

# Oracle SALT を使用して Tuxedo サービスを SOAP Web サービスとして公開する方法



## 概要

このドキュメントは、Oracle Service Architecture Leveraging Tuxedo (Oracle SALT) のユースケースをほんの数分で実装できるように作成されています。Oracle SALT を使用すると、プロジェクトをゼロからブートストラップし、既存のプロジェクトに Web サービスの機能を迅速に追加し、製品のプロトタイプ化と評価を実行できます。

この手順書は、TOUPPER や環境内の既存のアプリケーションなど、簡単にあらゆるサービスに置き換えられるサービス例を中心として作られています。

## 前提条件：

- Oracle Tuxedo と Oracle SALT 12.2.2 がインストールされていること。Tuxedo は[こちら]からダウンロードしてください。  
<http://www.oracle.com/technetwork/jp/middleware/tuxedo/downloads/index.html>
- Tuxedo [RP004](#) および Oracle SALT [RP002](#)
- uBike サーバーで Tuxedo ドメインを構成すること。uBike.zip ファイルはこちらからダウンロードしてください。  
<http://www.oracle.com/technetwork/indexes/samplecode/tuxedo-sample-522120.html>
- 環境変数を Tuxedo とドメインに設定すること (TUXDIR、APPDIR、SALTCONFIG)

## GWWS、TMADMSVR、およびTMMETADATAをUBBCONFIGに追加する

Oracle SALT をまだ有効にしていない場合は、UBB ファイルを編集して次の行を追加し、有効にします。

```
TMMETADATA  SRVGRP=GROUP1 SRVID=2
              CLOPT="-A -- -f meta.repos"
TMADMSV     SRVGRP=GROUP1 SRVID=3
              CLOPT="-A -- -a http://localhost:4011"
GWW         SRVGRP=GROUP1 SRVID=4
              CLOPT="-A -- -i GWWS1"
```

必要に応じて、'localhost'を環境に適したホスト名または IP アドレスに置き換えます。また 4011 のポートがすでにマシンで使用されている場合は、別のポートを使用します。

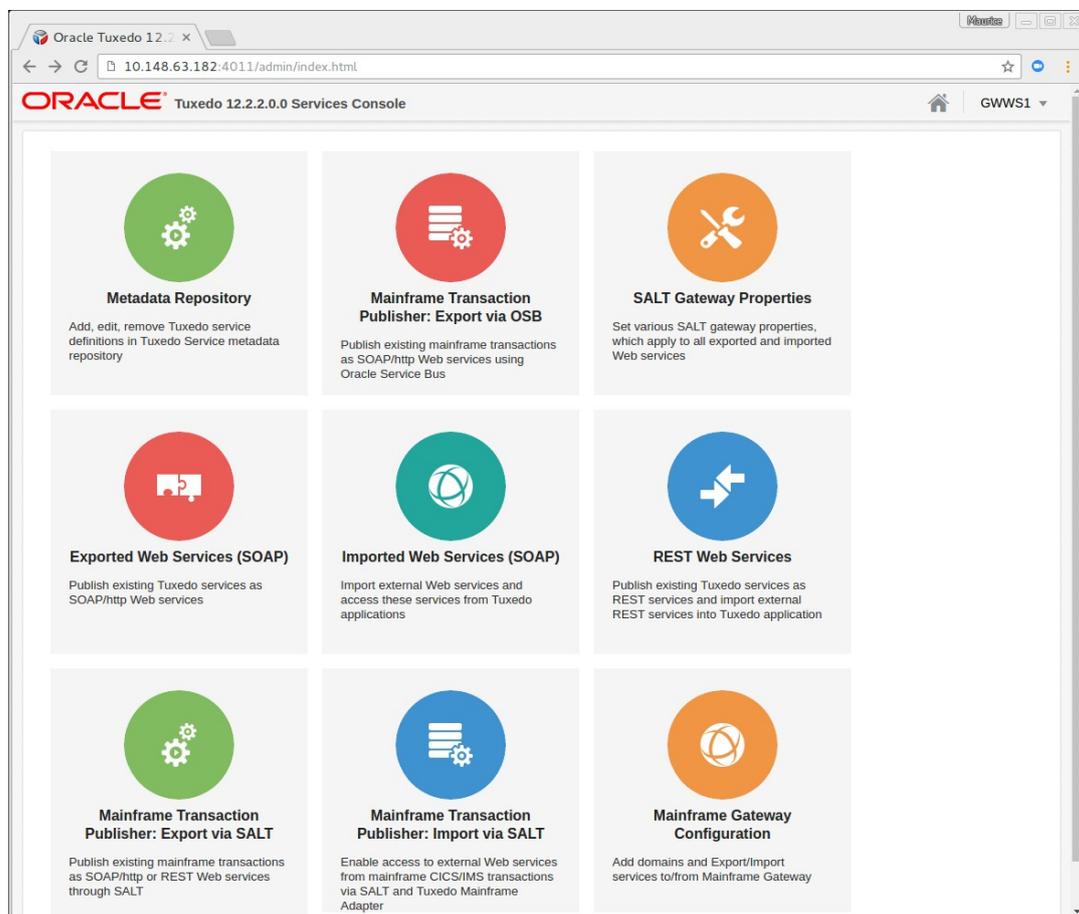
tuxconfig を再ロードして、次のようなコマンドを使用してブートします。

```
$ tmloadcf -y ubbconfig
$ tmboot -y
```

