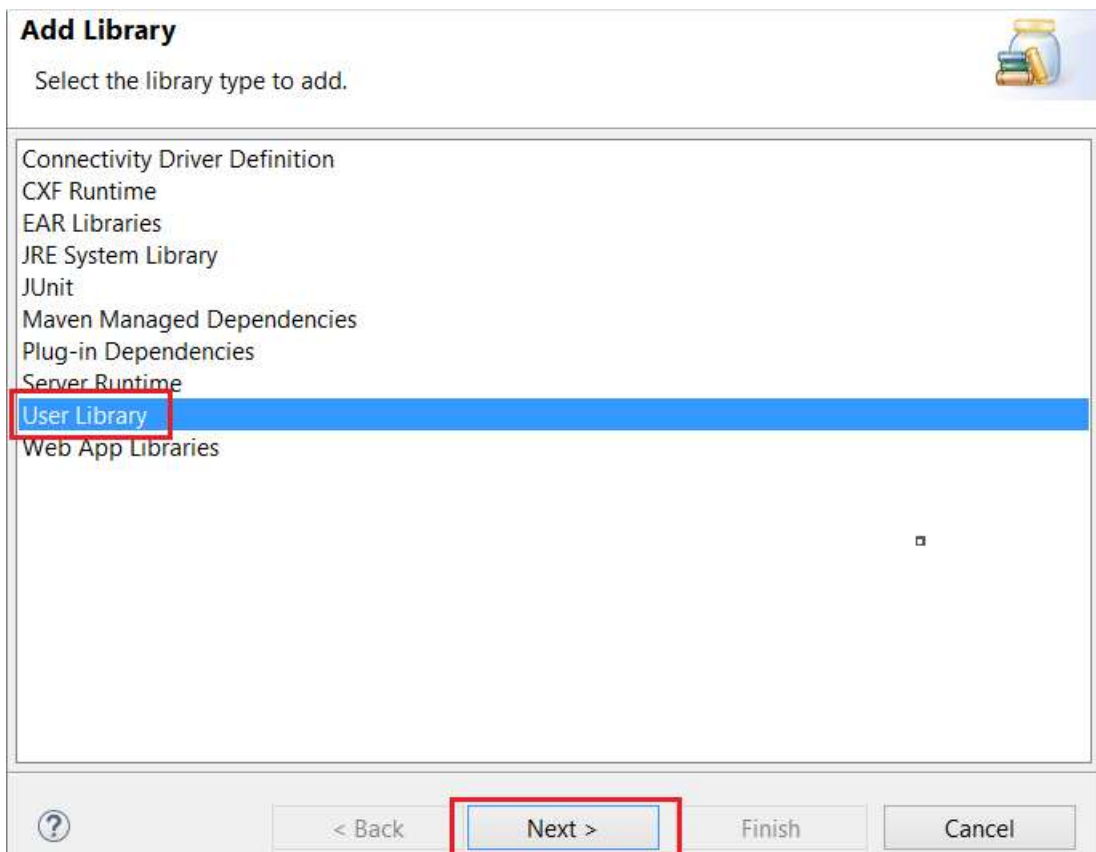
- Cliquer sur **Next>** :



