



CONTENTS INCLUDE:

- What is ColdFusion?
- ColdFusion 9 Exposed Services
- Setting up ColdFusion as a Service
- SOAP vs REST Web Services
- Retrieving and Managing POP Email
- Hot Tips and more...

Leveraging ColdFusion 9 Exposed Services from Java

By Mark Mandel

WHAT IS COLDFUSION?

ColdFusion is a Web application development server based on J2EE technology, that allows for system programming through the ColdFusion Markup language.

ColdFusion comes bundled with multiple services to simplify the development process of Web applications, including PDF manipulation, image manipulation, integrated search support, Microsoft Office integration, Caching, Encryption services and many more.

You can download and use ColdFusion 9 for free for 30 days <http://www.adobe.com/go/trycoldfusion>

COLDFUSION 9 EXPOSED SERVICES

ColdFusion Exposed Services are a series of Web Services that are available as a new core feature in ColdFusion 9, that expose many of its in built capabilities, so that other languages and technologies can take advantage of them.

These Web Services can be consumed as either SOAP based Web Services, or as REST Web Services, depending on what is the best fit for your system.

The following capabilities are exposed through the ColdFusion as a Service Web Services:

- Creating PDFs from HTML
- Editing existing PDFs, including:
 - merging PDFs
 - deleting pages
 - extracting pages
 - creating thumbnails
 - adding watermarks
 - processing DDX
- Manipulating Images, including:
 - drawing on Images
 - cropping
 - scaling
- Retrieving meta data from Images
- Sending and receiving Email
- Generating Graphical Charts

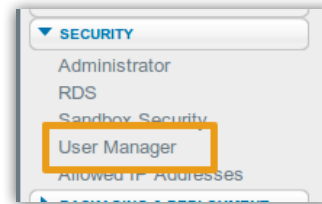
SETTING UP COLDFUSION AS A SERVICE

By default, ColdFusion as a Service is not set up to be publicly available, for security reasons. To that end, we will need to go into the ColdFusion Administrator to enable access to these Web Services and set up a user who has access to the ColdFusion Web Services, and grant them access.

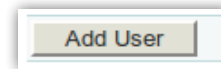
To create this user, point your browser to <http://localhost/CFIDE/administrator>, and log in with the password you set up when you first installed ColdFusion.

(We will assume that your ColdFusion install has been set up locally from here on in.)

Navigate to Security > User Manager through the left hand side menu.



Click the "Add User" Button to open the form to add a User.



Under "User Authentication", fill in the user and password fields with "cfaas". This will create a new user named "cfaas" with a password of "cfaas". At a later date you may wish to change this to something more secure, but this serves well for demonstrative purposes.

User Authentication (Add Mode)

*User name:

*Password:

*Confirm Password:

Description:

At the bottom of the page is a section entitled "Exposed Services", in which we want to move all the services from

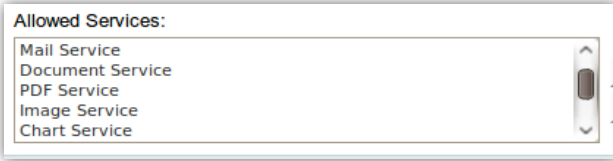
ADOBE® COLD FUSION® 9

Less code. Less time.
More efficient. More features.
See why less is more.

ADOBE® FLASH® PLATFORM

Try it today.
www.adobe.com/go/trycoldfusion

“Prohibited Services” to “Allowed Services”. This gives our user “cfaas” access to all the Services that ship with ColdFusion.



Hit “Save” when this is complete to save the user. You have now successfully created the user for accessing the ColdFusion as a Service Web Services.

Hot Tip The administration section named “Allowed IP Addresses” can also be used to limit the IP addresses the ColdFusion Services can be accessed from. Depending on your set up, you may need to set up access to the IP address the request is coming from.

SOAP VS REST WEB SERVICES

Generally speaking, Web Services are accessed via SOAP, or they can be accessed via REST. ColdFusion Services can actually be accessed via either, however for this Refcard, we will be accessing the Web Service via SOAP, as there are some limitations to what can be accomplished with ColdFusion Services via their REST interface.

GENERATING THE WEB SERVICE CODE WITH WSIMPORT

To generate the Java code to access the ColdFusion Web Services, we run the tool “wsimport” that comes with the Java SDK from the root of our Java project. There is an assumption that the Java source will be in a “src” directory, and the compiled .class files will be in a “bin” directory.

By pointing this tool at Web Services, it will generate all the Java code that is required to interact with the ColdFusion Web Services.

Hot Tip If you are using ANT or Maven to build your Java projects there are integrated tasks for both of these build tools for executing wsimport against Web Services.

Since there are several end points for interaction with ColdFusion we will have to do this several times.

Also, since ColdFusion Services share components, it is easier to generate the Java code into separate packages for each service, than to use the default generated packages.

Hot Tip For more information on wsimport and consuming Web Services with Java, see: <http://java.sun.com/javase/6/docs/technotes/tools/share/wsimport.html>

GENERATING A PDF FROM HTML

First thing we will do is generate the required Java source for the Document package, like so:

```
wsimport http://localhost/CFIDE/services/document.cfc?wsdl -d bin/ -s src/ -p com.coldfusion.document
```

This will create the required Java code to access the Document ColdFusion Web Service.

