



TUESDAY, 6 MARCH 2012

Programmatically Accessing Adobe CQ Content using the JCR API

You can programmatically modify nodes and properties located within the Adobe CQ 5.5 repository, which is part of the Adobe Digital Marketing Suite. To access the CQ repository, you use the Java Content Repository (JCR) API. You can use the Java JCR API to perform create, replace, update, and delete (CRUD) operations on content located within the DAY CQ repository. For more information about the Java JCR API, see <http://jackrabbit.apache.org/jcr-api.html>.

This development article creates a Java class that modifies nodes and properties within Adobe CQ. The Java class connects to a local instance of Adobe CQ and creates nodes and stores data values to node properties. You can store data by manipulating node properties.

Creating a Repository instance

Although there are different ways to connect to a repository and establish a connection, this development article uses a static method that belongs to the `org.apache.jackrabbit.commons.JcrUtils` class. The name of the method is `getRepository`. This method takes a string parameter that represents the URL of the Adobe CQ server. For example `http://localhost:4503/crx/server`.

The `getRepository` method returns a `Repository` instance, as shown in the following code example.

```
//Create a connection to the Day CQ repository running on local
host
Repository repository =
JcrUtils.getRepository("http://localhost:4503/crx/server");
```

Creating a Session instance

The `Repository` instance represents the CRX repository. You use the `Repository` instance to establish a session with the repository. To create a session, invoke the `Repository` instance's `login` method and pass a `javax.jcr.SimpleCredentials` object. The `login` method returns a `javax.jcr.Session` instance.

You create a `SimpleCredentials` object by using its constructor and passing the following string values:

- The user name
- The corresponding password

When passing the second parameter, call the `String` object's `toCharArray` method. The following code shows how to call the `login` method that returns a `javax.jcr.Session` instance.

```
//Create a Session instance
javax.jcr.Session session = repository.login( new
SimpleCredentials("admin", "admin".toCharArray()));
```

Creating a Node instance

FOLLOWERS

Join this site
with
Google
Friend
Connect

BLOG ARCHIVE

- ▼ 2012 (7)
 - May (3)
 - April (2)
 - ▼ March (1)
 - Programmatically Accessing Adobe CQ Content using ...
 - February (1)
- 2011 (4)

You can use a `Session` instance to create a `javax.jcr.Node` instance. A `Node` instance enables you to perform node operations. For example, you can create a new node. To create a node that represents the root node, invoke the `Session` instance's `getRootNode` method, as shown in the following line of code.

```
//Create a Node
Node root = session.getRootNode();
```

Once you create a `Node` instance, you can perform tasks such as creating another node and adding a value to it. For example, the following code creates two nodes and adds a value to the second node.

```
// Store content
Node adobe = root.addNode("adobe");
Node day = adobe.addNode("day");
day.setProperty("message", "Adobe CQ is part of the Adobe
Digital Marketing Suite!");
```

As shown in the previous code example, to store a data value to a node, invoke the `Node` object's `setProperty` method. As the first parameter, specify a property name. In this example, notice that `message` is specified. This value indicates that you want to set the message property. The second parameter is a string value that represents a data value to store.

Retrieving Node Values

To retrieve a node and its value, invoke the `Node` instance's `getNode` method and pass a string value that represents the fully-qualified path to the node. Consider the node structure created in the previous code example. To retrieve the `day` node, specify `adobe/day`, as shown in the following code:

```
// Retrieve content
Node node = root.getNode("adobe/day");
System.out.println(node.getPath());
System.out.println(node.getProperty("message").getString());
```

Adding the JCR JAR file

To use the JCR API, add the *jackrabbit-standalone-2.4.0.jar* file to your Java application's class path. You can obtain this JAR file from the Java JCR API web page at <http://jackrabbit.apache.org/jcr-api.html>.

