Getting Started

- Creating and Consuming Atom Documents
- AtomPub
- Overview of Abdera Modules
- The Classpath
  - Maven
  - Manual Classpath Setup
- References

Apache Abdera provides an implementation of the IETF Atom Syndication Format and Atom Publishing Protocol standards (RFC's 4287 and 5023).

Creating and Consuming Atom Documents

AtomPub

Your first AtomPub Server

AtomPub Client

Overview of Abdera Modules

Abdera is composed of a number of individual modules:

- Core: Provides the core interfaces used throughout Abdera:
  - Feed Object Model - A set of Java interfaces and abstract classes modeled after the various document and element types defined by the Atom specifications.
  - Parser and Factory interfaces - Interfaces that provide the mechanisms for creating and consuming Atom documents.
  - ExtensionFactory - The primary means by which extensions to the Atom format are integrated into Abdera. Extension Factories are used by both the Parser and Factory.
  - Writer - The means by which Feed Object Model objects are serialized into XML or other formats (Custom Writers, JSON Serialization)
  - ParseFilter - Provides a means of filtering the stream of parse events
  - Converter - Base interfaces and utilities for converting objects into Feed Object Model objects
  - XPath - Base interface for navigating the Feed Object Model using XPath
- Dependencies: Contains all of the other jars and code Abdera depends on
- Parser: Provide an implementation of the Feed Object Model based on the Apache Axiom project
- Protocol: Provides the base interfaces and utility classes used for the RFC 5023 (Atompub) implementation
- Server: Provides framework code used to build Atompub servers
- Client: Provides an Atompub client implementation that is based on the Apache Commons HTTP Client.
- Spring: Extends the Server module by adding support for the Spring framework
- Security: Provides support for XML Digital Signatures and XML Encryption
- Extensions: Provides support for a number of standard and non-standard extensions to the Atom format
  - Atom Threading Extensions
  - Atom License Extension
  - Atompub Feature Discovery
  - Atom Bidi Attribute
  - Feed Paging and Archiving
  - GeoRSS
  - Simple Sharing Extensions
  - MediaRSS
  - OpenSearch
  - GData and WSSE Authentication
- i18n: URI Templates
- Examples: Provides a number of examples

The Classpath

Maven

Sample for adding a single Abdera module:
Adding Abdera with all existing submodules:

```xml
<dependencies>
  <dependency>
    <groupId>org.apache.abdera</groupId>
    <artifactId>abdera-client</artifactId>
    <version>1.1.1</version>
  </dependency>
</dependencies>
```

If you would like to use Abdera SNAPSHOTS you can add the Apache SNAPSHOT repository:

```xml
<repositories>
  <repository>
    <id>apache-snapshots</id>
    <name>Apache Snapshot Repository</name>
    <url>http://repository.apache.org/snapshots/</url>
    <snapshots>
      <enabled>true</enabled>
    </snapshots>
  </repository>
</repositories>
```

---

**Manual Classpath Setup**

Abdera uses a modular architecture that divides functionality up into a number of individual jars. Different applications will require different classpath configurations depending on the Abdera features they are utilizing.

- **Abdera Jars**
  - Core API: abdera.core.0.3.0-incubating.jar (required)
  - Parser Impl: abdera.parser.0.3.0-incubating.jar (required)
  - Security Impl: abdera.security.0.3.0-incubating.jar (optional)
  - Protocol Core API: abdera.protocol.0.3.0-incubating.jar (optional)
  - Protocol Client Impl: abdera.client.0.3.0-incubating.jar (optional, requires protocol)
  - Protocol Server Impl: abdera.server.0.3.0-incubating.jar (optional, requires protocol)
  - Spring Integration: abdera.spring.0.3.0-incubating.jar (optional, requires server)
  - IRI Support: abdera.iri.0.3.0-incubating.jar (required)

- **Abdera Extension Jars (optional)**
  - Main (thread, license, paging, etc): abdera.extensions.main.0.3.0-incubating.jar (requires client)
  - GoogleLogin Auth: abdera.extensions.gdata.0.3.0-incubating.jar (requires client)
  - GeoRSS: abdera.extensions.geo.0.3.0-incubating.jar
  - JSON Writer: abdera.extensions.json.0.3.0-incubating.jar
  - MediaRSS: abdera.extensions.media.0.3.0-incubating.jar
  - OpenSearch: abdera.extensions.opensearch.0.3.0-incubating.jar
  - Simple Sharing Extensions: abdera.extensions.sharing.0.3.0-incubating.jar
  - WSSE Auth: abdera.extensions.wss.0.3.0-incubating.jar (requires client)

- **Dependencies**
  - Axiom (required by parser)
    - axiom-api-1.2.5.jar
    - axiom-impl-1.2.5.jar
  - Apache Commons
    - Codec: commons-codec-1.3.jar (required by protocol)
    - HttpClient: commons-httpclient-3.1-rc1.jar (required by client)
Logging: commons-logging-1.0.4.jar (required)

Geronimo
- geronimo-activation_1.0.2_spec-1.1.jar (any JAF impl is required)
- geronimo-servlet_2.4_spec-1.0.jar (required by server, security and spring modules)

Jetty (required only by compile time for junit tests)
- jetty-6.1.3.jar
- jetty-util-6.1.3.jar

Jaxen: jaxen-1.1.1.jar (required by the core)

JSON: json-1.0.jar (required by json extension)

JUnit: junit-4.3.jar (required at compile time)

log4j-1.2.24.jar (required)

Retroweaver: retroweaver-rt-2.0.jar (required for JDK 1.4 support)

Stax (any Stax API impl is required at runtime)
- stax-api-1.0.1.jar
- wstx-asl-3.2.1.jar

XML Security: xmlsec-1.4.1.jar (required by security module)

References
- The Atom Syndication Format
- The Atom Publishing Protocol
- Atom Threading Extensions
- Atom License Extension
- Feed Paging and Archiving
- IETF Atom Workgroup Wiki