

Les Services WEB

TP N°2: Utilisation du plugin "TCP/IP Monitor" pour visualiser les messages SOAP

SOMMAIRE

I- Objectifs :.....	3
II- Pré-requis :.....	3
III- Configuration de TCP/IP Monitor	3
IV- Tester.....	6

I- Objectifs :

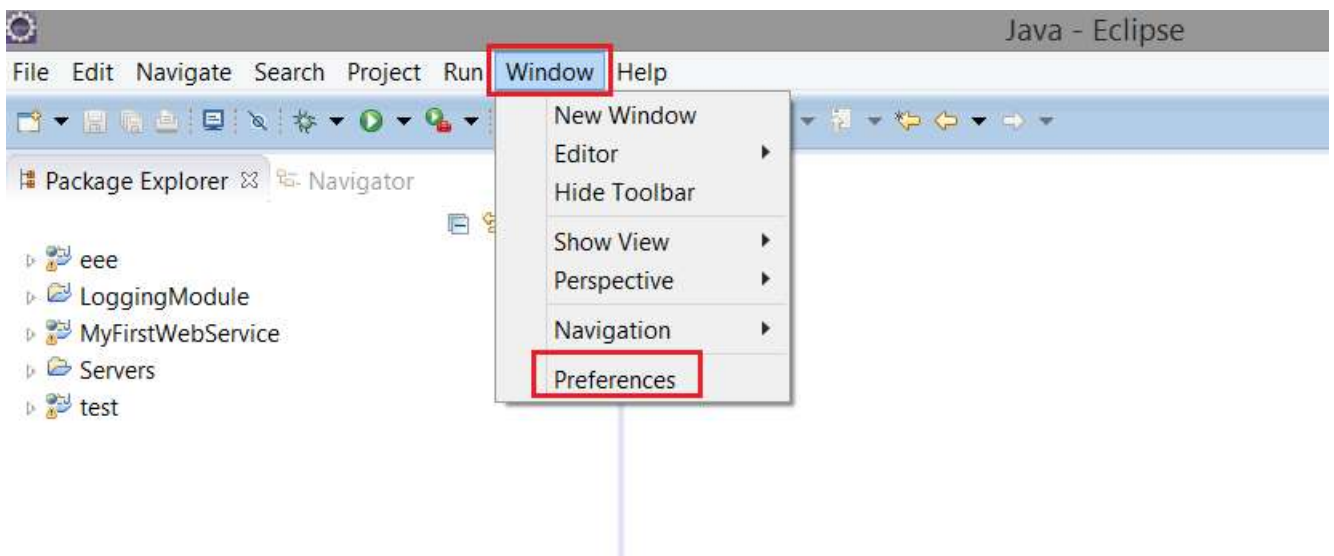
- Configurer le plugin TCP/IP Monitor ;
- Visualiser les requêtes et les réponses SOAP.

II- Pré-requis :

