

How To Generate CloudStack API Documentation

How we make API references.

- [Overview](#)
- [Annotations](#)
 - [Java to HTML Conversion Code](#)
 - [Cosmetic Touch-Ups](#)
 - [XML to HTML using Jenkins](#)
- [Building API Documentation with Maven](#)
- [Release Notes](#)
- [Troubleshooting notes](#)

Overview

How do we create API documentation for CloudStack?

1. Java Annotations in API code.
2. Cosmetic touch-ups in doc generation code.
3. Java to HTML using Jenkins.
4. Troubleshooting notes

Annotations

Engineers include descriptive annotations when they write code. In Java, annotations start with @. Some of the annotations we have defined to contain documentation strings look like this:

```
@Implementation(description="Creates an account", responseObject=UserResponse.class)
public class CreateAccountCmd extends BaseCmd {
public static final Logger s_logger = Logger.getLogger(CreateAccountCmd.class.getName());
private static final String s_name = "createaccountresponse";
////////////////////////////////////
//////////////////////////////////// API parameters //////////////////////////////////////
////////////////////////////////////
@Parameter(name=ApiConstants.ACCOUNT, type=CommandType.STRING,
description="Creates the user under the specified account.")
private String accountName;
```

See the description=" " inside @Implementation and @Parameter. That text will be extracted in the Java to HTML step.

For more information, see [Annotations in the API](#).

Java to HTML Conversion Code

There is one piece of code that converts the Java annotations to XML output. Then there is another piece of code that takes in the XML for the table of contents as well as each command, and generates the HTML documents which we upload to our website.

The API doc building script is located at:

```
setup/apidoc/build-apidoc.sh
```

It calls the following to create intermediate XML output:

```
server/src/com/cloud/api/doc/ApiXmlDocWriter.java
```

After creating the XML files with ApiXmlDocWriter, the script calls the following:

```
setup/apidoc/XmlToHtmlConverter.java
```

This XmlToHtmlConverter uses a series of .xsl transforms (aka style sheets) to convert the XML to HTML. Those .xsl's are all stored in:

```
setup/apidoc/*.xsl
```

Cosmetic Touch-Ups

1. Check out the code. Be in the right branch.
2. In generatetocforuser.xsl, generatetocfordomainadmin.xsl, generatetocforadmin.xsl, generateadmincommands.xsl, generateusercommands.xsl, and generatedomainadmincommands.xsl, update the software version and any other boilerplate text.
3. Check in your changes to the right branch.

XML to HTML using Jenkins

1. Within a few minutes of checking in your cosmetic changes, you can get the new output. Go to jenkins.cloudstack.org and click the name of the build, depending on which branch you want. Under 'Last Successful Artifacts' you'll see the zip file with everything in it.
2. Check the Build History, and make sure that the most recent build succeeded. If it didn't, you will want to find out why and fix any important problems.
3. Assuming build success, grab that zip file, and unzip it on your machine. Your final unzipped output will be the ~/html folder, which will contain the dynamically generated content, and the static hardcoded files.
4. Upload the HTML to the docs website (procedure: TODO. David Nalley).

Building API Documentation with Maven

The API documentation can be built with Maven using the following commands (should work on 4.1 and above). The belt-and-braces approach which always work is to build everything:

1. `mvn -Pdeveloper -Dnoredist clean install`

However if you've just edited the .xsl files since building you can use:

1. `mvn -Pdeveloper -Dnoredist clean install -pl :cloud-apidoc`

The built documentation can then be found here: `tools/apidoc/target/xmldoc/html`

Release Notes

The `ApiXmlDocReader` class will read and generate a text file of the difference between two versions of the API XML documents of Apache CloudStack. It needs access to the `Xstream` and the `ApiXmlDocReader` class.

You can call it with the following command:

```
java -cp ~/.m2/repository/com/thoughtworks/xstream/xstream/1.4.10/xstream-1.4.10.jar:/path/to/source/cloudstack/dist/rpmbuild/BUILD/cloudstack-4.12.0-SNAPSHOT/server/target/classes \
com.cloud.api.doc.ApiXmlDocReader -old /path/to/4.11.commands.xml -new /path/to/4.12.commands.xml -d /path/to/diff/
```

You'll find `diff.txt` in `/path/to/diff/diff.txt` which will contain the changes between the two files.

Troubleshooting notes

1. You may have to update `"tools/apidoc/gen_toc.py"` and update the data structure: `"known_categories"` in case you add a command that it doesn't know how to categorize.
2. You cannot have empty descriptions in your annotations (like `description=""`). The XML parser python code will get mad and crash, forcing you to have to run build with `-e` and then go figure out what broke. [Thanks for doing -e and updating this wiki with this tip. !!](#)