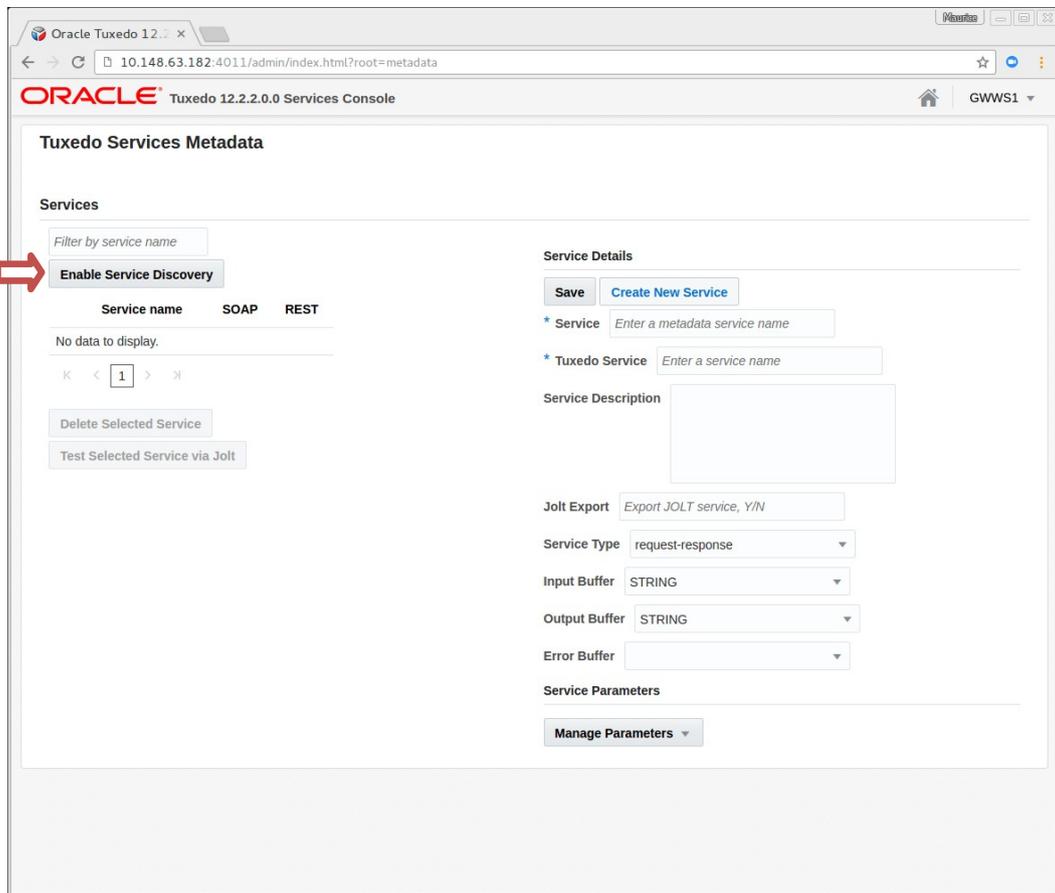
SALTCONFIG ファイルを作成する必要はありません。GWWS によって自動的に生成されます。

## Tuxedoサービス・コンソールにログインして、サービス契約の検出をアクティブ化する

ブラウザで次の URL に移動します。http://localhost:4011/admin



Metadata Repository に移動して、「Enable Service Discovery」をクリックします。



これで、Tuxedo サービス・メタデータ・リポジトリの自動移入がアクティブ化されます。Tuxedo サービスを Web サービスとして公開するには、そのサービス・メタデータが存在し、適切な WSDL が生成できるようになっている必要があります。

## 一般的なサービス・コールを実行してサービス・インタフェースを検出する（契約）

Oracle SALT では、サービス検出をオンにしてアプリケーションをコールすると、このコールの間にやり取りされる情報に対応するサービス・リポジトリ・エントリの作成に必要な情報が自動的に収集されます。このインタフェースは、やり取りされるデータによりコール間で異なる可能性があります。たとえば同じ Tuxedo サービスに対するコールでも、あるコンテキストでは ADDRESS フィールドが返され、他のコンテキストでは返されない場合があります。