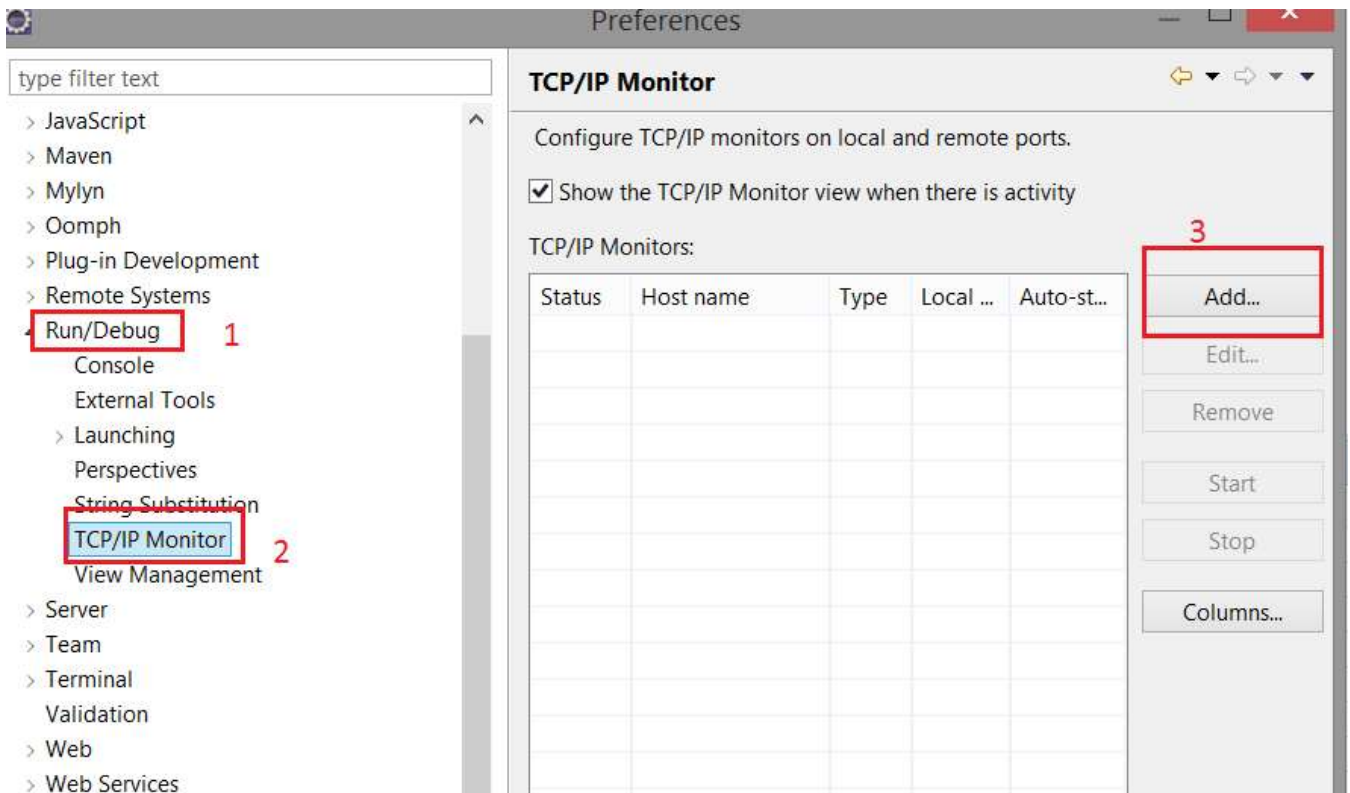
Il faut faire tout d'abord le TP n°1 afin de pouvoir tester le plugin TCP/IP Monitor.

III- Configuration de TCP/IP Monitor

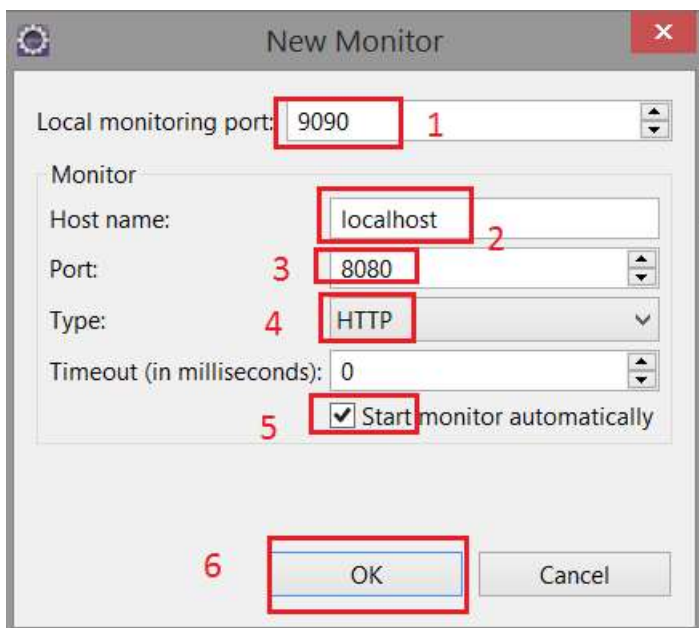
Le plugin TCP/IP Monitor est accessible au niveau de Eclipse Mars moyennant le menu suivant :



- Cliquer sur **Preferences** :