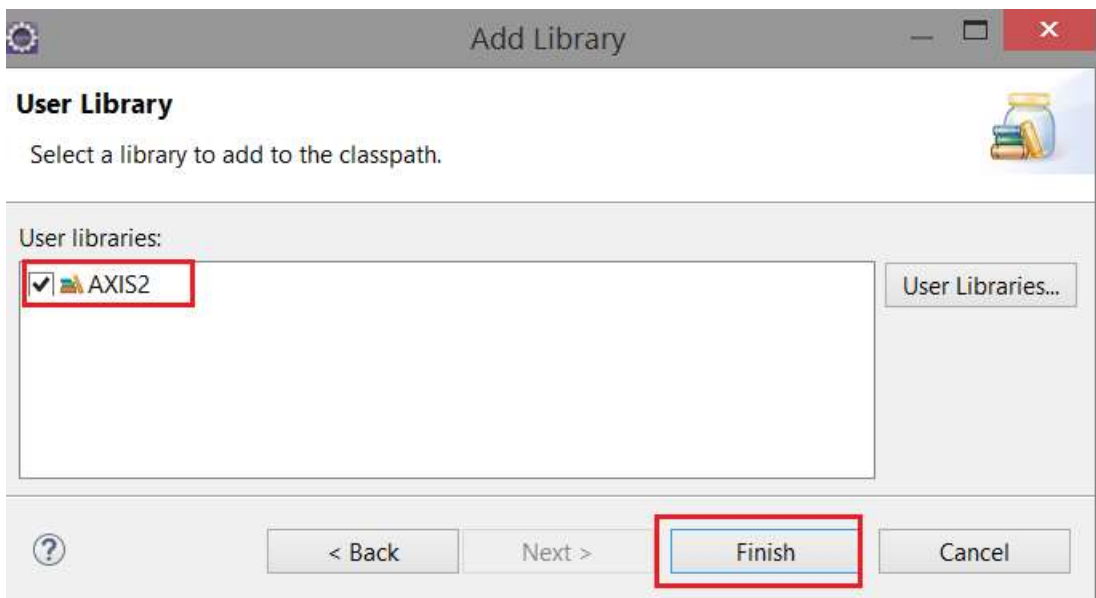
Cliquer sur **Libraries** :



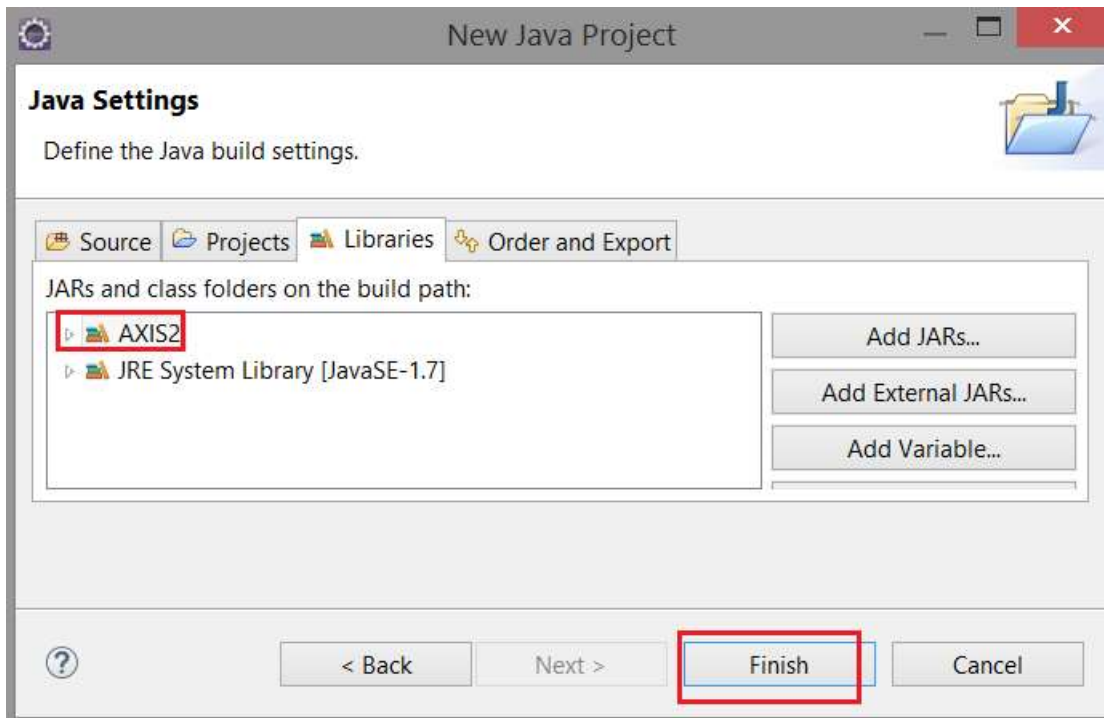
Cliquer sur **"Add Library..."** :



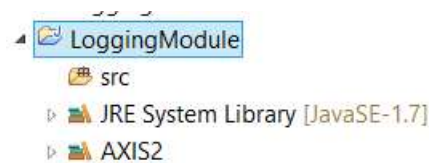
Cliquer sur **Next >** :



Cocher **AXIS2** et cliquer sur **Finish** :