これは、既存のアプリケーションを使用またはシミュレートすることで実行できます。この場合、次の入力で UD32 を使用します。たとえば、次のとおりです。

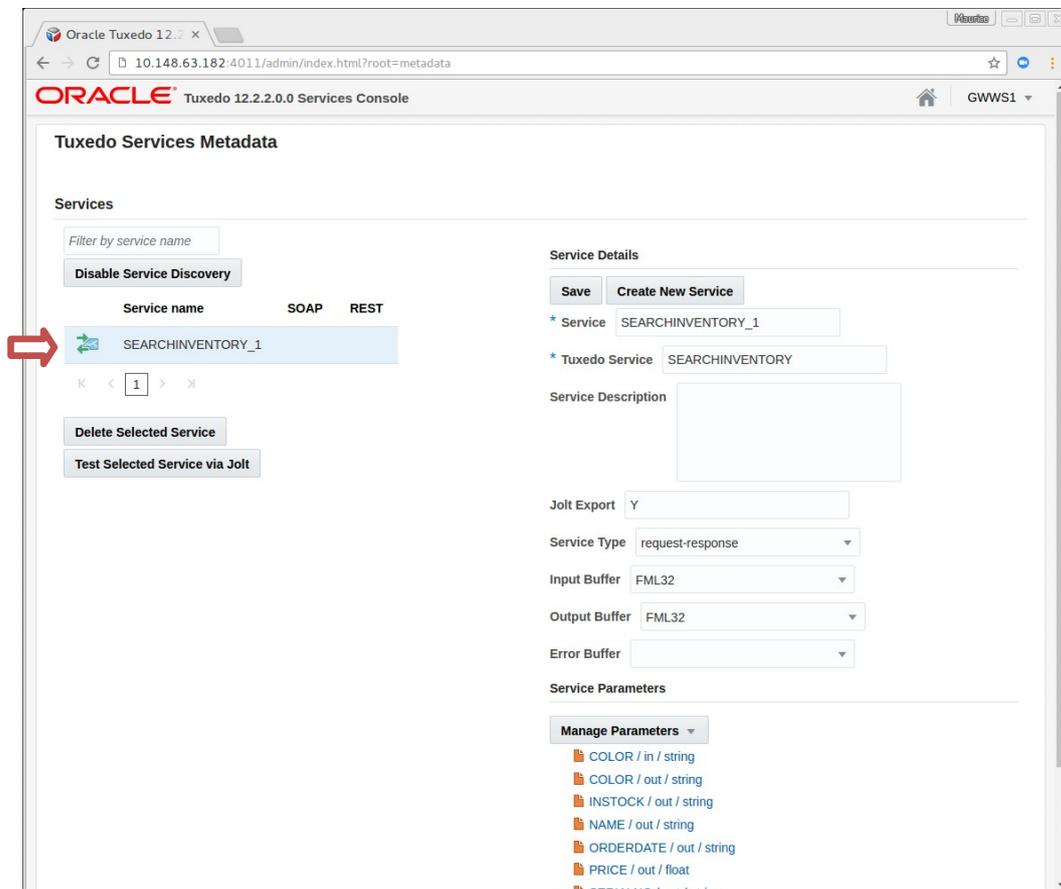
```
SRVCNM      SEARCHINVENTORY
COLOR ORANGE
```

このコンテンツを、たとえば uBike.ud32 などのファイルに保存します。

この入力で ud32 をコールすると、次の数行が表示されます。

```
$ ud32 < uBike.ud32
SENT pkt(1) is :
SRVCNM      SEARCHINVENTORY
COLOR ORANGE
RTN pkt(1) is :
SIZE  58
SIZE  21
SIZE  16
...
```