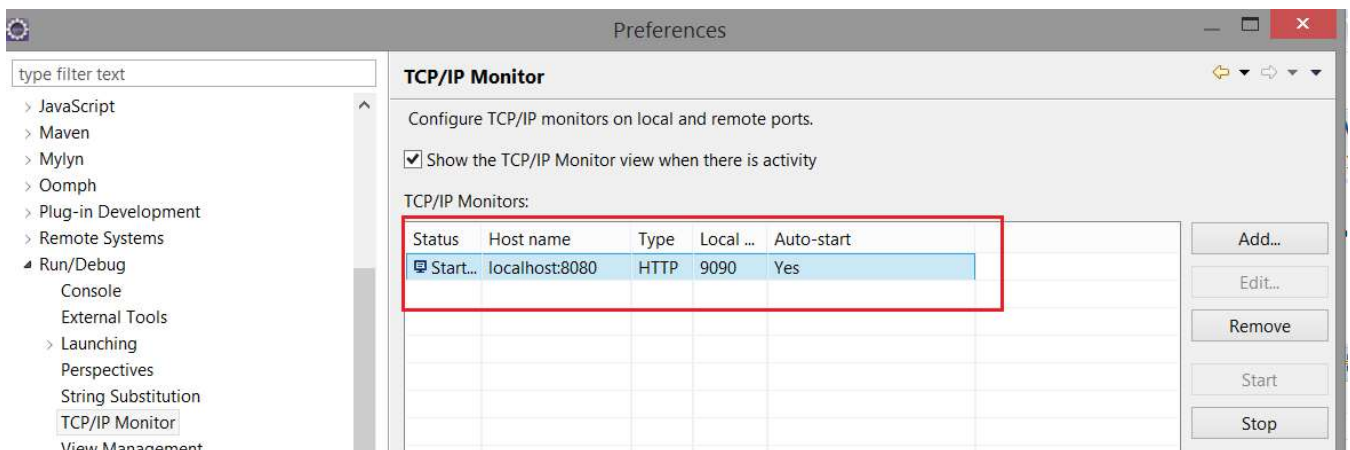


- Cliquer sur **Add...** :

