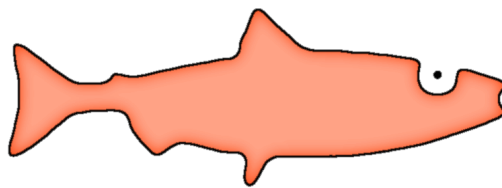


SALMon

Reference Manual

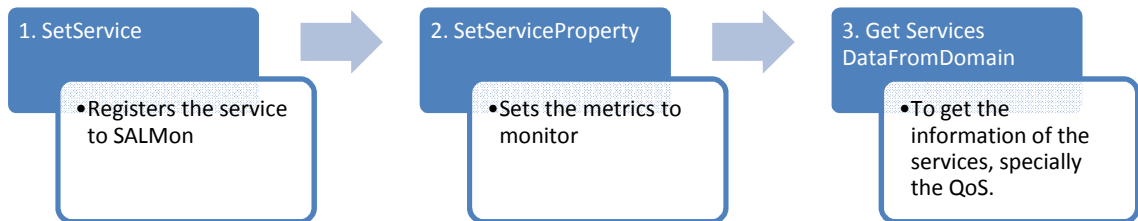


Based on v.0.4

Reference Manual

In this tutorial We are going to set two Services of the “Currency” domain (services that transform from one currency to another) and get their current availability and response time.

The workflow in order to do it is the following.



1. SetService()

Used to register the services to monitor. The information provided is the following:

Name: Name of the service

Domain: The domain to which the service belongs. (e.g: weather forecasting, currency, billing,...)

WSDL: the URL of the WSDL file

Description: A brief description of the service.

User and Password: If authentication is required, the user and password should be set, otherwise they can be blank.

URL and Port: URL and Port of the Endpoint to access the service.

MeasureInformation: It is used to set the frequency in which the service must be tested (monitoring interval), and also the timeout. (both parameters in seconds)

Example - Service1:

<http://www.websvc.com/currencyconvertor.asmx?WSDL>

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
  <q0:SetServiceRequest>
  <q0:service xsi:type="q0:WebServiceInformation">
    <q0:name>Currency convertor</q0:name>
    <q0:description>Get conversion rate from one currency to another currency</q0:description>
```

```

<q0:domain>Currency</q0:domain>
<q0:password></q0:password>
<q0:port>80</q0:port>
<q0:URL>http://www.websvc.com/currencyconvertor.asmx</q0:URL>
<q0:user></q0:user>
<q0:measureInformation>
  <q0:monitoringInterval>60</q0:monitoringInterval>
  <q0:timeOut>30</q0:timeOut>
</q0:measureInformation>
<q0:wsdl>http://www.websvc.com/currencyconvertor.asmx?WSDL</q0:wsdl>
</q0:service>
</q0:SetServiceRequest>
</soapenv:Body>
</soapenv:Envelope>

```

Example - Service2:

<http://www.exporttechnologies.com/webservices/currency/currency.service.asmx?wsdl>

```

<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
  <q0:SetServiceRequest>
    <q0:service xsi:type="q0:WebServiceInformation">
      <q0:name>Currency service</q0:name>
      <q0:description>Get conversion rate from one currency to another currency</q0:description>
      <q0:domain>Currency</q0:domain>
      <q0:password></q0:password>
      <q0:port>80</q0:port>
    <q0:URL>http://www.exporttechnologies.com/webservices/currency/currency.service.asmx</q0:URL>
    <q0:user></q0:user>
    <q0:measureInformation>
      <q0:monitoringInterval>60</q0:monitoringInterval>
      <q0:timeOut>30</q0:timeOut>
    </q0:measureInformation>
    <q0:wsdl>http://www.exporttechnologies.com/webservices/currency/currency.service.asmx?WSDL</q0:wsdl>
  </q0:service>
  </q0:SetServiceRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

2. SetServiceProperty()

To start the monitoring or testing of a concrete metric. The information provided is the following:

Metric: Identifies the metric to *monitor* (*currently works for CurrentAvailability and CurrentResponseTime*)

serviceID: The identifier of the service to monitor

operationID: The identifier of a concrete operation to monitor (for the whole service it should be set to 0).