Cliquer sur **Finish** :



- Créer la classe **LoggingModule** suivante :

```
package handler;

import org.apache.axis2.AxisFault;
import org.apache.axis2.context.ConfigurationContext;
import org.apache.axis2.description.AxisDescription;
import org.apache.axis2.description.AxisModule;
import org.apache.axis2.modules.Module;
import org.apache.neethi.Assertion;
import org.apache.neethi.Policy;

public class LoggingModule implements Module {

    // initialize the module
    public void init(ConfigurationContext configContext, AxisModule module) throws AxisFault {
    }

    public void engageNotify(AxisDescription axisDescription) throws AxisFault {
    }

    // shutdown the module
    public void shutdown(ConfigurationContext configurationContext) throws AxisFault {
    }
}
```

```

    public String[] getPolicyNamespaces() {
        return null;
    }

    public void applyPolicy(Policy policy, AxisDescription axisDescription) throws AxisFault {
    }

    public boolean canSupportAssertion(Assertion assertion) {
        return true;
    }
}

```

- Créer la classe **LoggingHandler** suivante :

```

package handler;

import org.apache.axis2.AxisFault;
import org.apache.axis2.context.MessageContext;
import org.apache.axis2.engine.Handler;
import org.apache.axis2.handlers.AbstractHandler;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

public class LoggingHandler extends AbstractHandler implements Handler {
    private static final Log log = LogFactory.getLog("LoggingHandler");
    private String name;

    @Override
    public String getName() {
        return name;
    }

    @Override
    public InvocationResponse invoke(MessageContext msgContext) throws AxisFault {
        log.info("The web service " + msgContext.getAxisService().toString() + ".");
        log.info("The operation is " + msgContext.getAxisOperation().toString() + ".");
        log.info("This is the SOAP envelop : " + msgContext.getEnvelope().toString());
        return InvocationResponse.CONTINUE;
    }

    public void setName(String name) {
        this.name = name;
    }
}

```

- Modifier le fichier log4j.properties (\apache-tomcat-7.0.57\webapps\axis2\WEB-INF\classes) comme suit :

```

#log4j.rootCategory=INFO, CONSOLE
log4j.rootCategory=INFO, CONSOLE, LOGFILE

# Set the enterprise logger priority to FATAL
log4j.logger.org.apache.axis2.enterprise=FATAL
log4j.logger.de.hunsicker.jalopy.io=FATAL
log4j.logger.httpclient.wire.header=FATAL
log4j.logger.org.apache.commons.httpclient=FATAL

# CONSOLE is set to be a ConsoleAppender using a PatternLayout.
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=[%p] %m%n

# LOGFILE is set to be a File appender using a PatternLayout.

```



```
log4j.appender.LOGFILE=org.apache.log4j.FileAppender
log4j.appender.LOGFILE.File=c:\\logs\\axis2.log
log4j.appender.LOGFILE.Append=true
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d [%t] %-5p %c %x - %m%n
```

Observer les deux modifications suivantes :

```
20 # Set root category priority to INFO and its only appender to CONSOLE.
21 #log4j.rootCategory=INFO, CONSOLE
22 log4j.rootCategory=INFO, CONSOLE, LOGFILE
23
24 # Set the enterprise logger priority to FATAL
25 log4j.logger.org.apache.axis2.enterprise=FATAL
26 log4j.logger.de.hunsicker.jalopy.io=FATAL
27 log4j.logger.httpclient.wire.header=FATAL
28 log4j.logger.org.apache.commons.httpclient=FATAL
29
30 # CONSOLE is set to be a ConsoleAppender using a PatternLayout.
31 log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
32 log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
33 log4j.appender.CONSOLE.layout.ConversionPattern=[%p] %m%n
34
35 # LOGFILE is set to be a File appender using a PatternLayout.
36 log4j.appender.LOGFILE=org.apache.log4j.FileAppender
37 log4j.appender.LOGFILE.File=c:\\logs\\axis2.log
38 log4j.appender.LOGFILE.Append=true
```