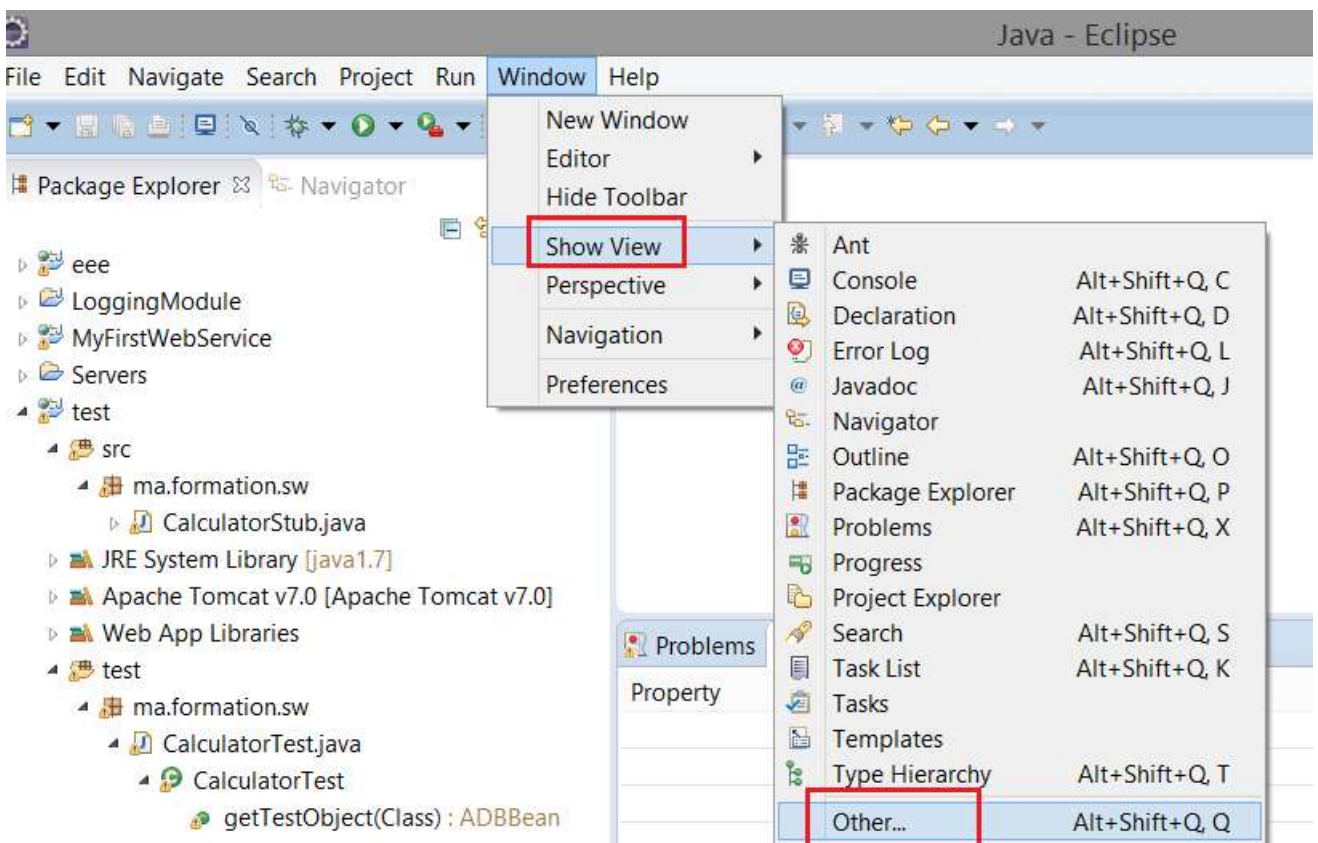


- Saisir 9090 par exemple dans le port qui sera utilisé par TCP/IP Monitor. Il va falloir modifier l'URL au niveau de la classe stub du client (remplacer 8080 par 9090) pour que ce dernier envoie les requêtes à TCP/IP Monitor (voir après).
- Mettre **localhost** dans **Host name**.
- Mettre 8080 dans le **Port**.
- Cocher **Start monitor automatically**.
- Enfin, cliquer sur **OK**.

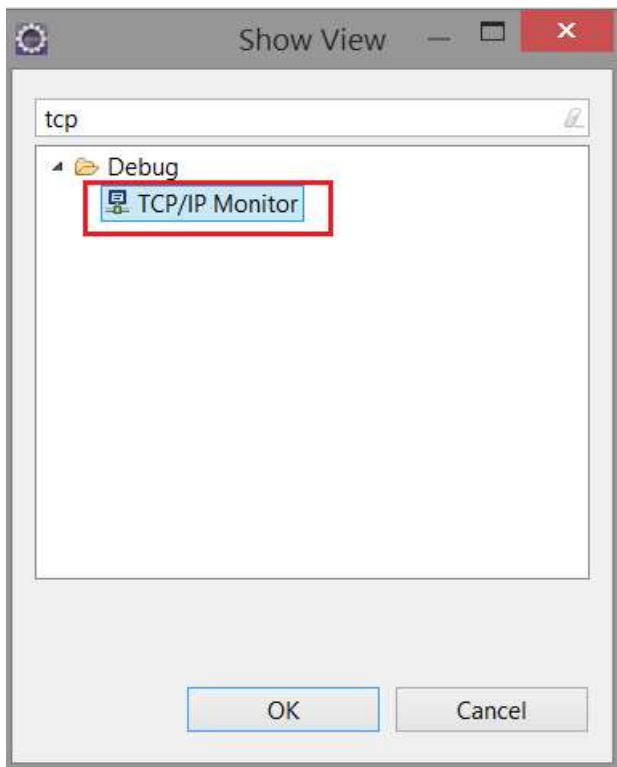
Le moniteur est ajouté comme illustré ci-dessous :



