

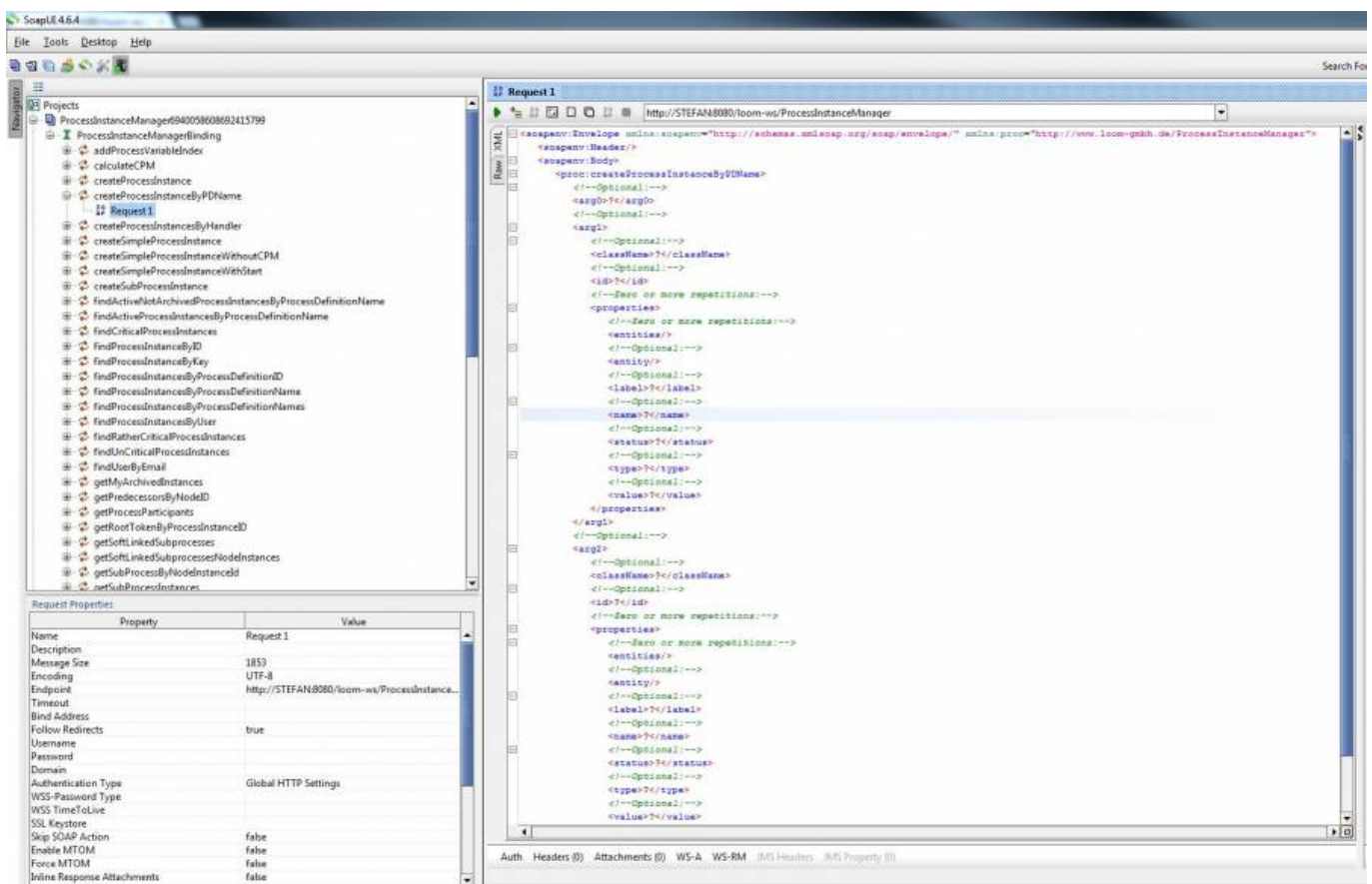
## WSDL

Um die aktuelle WSDL auszulesen, kann folgende Datei genutzt werden:

```
<JBOSSPFAD>/server/default/data/wsd/loom.ear/loom-  
ejb.jar/ProcessInstanceManager<Zahl>.wsdl
```

Da bei jedem Neustart eine neue WSDL generiert wird, muss unbedingt die aktuellste WSDL des ProcessInstanceManagers eingelesen werden! Um eine Prozessinstanz zu starten, wird die Methode createProcessInstanceByPDName benötigt.

Für diese Dokumentation wird im weiteren Verlauf mit SoapUI gearbeitet.



die Struktur für den Aufruf angezeigt. Folgende Parameter werden erwartet:

- Prozessdefinitionname
- WSEntity (Hier wird später der Prozessname enthalten sein, falls dieser gesetzt werden soll)
- WSEntity (Hier werden die Prozessvariablen und dazugehörigen Werte enthalten sein)