このコールの後にメタデータ・リポジトリ・ページを再ロードすると、新しいエントリが表示されます。



このエントリをクリックすると、サービスの詳細情報が表示されます。

## Webサービスの作成

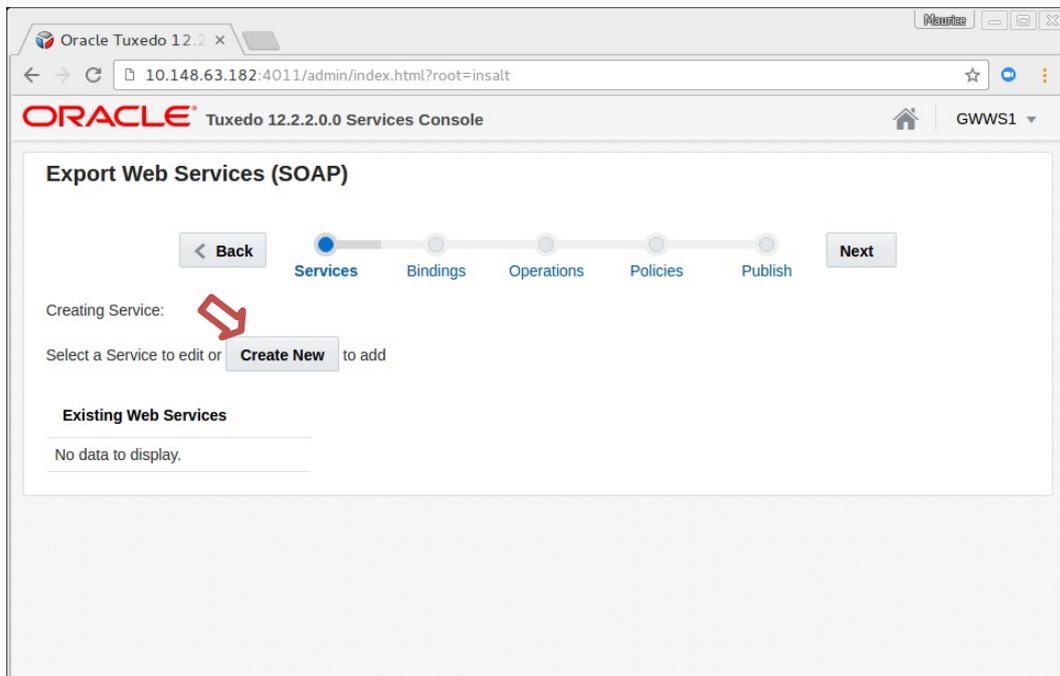
ホーム・アイコンをクリックしてホームページに戻ります。



次に「**Exported Web Services (SOAP)**」をクリックして、ネイティブの Web サービス管理ページに移動します。



「**Create New**」をクリックします。



サービス名を入力します。

Oracle Tuxedo 12.2.2.0.0 Services Console

### Export Web Services (SOAP)

Creating Service: uBikeSvc1

Services Bindings Operations Policies Publish

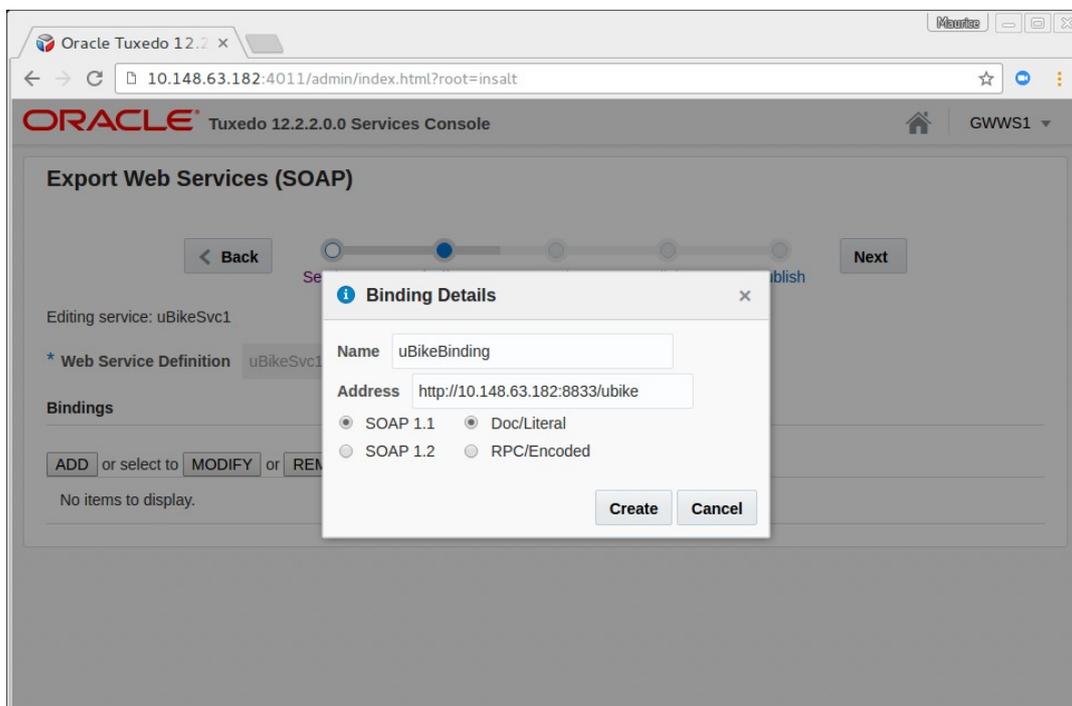
\* Web Service Definition

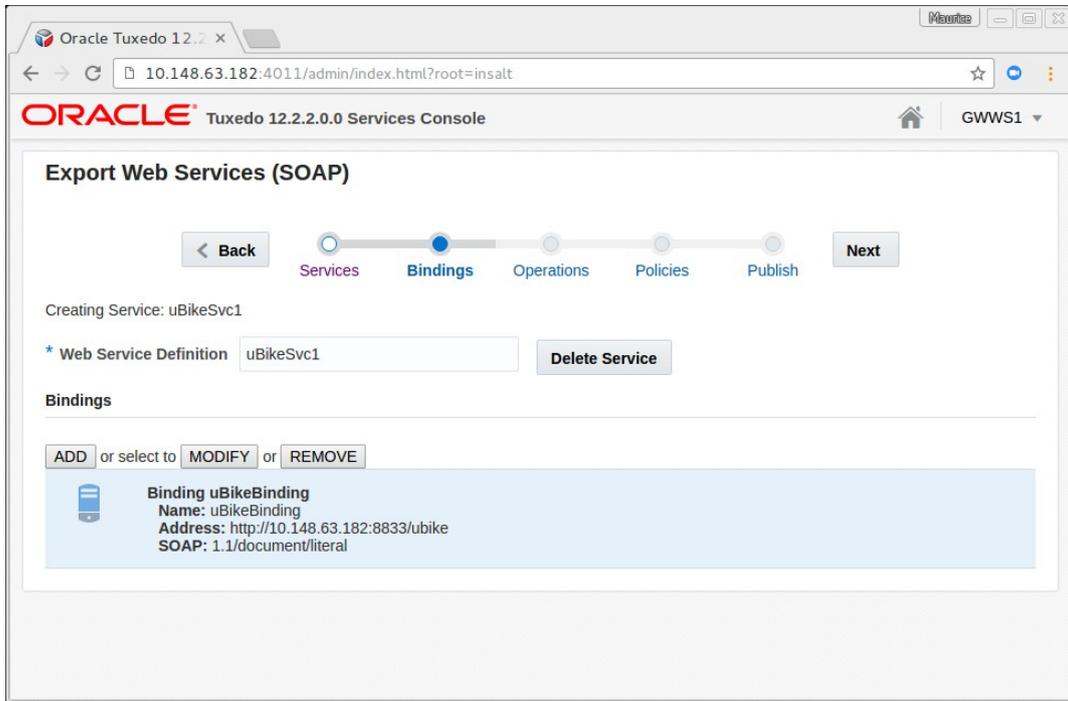
enter a web service definition name

2  or select to  or

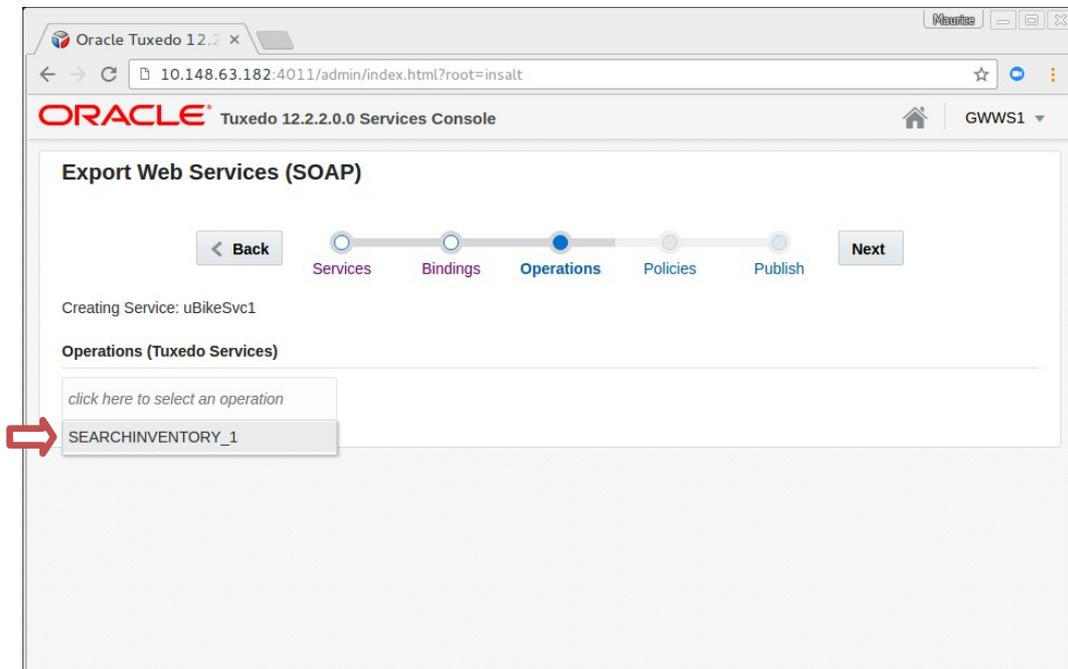
No items to display.

次にバインディングを追加します。バインディングを使用して、このサービスのエンドポイント（起動できるネットワーク・アドレス）と SOAP 特性（使用されるバージョンとペイロード・エンコーディング）を指定します。この例では、もっとも一般的なバインディングを示します。





「Next」をクリックしてから、新しく作成したメタデータ・サービスを、この Web サービスの操作として追加します。



Oracle Tuxedo 12.2 x

10.148.63.182:4011/admin/index.html?root=insalt

**ORACLE** Tuxedo 12.2.2.0.0 Services Console GWWS1

### Export Web Services (SOAP)

[Back](#)   **Services**   Bindings   **Operations**   Policies   Publish   [Next](#)

Creating Service: uBikeSvc1

**Operations (Tuxedo Services)**

SEARCHINVENTORY\_1 x

ポリシーは非常に一般的なユースケースではありません。スキップして、サービスのサマリーが表示されている公開に進みます。

Oracle Tuxedo 12.2.2.0.0 Services Console

### Export Web Services (SOAP)

[Back](#) Services Bindings Operations Policies **Publish** [Finish](#)