(Currently metrics for specific operations are not available in the stable version)

Example - Metric1: Current Response Time of Service1.

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
</soapenv:Header>
  <soapenv:Body>
<q0:SetServicePropertyRequest>
  <q0:metric xsi:type="q0:CurrentResponseTime"/>
  <q0:serviceID>1</q0:serviceID>
  <q0:operationID>0</q0:operationID>
</q0:SetServicePropertyRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Example - Metric2: Current Availability of Service1.

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
</soapenv:Header>
  <soapenv:Body>
<q0:SetServicePropertyRequest>
  <q0:metric xsi:type="q0:CurrentAvailability"/>
  <q0:serviceID>1</q0:serviceID>
  <q0:operationID>0</q0:operationID>
</q0:SetServicePropertyRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Example - Metric3: Current Response Time of Service2.

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
</soapenv:Header>
  <soapenv:Body>
<b><q0:SetServicePropertyRequest>
  <q0:metric xsi:type="q0:CurrentResponseTime"/>
  <q0:serviceID>2</q0:serviceID>
  <q0:operationID>0</q0:operationID>
</q0:SetServicePropertyRequest>
</soapenv:Body>
</soapenv:Envelope>
```

Example - Metric4: Current Availability of Service4.

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.edu/salmon/services/Monitor"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
</soapenv:Header>
  <soapenv:Body>
<b><q0:SetServicePropertyRequest>
  <q0:metric xsi:type="q0:CurrentAvailability"/>
  <q0:serviceID>2</q0:serviceID>
  <q0:operationID>0</q0:operationID>
</q0:SetServicePropertyRequest>
</soapenv:Body>
</soapenv:Envelope>
```

3. Get Services Data From Domain

To get the information of the services and their QoS. The information provided is the following:

Domain: The domain to which the service belongs. In the given example is Currency.

Example:

```
<soapenv:Envelope
  xmlns:q0="http://appserv.lsi.upc.es/salmon/schemas/WSConnectionService_schema.xsd"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
  <q0:GetServicesDataFromDomainRequest>
    <domain>Currency</domain>
  </q0:GetServicesDataFromDomainRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

The result should be something like this:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<GetServicesDataFromDomainResponse
  xmlns:ns1="http://appserv.lsi.upc.es/salmon/schemas/WSConnectionService_schema.xsd">
<servicesData>
  <name>Currency convertor</name>
  <wsdl>http://www.webservicex.com/currencyconvertor.asmx?WSDL</wsdl>
  <endPoint>http://www.webservicex.com/currencyconvertor.asmx</endPoint>
  <description>Get conversion rate from one currency to another currency</description>
<metricValues>
  <metric>CurrentResponseTime</metric>
  <value>799</value>
  </metricValues>
<metricValues>
  <metric>CurrentAvailability</metric>
  <value>1</value>
  </metricValues>
</servicesData>
<servicesData>
  <name>Currency service</name>
  <wsdl>http://www.exporttechnologies.com/webservices/currency/currency/currency.service.asmx?WSDL</wsdl>
  <endPoint>http://www.exporttechnologies.com/webservices/currency/currency/currency.service.asmx</endPoint>
  <description>Get conversion rate from one currency to another currency</description>
<metricValues>
  <metric>CurrentResponseTime</metric>
  <value>238</value>
  </metricValues>
<metricValues>
  <metric>CurrentAvailability</metric>
  <value>1</value>
  </metricValues>
</servicesData>
```

```
</GetServicesDataFromDomainResponse>  
</soapenv:Body>  
</soapenv:Envelope>
```

This is just an example,
feel free to test SALMon with other Services of different Domains ☺