## Aufbau des XML zur Übertragung

Folgende Struktur muss eingehalten werden, um einen entsprechenden Prozess zu starten. Alle Werte

in Großbuchstaben müssen nach dem entsprechenden Wünschen angepasst werden.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<loom:createProcessInstanceByPDName
xmlns:loom="http://www.loom-gmbh.de/ProcessInstanceManager">
<arg0>PROZESSDEFINITIONSNAME</arg0>
<arg1>
  <id>0</id>
  <className>ProcessInstance</className>
  <properties>
    <name>name</name>
    <value>PROZESSINSTANZNAME</value>
  </properties>
</arg1>
<arg2>
  <id>0</id>
  <className>ProcessVariable</className>
  <properties>
    <name>PROZESSVARIABLENAME1</name>
    <value>PROZESSVARIABLEWERT1</value>
  </properties>
  <properties>
    <name>PROZESSVARIABLENAME2</name>
    <value>PROZESSVARIABLEWERT2</value>
  </properties>
</arg2>
</loom:createProcessInstanceByPDName>
</soap:Body>
</soap:Envelope>
```

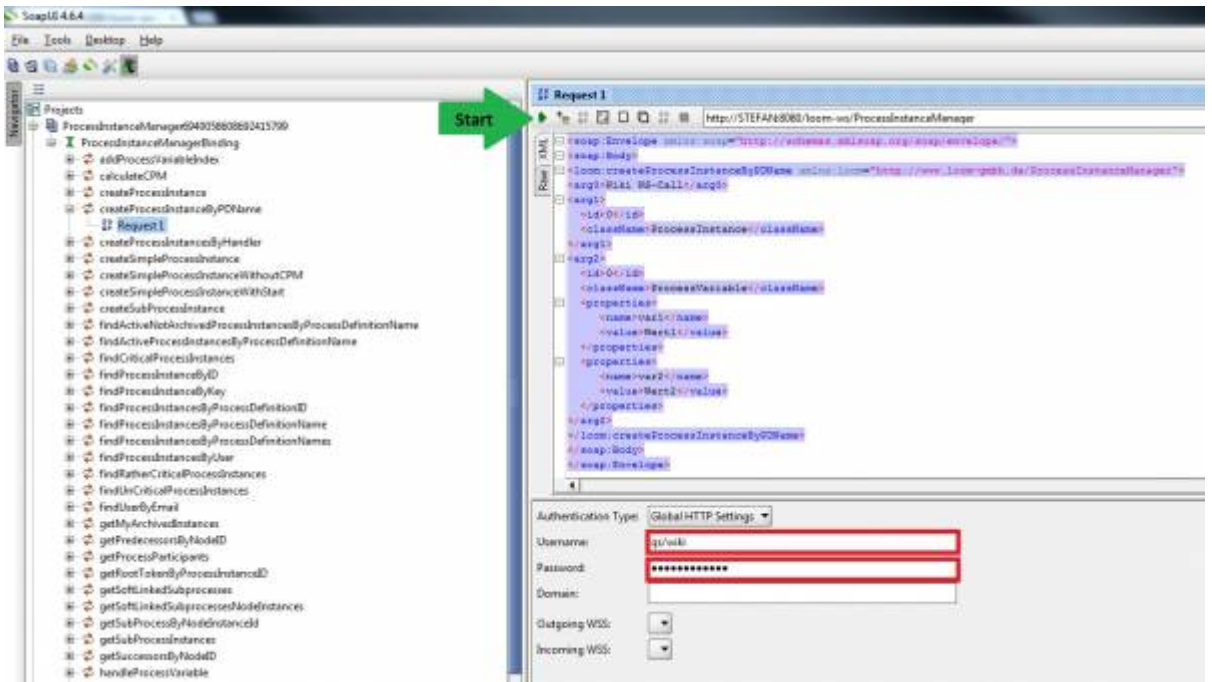
Optional kann der Teil auch entfernt werden, falls im Prozess bereits ein [RenameInstanceHandler](#) eingesetzt wird.

```
<properties>
  <name>name</name>
  <value>PROZESSINSTANZNAME</value>
</properties>
```

--

## Prozessinstanz aus SoapUI starten

Aufbauend auf der Struktur kann nun ein Webservice call an TIM - Task In Motion geschickt werden. Hierzu muss im rechten Bereich von SoapUI das XML eingefügt werden,



welches an den Webservice geschickt werden soll. In diesem Testfall werden wir den Prozess „Wiki WS-Call“ starten und unten stehenden Prozessvariablen anlegen, die nach dem Start in der Smartform ersichtlich sind.

Prozessvariablen	
Name der Variable	Wert der Variable
var1	Wert1
var2	Wert2

Hierzu befüllen wir das XML entsprechend und kopieren es in den Rechten Bereich von Soap UI. Danach muss noch im Reiter **Auth** der Benutzer + Passwort eingegeben werden, der berechtigt ist den Prozess zu starten. Ist alles korrekt befüllt worden, kann über den „Play“-Button der Call ausgeführt werden und bekommen ein XML als Antwort, in dem die Antwort von TIM - Task In Motion hinterlegt ist.

## Antwort und Überprüfung



Wenn alles korrekt verarbeitet werden konnte gibt TIM - Task !n Motion die Informationen des gestarteten Prozesses zurück, die für die weitere Verarbeitung auf dem aufrufenden Server genutzt werden kann. Loggt sich der User nun entsprechend in TIM - Task !n Motion ein, kann man unter **Meine Instanzen** die gestartete Instanz mit den entsprechenden Prozessvariablen einsehen.

From: <https://wiki.tim-solutions.de/> - **TIM Wiki** / **[NEW TIM 6 Documentation](#)**

Permanent link: **<https://wiki.tim-solutions.de/doku.php?id=software:tim:webservices:startpibypdname&rev=1392203110>**

Last update: **2021/07/01 09:59**