Quick Start: Creating nodes in the DAY CQ Repository

The following Java code example represents a Java class that connects to the Adobe CQ server, creates a `Session` instance and adds new nodes. A node is assigned a data value and then the value of the node and its path is written out to the console.

```
/*
 * This Java Quick Start uses the jackrabbit-standalone-
 2.4.0.jar
 * file. See the previous section for the location of this JAR
 file
 */

import javax.jcr.Repository;
import javax.jcr.Session;
import javax.jcr.SimpleCredentials;
import javax.jcr.Node;

import org.apache.jackrabbit.commons.JcrUtils;
import org.apache.jackrabbit.core.TransientRepository;
```

```

public class GetRepository {

    public static void main(String[] args) throws Exception {

        //Create a connection to the Day CQ repository running on local
        host
        Repository repository =
        JcrUtils.getRepository("http://localhost:4503/crx/server");

        //Create a Session
        javax.jcr.Session session = repository.login( new
        SimpleCredentials("admin", "admin".toCharArray()));

        try {

            //Create a node that represents the root node
            Node root = session.getRootNode();

            // Store content
            Node adobe = root.addNode("adobe");
            Node day = adobe.addNode("day");
            day.setProperty("message", "DAY CQ is part of the Adobe
            Digital Marketing Suite!");

            // Retrieve content
            Node node = root.getNode("adobe/day");
            System.out.println(node.getPath());
            System.out.println(node.getProperty("message").getString());

            // Save the session changes and log out
            session.save();
            session.logout();
        }
        catch(Exception e){
            e.printStackTrace();
        }
    }
}

```

View the new nodes in CRXDE Lite

After you run the full code example and create the nodes, you can view the new nodes in the CRXDE Lite, as shown in the following illustration.



About the Author

I (Scott Macdonald) am a senior content and community lead at Adobe Systems with over 15 years in the high tech industry. I am also a programmer with knowledge in Java, JavaScript, C#, C++, HTML, XML and ActionScript. If you would like to see more CQ or other Adobe Digital Marketing end to end walkthroughs like this, then leave a comment and let me know what content you would like to see.

Posted by Scott Macdonald at 19:16

+1
Recommend

6 comments:



Anonymous 5 May 2012 03:09

Hi Scott,

This is a great article!

I needed to access the JCR repository from within an OSGI-Bundle.
Please suggest what would be the way to get the repository instance.

The below step:

```
Repository repository = JcrUtils.getRepository("http://localhost:4503/crx/server");  
seems to be to get the instance for an external application.
```

Also my bundle is spring-based and I am using spring-dm.

Thanks alot
karan
(jaskaran81@yahoo.com)

[Reply](#)



Anonymous 5 May 2012 07:04

Hello,

For anyone who is interested this is the way to access the repository in a spring bundle:

1) Update the manifest:

```
Import-Package:org.apache.sling.jcr.api,javax.jcr
```

2) Update the spring xml to get the sling repository:

3) Sample Code in the class:

```
session = slingRepository.loginAdministrative(null);  
Node root = session.getRootNode();  
Node test = root.addNode("test");  
session.save();  
session.logout();
```

Thanks

[Reply](#)



Anonymous 5 May 2012 07:06

sorry point 2 is incomplete

here you go:

2) Update the spring xml to get the sling repository:

[Reply](#)



Scott Macdonald 7 May 2012 10:40

Great suggestion that you specified here.

[Reply](#)



Akash 9 May 2012 20:35

Hi Scott

Can you please tell me how to create a sling servlet in day cq5.5? I am new to day cq , i am using scr annotations to create sling servlet. But scr annotations are not recognized by CQ. Its throwing exception. I am creating sling servlet using crxde.

Thanks

Akash

[Reply](#)



Hemant Bellani 12 May 2012 22:03

Hi Scott,

At onset, sincere thanks for the above article, and answers to questions around accessing the repository from within a Spring Bundle.

I have a slightly different problem, I have created a simple OSGI bundle(not a listener or a scheduler or a service). Request you to please share some code-snippet, if possible to access the repository. I understood the step 1 for importing the package in the manifest file. However, i require a snippet to move further.

All help is much appreciated !

Thanks,
Hemant

[Reply](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Picture Window template. Powered by Blogger.