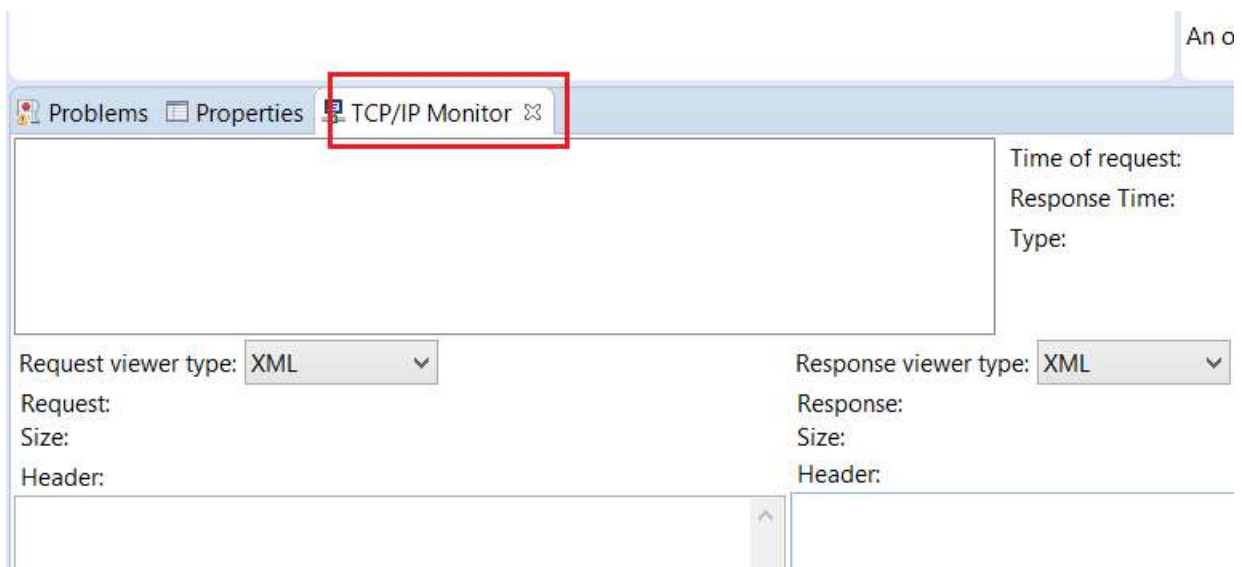
- Ajouter la vue TCP/IP Monitor à la perspective JAVA comme montré ci-dessous :



- Cliquer sur **Other...** :



- Chercher **TCP/IP Monitor**, ensuite cliquer sur **OK**. La vue TCP/IP Monitor sera ajouté au niveau d'Eclipse :



IV- Tester

- Dans le WS client concernant le WS "Calculator" réalisé dans le TP1, éditer la classe **CalculatorStub** et remplacer 8080 par 9090.
- Tester par exemple la méthode add :

```

public void testadd() throws java.lang.Exception {

    ma.formation.sw.CalculatorStub stub = new ma.formation.sw.CalculatorStub();

    ma.formation.sw.CalculatorStub.Add add14 =
(ma.formation.sw.CalculatorStub.Add) getTestObject(
        ma.formation.sw.CalculatorStub.Add.class);
    // TODO : Fill in the add14 here
    add14.setA(14);
    add14.setB(20);
    // assertNotNull(stub.add(
    // add14));
    System.out.println("la somme de 14 et 20 est:" +
stub.add(add14).get_return());

}

```

- Observer le résultat suivant :

The screenshot shows the TCP/IP Monitor tool interface. The left pane displays the request details for a POST request to `/axis2/services/Calculator.CalculatorHttpSoap11Endpoint/` on `localhost:9090`. The request is in XML format, with a size of 287 (504) bytes. The XML body is highlighted with a red box and labeled "Requête SOAP".

The right pane displays the response details for the same request, received from `localhost:8080`. The response is in XML format, with a size of 273 (434) bytes. The response time is 3 ms. The XML body is highlighted with a red box and labeled "Réponse SOAP".

Requête SOAP

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns4:add xmlns:ns4="http://sw.formation.ma">
      <ns4:a>14.0</ns4:a>
      <ns4:b>20.0</ns4:b>
    </ns4:add>
  </soapenv:Body>
</soapenv:Envelope>

```

Réponse SOAP

```

<?xml version="1.0" encoding="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:addResponse xmlns:ns="http://sw.formation.ma">
      <ns:return>34.0</ns:return>
    </ns:addResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

TCP/IP Monitor permet de visualiser les requêtes SOAP ainsi que les réponses SOAP.