

# Creating and Consuming Atom Documents

## Creating and Consuming Atom Documents

The two most common uses for Abdera are creating and parsing Atom documents.

Minimal Required Maven dependency for both examples:

```
<dependency>
  <groupId>org.apache.abdera</groupId>
  <artifactId>abdera-parser</artifactId>
  <version>1.1.1</version>
</dependency>
```

## Parsing an Atom Document and printing entry titles

Example Code:

```
Abdera abdera = new Abdera();
Parser parser = abdera.getParser();

URL url = new URL("http://intertwingly.net/blog/index.atom");
Document<Feed> doc = parser.parse(url.openStream(),url.toString());
Feed feed = doc.getRoot();
System.out.println(feed.getTitle());
for (Entry entry : feed.getEntries()) {
    System.out.println("\t" + entry.getTitle());
}
System.out.println (feed.getAuthor());
```

## Creating an Atom Document and adding an entry

Example Code:

```
Abdera abdera = new Abdera();
Feed feed = abdera.newFeed();

feed.setId("tag:example.org,2007:/foo");
feed.setTitle("Test Feed");
feed.setSubtitle("Feed subtitle");
feed.setUpdated(new Date());
feed.addAuthor("James Snell");
feed.addLink("http://example.com");
feed.addLink("http://example.com/foo","self");

Entry entry = feed.addEntry();
entry.setId("tag:example.org,2007:/foo/entries/1");
entry.setTitle("Entry title");
entry.setSummaryAsHtml("<p>This is the entry title</p>");
entry.setUpdated(new Date());
entry.setPublished(new Date());
entry.addLink("http://example.com/foo/entries/1");
```

## Instantiating Abdera

As you can see in both of these examples, the first step to using Abdera is to create an instance of the Abdera object. This special thread-safe class bootstraps the Abdera configuration and provides access to all of the other components. One of the most important tasks that the Abdera object performs is the automatic discovery of extensions.

Creating a new instance of the Abdera object can be time consuming so it is recommended that only a single instance be creating per application. The Abdera object and it's direct children (Parser, Factory, XPath, etc) are threadsafe and can be stored statically.

#### Creating a static instance of Abdera

```
public class MyApplication {
    private static Abdera abdera = null;

    public static synchronized Abdera getInstance() {
        if (abdera == null) abdera = new Abdera();
        return abdera;
    }
}
```

Once an Abdera instance is created, you can begin parsing and creating Atom documents.