7.4 RSS

RSS Data Model

RSS (Really Simple Syndication) is an XML-based document format for the syndication of web content such as weblogs and news headlines to Web sites as well as directly to user agents. Apache Wink supports the RSS 2.0 specification.

RSS Data Model Overview

Apache Wink provides an RSS data model for consuming and producing RSS Feeds (application/xml). All of the model classes are located under **org.apache.wink.common.model.rss** package.

RSS Feed Support

The following table shows the RSS Feed data models and the representations in which it can be serialized and de-serialized.

	Supported	Media Types	Data Model	Provider registration
Read	Yes	application/xml	org.apache.wink.common.model.rss.RssFeed	Not required. Registered by default
Write	Yes	application/xml	org.apache.wink.common.model.rss.RssFeed	Not required. Registered by default

Examples

The following code example demonstrates reading and writing of RSS Feeds.

Producing RSS Feed

The following code example demonstrates the creation of an RSS Feed.

```
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public RssFeed getFeed() {
        RssFeed rss = new RssFeed();

        RssChannel channel = new RssChannel();
        channel.setTitle("Liftoff News");
        channel.setLink("http://liftoff.msfc.nasa.gov");
        channel.setDescription("Liftoff to Space Exploration.");
        channel.setPubDate(new Date().toString());

        RssItem item = new RssItem();
        item.setTitle("Star City");
        item.setLink("http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp");
        item.setDescription("How do Americans get ready to work with Russians aboard the International Space
Station?");
        channel.getItems().add(item);

        ...

        rss.setChannel(channel);
        return rss;
    }
```

Consuming RSS Feed

The following code example demonstrates the consumption of an RSS Feed.

```
@POST
@Consumes(MediaType.APPLICATION_XML)
public void setFeed(RssFeed feed) {
    ...

    return;
}
```

RSS Mapped into Syndication Object Model

Starting with v1.1 of Apache Wink, the RSS object model has been mapped into Apache Wink's Syndication Object Model. This enables common Synd APIs (available in **org.apache.wink.common.model.synd** package) to be used for consuming/producing both RSS and Atom documents.

Mapping between RSS and Atom:

The following table illustrates the mapping between RSS and Syndication Object Model (the same as the Atom model):

RSS document	Mapped into Atom
<?xml version="1.0"?> <rss version="2.0"> <channel>	<?xml version="1.0" encoding="utf-8"?> <feed xmlns="http://www.w3.org/2005/Atom" xml:lang="en-us">
<title>Liftoff News</title>	<title type="text">Liftoff News</title>
<link>http://liftoff.msfc.nasa.gov/</link>	<link rel="alternate" href="http://liftoff.msfc.nasa.gov/" />
<description>Liftoff to Space Exploration.</description>	<subtitle type="text">Liftoff to Space Exploration.</subtitle>
<language>en-us</language>	//see the attribute "xml:lang" of "feed" element
<copyright>Copyright 2002, Spartanburg Herald-Journal</copyright>	<rights>Copyright 2002, Spartanburg Herald-Journal</rights>
<managingEditor>editor@example.com</managingEditor>	<author> <name>editor</name> <email>editor@example.com</email> </author>
<lastBuildDate>Tue, 10 Jun 2003 09:41:01 GMT</lastBuildDate>	<updated>2003-06-10T09:41:01Z</updated>
<category domain="http://www.fool.com/cusips">MSFT</category> <category domain="http://www.fool.com/musips">MOTO</category>	<category scheme="http://www.fool.com/cusips" label="MSFT"/> <category scheme="http://www.fool.com/musips" label="MOTO"/>
<generator>Weblog Editor 2.0</generator>	<generator>Weblog Editor 2.0</generator>
<image> <url>http://liftoff.msfc.nasa.gov/news.gif</url> <title>Litoff News</title> <link>http://liftoff.msfc.nasa.gov/</link> <width>100</width> <height>100<height> <description>News</description> </image>	<logo>http://liftoff.msfc.nasa.gov/news.gif</logo>
<item>...</item>	<entry>...</entry>
<item>...</item>	<entry>...</entry>
<item>	<entry>
<title>Star City</title>	<title type="text">Star City</title>
<link>http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp</link>	<link rel="alternate" href="http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp"/>
<description>How do Americans get ready to work with Russians aboard the International Space Station? They take a crash course in culture, language and protocol at Russia's Star City.</description>	<summary type="text">How do Americans get ready to work with Russians aboard the International Space Station? They take a crash course in culture, language and protocol at Russia's Star City.</summary>
<author>author1@rssboard.org</author>	<author> <name>author1</name> <email>author1@rssboard.org</email> </author>
<category domain="http://www.fool.com/cusips">MSFT</category> <category domain="http://www.fool.com/musips">MOTO</category>	<category scheme="http://www.fool.com/cusips" label="MSFT" /> <category scheme="http://www.fool.com/musips" label="MOTO" />
<enclosure url="http://www.scripting.com/mp3s/weatherReportSuite.mp3" length="12216320" type="audio/mpeg" />	<link rel="enclosure" type="audio/mpeg" length="12216320" href="http://www.scripting.com/mp3s/weatherReportSuite.mp3"/>

