jUDDI 3.0 - Developer Guide

Developer Guide

Contents

Table of Contents

About This Guide

What This Guide Contains

The Developer Guide document describes

Audience

This guide is most relevant to engineers who are responsible for setting up jUDDI 3.0 - Developer Guide installations.

Prerequisites

None.

Organization

This guide contains the following chapters:

- **Chapter 1, Dev Environment Setup**

- **Chapter 2, Testing**

- **Chapter 3,**

Documentation Conventions

The following conventions are used in this guide:

Convention	Description
Italic	In paragraph text, italic identifies the titles of documents that are being referenced. When used in conjunction with the Code text described below, italics identify a variable that should be replaced by the user with an actual value.
Bold	Emphasizes items of particular importance.
`Code`	Text that represents programming code.
Function | Function	A path to a function or dialog box within an interface. For example, "Select File | Open." indicates that you should select the Open function from the File menu.

( ) and |	Parentheses enclose optional items in command syntax. The vertical bar separates syntax items in a list of choices. For example, any of the following three items can be entered in this syntax:
	
	 `persistPolicy (Never | OnTimer | OnUpdate |`
	 `NoMoreOftenThan)`

Note: **Caution**:	A note highlights important supplemental information. A caution highlights procedures or information that is necessary to avoid damage to equipment, damage to software, loss of data, or invalid test results.

Table 1 Formatting Conventions

Additional Documentation

None on the subject.

Contacting Us

Email: juddi-dev@ws.apache.org

Dev Environment Setup

Introduction

Within jUDDI source, there are the following modules:

uddi-ws - JAXWS stubs built from the wsdls
juddi-core - the jUDDI jar containing the model, api and core jUDDI functionality
juddi-cxf - a WAR module that uses CXF as the web service framework, chosen by default
juddi-axis - a WAR module that uses Axis 2 as the web service framework, this is an alternate to using CXF
juddi-tomcat - a module which builds a Tomcat bundle with juddi-cxf installed and derby as a backend database
juddi-cargo - a module to help test jUDDI web services (currently no tests included)

jUDDI v3 is set up to produce a number of different deliverables – a JAR, a WAR, and a tomcat bundle. Depending on the scope of your application, or your interest in the project, you might want to use the Tomcat server bundle packaged with the Derby database and jUDDI, or you may just want to use the jUDDI JAR and make your own database and Web Service choices.   jUDDI is set up so that it can support a range of environments.

Building the project

First, check out the jUDDI sources:

```
% svn co http://svn.apache.org/repos/asf/webservices/juddi/branches/v3_trunk
```

Then build the entire project:

```
% cd v3_trunk
% mvn clean install
```

Next we will cover building the separate components.

Building the JAR

The juddi-core module produces a JAR which contains the jUDDI source and a jUDDI persistence.xml configuration.   jUDDI is currently setup so that you can choose between using either OpenJPA or Hibernate as your persistence framework.   The juddi-core pom.xml contains two profiles, triggered on the "persistence" property.

OpenJPA

```
% cd juddi-core
% mvn -Dpersistence=openjpa
```

Hibernate

```
% cd juddi-core
% mvn -Dpersistence=hibernate
```

Currently the parent POM uses hibernate as the default persistence layer, but this can be easily changed by modifying the persistence property that is specified in the parent pom.xml.

By default, the project is currently using hibernate as the persistence layer.   In the v3trunk, the pom.xml sets the persistence property :

v3trunk pom.xml

```
...cut...
<properties>
    <persistence>hibernate</persistence>
</properties>
...cut...
```

Building the WAR

As with the JAR, you need to make a decision on what framework you would like to use when building the WAR. The juddi-cxf module builds a war that uses CXF as a web service framework, and the juddi-axis target builds a war that uses Axis 2 as a web service framework.