Le fichier de log **axis2.log** sera placé dans le dossier **c:\logs**

- Créer le dossier META-INF dans **src** :
- Dans ce dossier, créer le fichier **module.xml** :

```
<module name="LoggingModule" class="handler.LoggingModule">
  <InFlow>
    <handler name="InFlowLogHandler" class="handler.LoggingHandler">
      <order phase="LoggingPhase"/>
    </handler>
  </InFlow>

  <OutFlow>
    <handler name="OutFlowLogHandler" class="handler.LoggingHandler">
      <order phase="LoggingPhase"/>
    </handler>
  </OutFlow>

  <OutFaultFlow>
    <handler name="FaultOutFlowLogHandler" class="handler.LoggingHandler">
      <order phase="LoggingPhase"/>
    </handler>
  </OutFaultFlow>

  <InFaultFlow>
    <handler name="FaultInFlowLogHandler" class="handler.LoggingHandler">
      <order phase="LoggingPhase"/>
    </handler>
  </InFaultFlow>
</module>
```

```
</InFaultFlow>
</module>
```

- Lancer l'invite de commande DOS. Aller au dossier C:\swtps\LoggingModule\bin et lancer la commande suivante pour créer le fichier MAR : **jar -cvf LoggingModule.mar ***

b-Déploiement du Module

- Copier le fichier **LoggingModule.mar** dans le dossier C:\serveurs\apache-tomcat-7.0.57\webapps\axis2\WEB-INF\modules.

Maintenant votre module "**LoggingModule**" est prêt pour déploiement.

- Pour ajouter ce module au service **MyFirstWebService**, éditer le fichier **services.xml** et ajouter la ligne suivante `<module ref="LoggingModule"/>`. Le contenu du fichier **services.xml** devient :

```
<service name="Calculator" >
  <Description>
    Service "Calculator" jour le rôle d'une calculatrice
  </Description>
  <module ref="LoggingModule"/>
  <messageReceivers>
    <messageReceiver mep="http://www.w3.org/2004/08/wSDL/in-only"
class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver" />
    <messageReceiver mep="http://www.w3.org/2004/08/wSDL/in-out"
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>
  </messageReceivers>
  <parameter name="ServiceClass" locked="false">ma.formation.sw.Calculator</parameter>
</service>
```

- Editer le fichier **axis2.xml** (\apache-tomcat-7.0.57\webapps\axis2\WEB-INF\conf) et ajouter la phase **LoggingPhase** pour les 04 phases comme expliqué ci-dessous :

Pour la phase **InFlow** :

```
<phaseOrder type="InFlow">
  <!-- System predefined phases -->
  <phase name="Transport">□
  <phase name="Addressing">□
  <phase name="Security" />
  <phase name="LoggingPhase" />
  <phase name="PreDispatch" />
```

Pour la phase **OutFlow** :

```

<phaseOrder type="OutFlow">
  <!-- user can add his own phases to this area -->
  <phase name="soapmonitorPhase" />
  <phase name="OperationOutPhase" />
  <!--system predefined phase -->
  <!--these phase will run irrespective of the service -->
  <phase name="RMPhase" />
  <phase name="PolicyDetermination" />
  <phase name="MessageOut" />
  <phase name="Security" />
  <phase name="LoggingPhase" />

```

Pour la phase **InFaultFlow** :

```

<phaseOrder type="InFaultFlow">
  <phase name="Addressing">□
  <phase name="Security" />
  <phase name="LoggingPhase" />
  <phase name="PreDispatch" />
  <phase name="Dispatch" class="org.apache.axis2.engine.DispatchPhase">□
  <phase name="RMPhase" />
  <!-- user can add his own phases to this area -->
  <phase name="OperationInFaultPhase" />
  <phase name="soapmonitorPhase" />

```

Pour la phase **OutFaultFlow** :

```

<phaseOrder type="OutFaultFlow">
  <!-- user can add his own phases to this area -->
  <phase name="soapmonitorPhase" />
  <phase name="OperationOutFaultPhase" />
  <phase name="RMPhase" />
  <phase name="PolicyDetermination" />
  <phase name="MessageOut" />
  <phase name="Security" />
  <phase name="LoggingPhase" />