<guid>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</guid>	<id>http://liftoff.msfc.nasa.gov/2003/06/03.html#item573</id>
<pubDate>Tue, 03 Jun 2003 09:39:21 GMT</pubDate>	<published>2003-06-03T09:39:21Z</published>
</item>	</entry>
</channel> </rss>	</feed>

Examples

The following code examples demonstrate how to convert an RssFeed object into a SyndFeed object and vice versa.

Converting RssFeed object into SyndFeed object:

The following code example demonstrates the conversion of an RssFeed object into a SyndFeed object:

```
    @Consumes(MediaType.APPLICATION_XML)
    public void readFeed(RssFeed rssFeed) {
        // Map RSS into SyndFeed
        SyndFeed syndFeed = new SyndFeed();
        syndFeed = rssFeed.toSynd(syndFeed);

        // Now access RSS using SyndFeed APIs
        if (syndFeed.getTitle() != null) {
            System.out.println("Title = " + syndFeed.getTitle().getValue());
        }
        if (syndFeed.getSubtitle() != null) {
            System.out.println("Descritpion = " + syndFeed.getSubtitle().getValue());
        }
        ...

        return;
    }
```

Converting SyndFeed object into RssFeed object:

The following code example demonstrates the conversion of a SyndFeed object into an RssFeed object:

```
    @GET
    @Produces(MediaType.APPLICATION_XML)
    public RssFeed getFeed() {
        SyndFeed syndFeed = new SyndFeed();
        syndFeed.setTitle(new SyndText("Liftoff News"));
        SyndLink syndLink = new SyndLink();
        syndLink.setRel("alternate");
        syndLink.setHref("http://liftoff.msfc.nasa.gov");
        syndFeed.addLink(syndLink);
        syndFeed.setSubtitle(new SyndText("Liftoff to Space Exploration."));

        SyndEntry syndEntry = new SyndEntry();
        syndEntry.setTitle(new SyndText("Star City"));
        SyndLink syndLink2 = new SyndLink();
        syndLink2.setRel("alternate");
        syndLink2.setHref("http://liftoff.msfc.nasa.gov/news/2003/news-starcity.asp");
        syndEntry.addLink(syndLink2);
        syndEntry.setSummary(new SyndText("How do Americans get ready to work with Russians aboard the
International Space Station?"));
        syndFeed.addEntry(syndEntry);

        ...

        //Map SyndFeed into RssFeed
        RssFeed rssFeed = new RssFeed(syndFeed);
        return rssFeed;
    }
```

For a further example, please see **ReadRssAsSyndFeed.java** under **examples\client\ReadRSS-client**.