Creating Service: uBikeSvc1

Web Service Definition uBikeSvc1

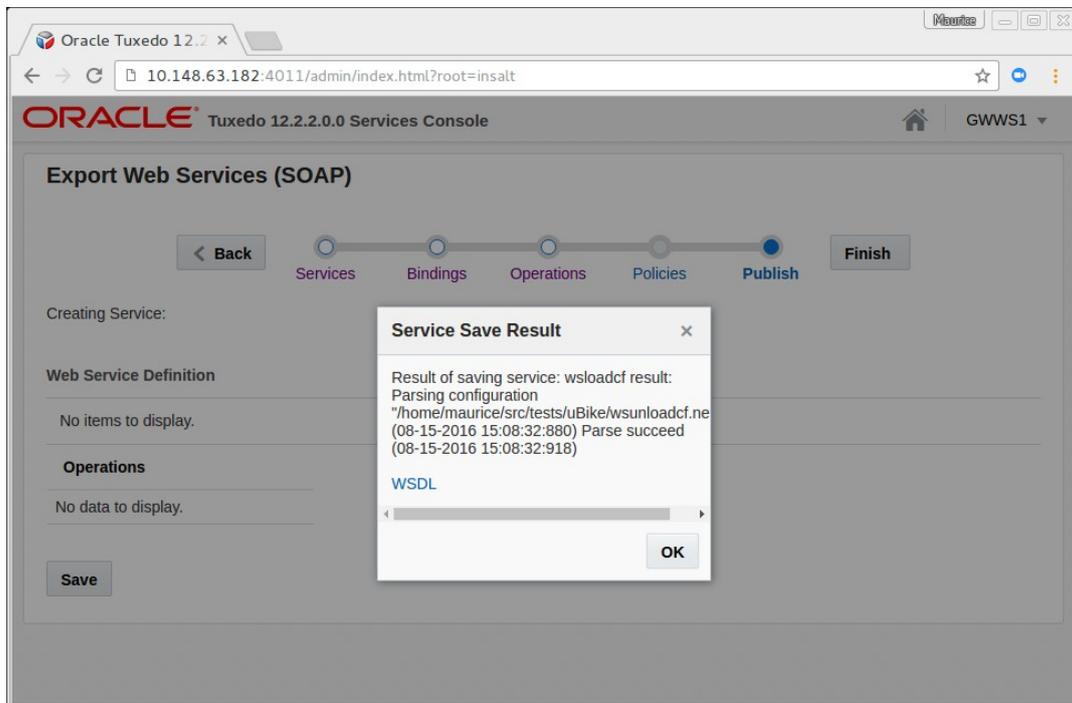
**Binding uBikeBinding**  
Name: uBikeBinding  
Address: http://10.148.63.182:8833/ubike  
SOAP: 1.1/document/literal

**Operations**

SEARCHINVENTORY\_1

[Save](#)

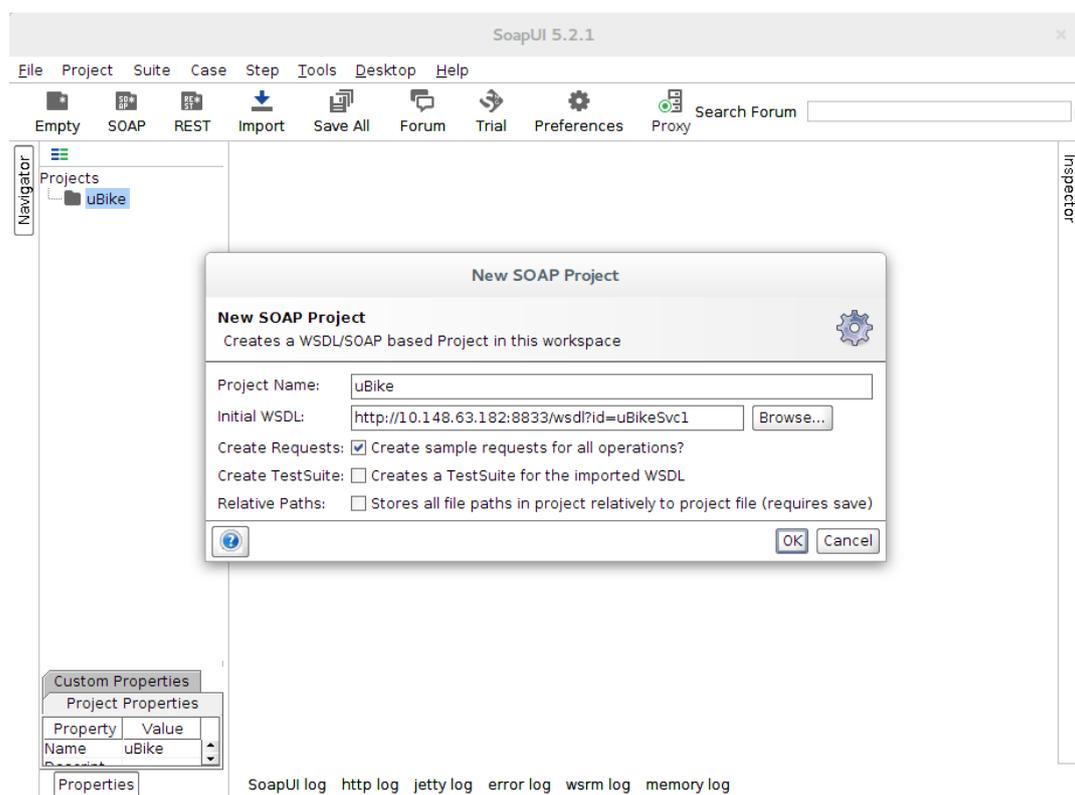
「**Save**」をクリックすると、サービスが作成され、起動できる状態になっています。