```

```

</phaseOrder>

```

c-Test du Module

- Démarrer le serveur (startup.bat).
- Lancer le classe **Test** du projet **MyFirstWebServiceClient** et observer la requête et la réponse SOAP pour la méthode add :

```

public class Test {
    public static void main(String[] args) {
        double a = 12.45;
        double b = 5.7;
        try {
            CalculatorStub stub = new CalculatorStub();

            // Tester la méthode add
            Add add = new Add();
            add.setA(a);
            add.setB(b);
            AddResponse ar = stub.add(add);
            System.out.println(a + "+" + b + " = " + ar.get_return());
        }
    }
}

```

La requête SOAP correspondante :

```

<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
    <soapenv:Header />
    <soapenv:Body>
        <ns2:add xmlns:ns2="http://sw.formation.ma">
            <ns2:a>12.45</ns2:a>
            <ns2:b>5.7</ns2:b>
        </ns2:add>
    </soapenv:Body>
</soapenv:Envelope>

```

La réponse SOAP correspondante :

```

<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
    <soapenv:Header />
    <soapenv:Body>
        <ns:addResponse xmlns:ns="http://sw.formation.ma">
            <ns:return>18.15</ns:return>
        </ns:addResponse>
    </soapenv:Body>
</soapenv:Envelope>

```

Le contenu du fichier **axis2.log** est :

```

2016-02-22 22:02:31,195 [localhost-startStop-1] INFO org.apache.axis2.deployment.ClusterBuilder - Clustering has been disabled
2016-02-22 22:02:31,367 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: addressing-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/addressing-1.6.4.mar
2016-02-22 22:02:31,376 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: jaxws-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/axis2-jaxws-mar-1.6.4.mar
2016-02-22 22:02:31,386 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: LoggingModule - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/LoggingModule.mar
2016-02-22 22:02:31,413 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: metadataExchange-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/mex-1.6.4.mar
2016-02-22 22:02:31,424 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: mtompolicy-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/mtompolicy-1.6.4.mar
2016-02-22 22:02:31,434 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: ping-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/ping-1.6.4.mar
2016-02-22 22:02:31,444 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: script-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/scripting-1.6.4.mar
2016-02-22 22:02:31,455 [localhost-startStop-1] INFO org.apache.axis2.deployment.ModuleDeployer - Deploying module: soapmonitor-1.6.4 - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/modules/soapmonitor-1.6.4.mar
2016-02-22 22:02:31,610 [localhost-startStop-1] INFO org.apache.axis2.deployment.DeploymentEngine - Deploying Web service: Calculator.aar - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/services/Calculator.aar
2016-02-22 22:02:31,622 [localhost-startStop-1] INFO org.apache.axis2.deployment.DeploymentEngine - Deploying Web service: version-1.6.4.aar - file:/C:/serveurs/apache-tomcat-7.0.57/webapps/axis2/WEB-INF/services/version-1.6.4.aar
2016-02-22 22:03:28,682 [http-apr-8080-exec-2] INFO LoggingHandler - The web service Calculator.

```

```

2016-02-22 22:03:28,682 [http-apr-8080-exec-2] INFO LoggingHandler - The operation is
org.apache.axis2.description.InOutAxisOperation@2bbb63e5.
2016-02-22 22:03:28,730 [http-apr-8080-exec-2] INFO LoggingHandler - This is the SOAP envelop : <?xml version='1.0'
encoding='utf-8'?><soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-
envelope"><soapenv:Header/><soapenv:Body><ns2:add
xmlns:ns2="http://sw.formation.ma"><ns2:a>12.45</ns2:a><ns2:b>5.7</ns2:b></ns2:add></soapenv:Body></soapenv:Envelope>
2016-02-22 22:03:28,750 [http-apr-8080-exec-2] INFO LoggingHandler - The web service Calculator.
2016-02-22 22:03:28,750 [http-apr-8080-exec-2] INFO LoggingHandler - The operation is
org.apache.axis2.description.InOutAxisOperation@2bbb63e5.
2016-02-22 22:03:28,750 [http-apr-8080-exec-2] INFO LoggingHandler - This is the SOAP envelop : <?xml version='1.0'
encoding='utf-8'?><soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-
envelope"><soapenv:Header/><soapenv:Body><ns:addResponse
xmlns:ns="http://sw.formation.ma"><ns:return>18.15</ns:return></ns:addResponse></soapenv:Body></soapenv:Envelope>

```