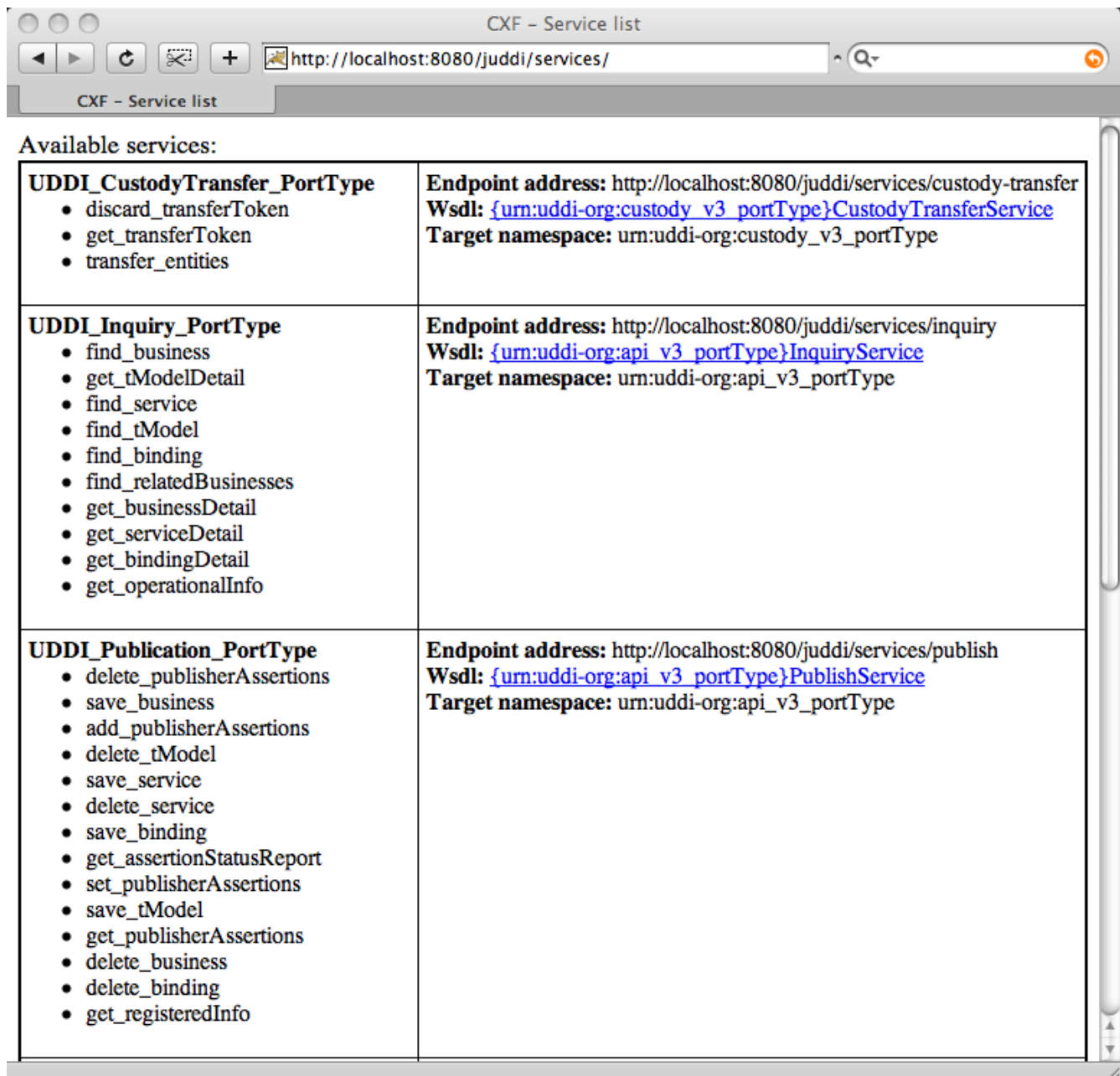There are currently some issues with the Axis 2 web service framework in juddi-axis.

Building the Tomcat Bundle

The jUDDI Tomcat bundle packages up one of the jUDDI WAR files, Apache Derby, and a few necessary configuration files and provides the user with a pre-configured jUDDI instance.   By default, the WAR produced by the juddi-cxf module is used – the example below shown uses URLs and endpoints using the jUDDI CXF configuration.   If you use the Axis 2 configuration, URLs and endpoints may differ.

To get started using the Tomcat bundle, unzip the juddi-tomcat-bundle.zip, and start Tomcat :

```
% cd apache-tomcat-5.5.23/bin
% ./startup.sh
```

Browse to http://localhost:8080/juddi/services

The services page shows you the available endpoints and methods available.   Using any SOAP client, you should be able to send some sample requests to jUDDI to test:

Using jUDDI with your application

As of the Alpha release, two of the UDDI v3 APIs should be active within jUDDI : inquiry and publish.

Index