WSDL リンクをクリックして、サービスの WSDL を表示できます。次の手順で、WSDL を使用してサービスをテストできます。

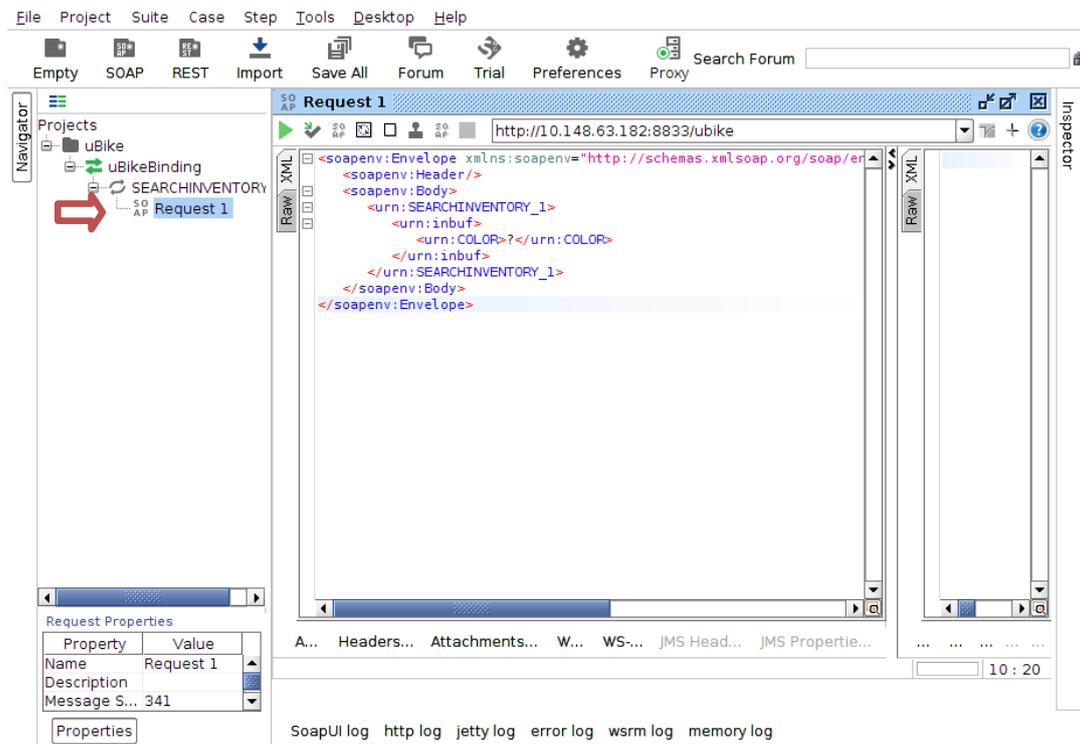
## SOAPUIを使用しWebサービスを起動して検証する

サービスが作成されたら、SOAPUI を起動して SOAP プロジェクトを作成します。**File** メニューで「**New SOAP Project...**」を選択します。



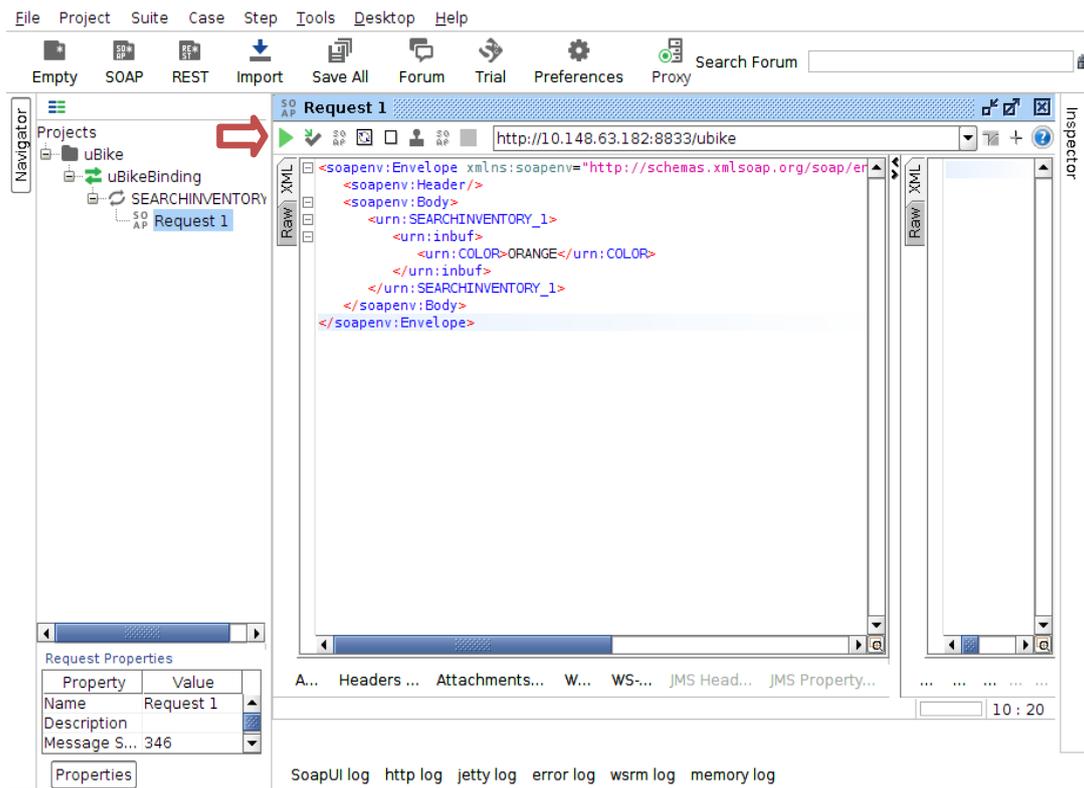
初期 WSDL は手動で入力することもできます。その場合はホスト、バインディング・エンドポイント用のポートの後に、/wsdl、およびこの Web サービス用に入力した名前が含まれる ID を入力します。スクリーンショットの例を参照してください。

すべてデフォルトのままにして「OK」をクリックし、操作名の下に「Request」をダブルクリックします。



次に、UD32 スクリプトの場合と同様に **COLOR** 要素に引数を入力できます。たとえば、**ORANGE** 色を入力します。

\



The screenshot shows the SoapUI interface with a SOAP request editor. The URL is `http://10.148.63.182:8833/ubike`. The XML body contains the following structure:

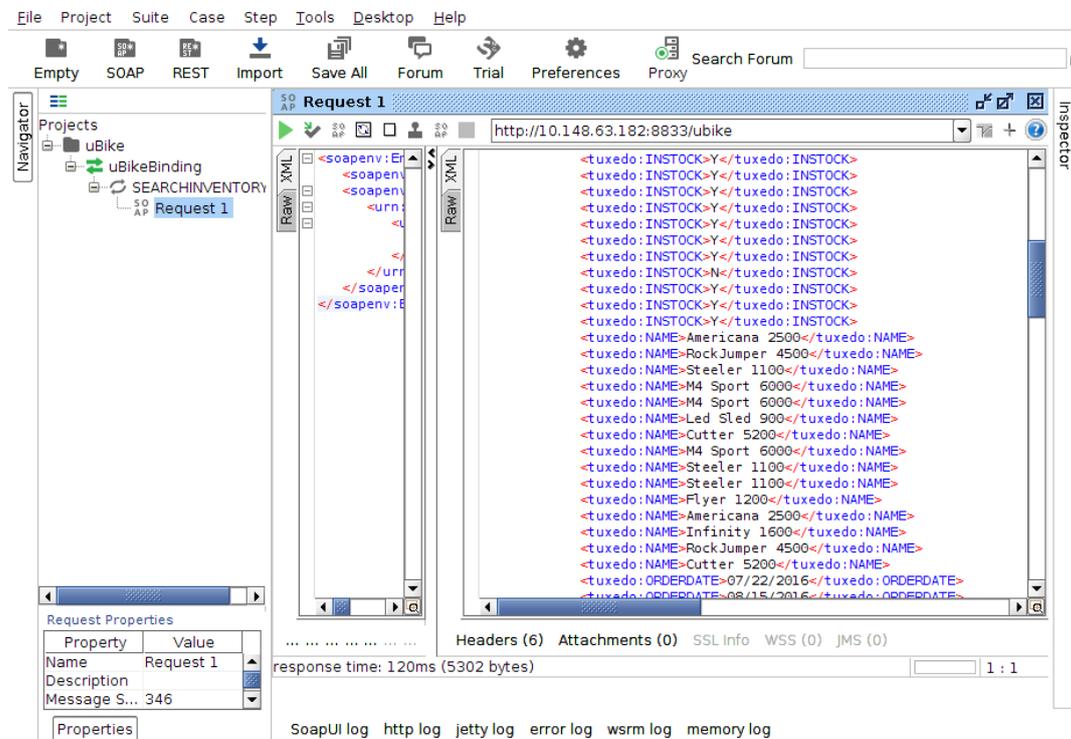
```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:SEARCHINVENTORY_1>
      <urn:inbuf>
        <urn:COLOR>ORANGE</urn:COLOR>
      </urn:inbuf>
    </urn:SEARCHINVENTORY_1>
  </soapenv:Body>
</soapenv:Envelope>
```

The `<urn:COLOR>ORANGE</urn:COLOR>` line is highlighted in red. A red arrow points to the 'Request 1' entry in the Navigator pane on the left. Below the XML editor, the 'Request Properties' table is visible:

Property	Value
Name	Request 1
Description	
Message S...	346

At the bottom of the interface, there are log options: SoapUI log, http log, jetty log, error log, wsrn log, memory log.

緑色の矢印をクリックすると、次のように表示されます。



## 結論

この手順書では、既存の Tuxedo サービスから SOAP Web サービスをほんの数分で作成する方法を説明しています。ここではプロセスを再起動せずに、どのようにしてすべての構成の詳細情報が非表示になり、すべてのアクションが動的になるかを示しています。

サービスの複雑さに関係なく、シンプルな STRING 型ベースのサービスや、VIEW または FML の複雑なリッチ・ペイロードなどをこの方法で公開できます。

一度設定をすれば、暗号化とセキュリティ（基本的なアクセス認証を使った SSL）、WS-\* 機能（WS-Security、WS-Addressing、WS-RM、WS-TX）などの機能をさらに追加できます。