One of the easiest ways of creating a brand new PDF Document is by converting an HTML document into a PDF via ColdFusion’s Document Web Service.

If you browse to <http://localhost/CFIDE/services/document.cfc>, you will get a basic view of the methods that are available on this Web Service, and also the arguments that you are able to take advantage of when calling this Web Service.

Except for “serviceusername” and “servicepassword”, the Document Web Service’s arguments corresponds directly to a ColdFusion Tag called “cfdocument”, for which the documentation can be found here:

http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7c21.html

You can see there are a variety of options for generating a PDF from HTML, which give us a lot of control over what gets generated in our PDF.

Let us attempt to generate a simple “Hello World!” PDF file. First we will write some HTML:

```
String html = "<html>" +
    "<head>" +
    "<title>Hello World!</title>" +
    "</head>" +
    "<body>" +
    "<p>Hello World!</p>" +
    "</body>" +
    "</html>";
```

We create a new DocumentService, and request a Document object from it:

```
DocumentService service = new DocumentService();
Document document = service.getDocumentCfc();
```

There are a wide variety of parameters that can be passed into the “document.generate()” method, but there are only a few that we are concerned with at the moment:

- serviceUsername – this is the username for our ColdFusion as a Service User.
- servicePassword – this is the password for our ColdFusion as a Service User.
- content – the HTML content we are passing through.

Hot Tip Pass “null” into the arguments of the ColdFusion Web Service methods in which you wish to use the defaults values.

```
String result = document.generate("cfaas", "cfaas", null, html, null,
    null, null, null, null, null, null, null, null, null, null,
    null, null, null, null, null, null, null, null, null, null,
    null, null, null, null);
System.out.println(result);
```

This will return a String, for example: http://localhost/CFFileServlet/_cfserVICelayer/_cf3733600242229376232.pdf

Which is a URL to a temporary download of the PDF you have generated. If you browse to that URL, it should open a PDF that says “Hello World!”.

To get more complicated with the PDF generation, we can add a header to our document.

To do this we need to create a List of "DocumentItem", with some values set on the objects.

You can find the documentation for the "DocumentItem" object here:

http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7758.html

For example:

```
Documentitem item = new Documentitem();
item.setType("header");
item.setContent("My Document Header");
List<Documentitem> documentItems = new ArrayList<Documentitem>();
documentItems.add(item);
String result = document.generate("cfaas", "cfaas", null, html, null,
null, null, null, null, null, null, null, null, null, null, null, null,
null, null, null, null, null, null, null, null, null, null, null, null,
null, null, null, null, documentItems);
System.out.println(result);
```

This creates a header and puts the content of "My Document Header", and passes it through to "generate" method. If you browse to the URL that is returned, a PDF with the header specified should be displayed.

There are many other things we can use to control how we author our PDF as well, but this should give you some of the basics to get you started.

MANIPULATING PDFs

Not only can ColdFusion Services create PDFs on the fly, it can also manipulate them in a variety of ways through the PDF ColdFusion Web Service.

To generate the required Java code to consume the PDF Web Service, run the following wsimport command:

```
wsimport http://localhost/CFIDE/services/pdf.cfc?wsdl -d bin/ -s src/
-p com.coldfusion.pdf
```

If you browse to <http://localhost/CFIDE/services/pdf.cfc> you will notice that with this command there are multiple methods that can be taken advantage of to manipulate PDFs.

- **addwatermark()** – Add a watermark to the PDF.
- **deletepages()** – Delete pages from a PDF.
- **extractPages()** – Extract pages from a single PDF into a new PDF document.
- **getinfo()** – Retrieve information about the PDF document.
- **mergeFiles()** – Merge multiple PDFs into a single document.
- **mergespecificpages()** – Merge specific pages from multiple documents into a single PDF.
- **processDDX()** – Process DDX operations on PDF files to perform more complex actions on PDF documents.
- **protect()** – Password protect PDFs, encrypt them and set security permissions.
- **removewatermark()** – Remove a watermark from a PDF.
- **setinfo()** – Set information about a PDF document.
- **Thumbnail()** – Generate thumbnails from pages in a PDF document.

These operations correspond to the "CFPDF" ColdFusion Custom Tag, for which the documentation can be found here:

http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7995.html

The important thing to note with all of these PDF services is that the ColdFusion server will need to be able to access the file path that the PDF resides on.

To give you an example, the PDF service will be used to

retrieve information about PDF that can be downloaded from Adobe.com, entitled "Installing and Using ColdFusion". Create a PdfService like you would do normally:

```
PdfService service = new PdfService();
Pdf pdf = service.getPdfCfc();
```

From there we use the "getInfo()" method to retrieve a "Pdfinfo" object we can then introspect for data about the PDF:

```
Pdfinfo info = pdf.getInfo("cfaas", "cfaas", "/home/mark/install.pdf",
null);
System.out.println("Date Created: " + info.getCreated());
System.out.println("Application: " + info.getApplication());
System.out.println("Author: " + info.getAuthor());
System.out.println("Title: " + info.getTitle());
System.out.println("Total Pages: " + info.getTotalPages());
This should output the following:
Date Created: D:20070613135822Z
Application: FrameMaker 7.2
Author: Adobe Systems, Incorporated
Title: install.book
Total Pages: 112
```

Which gives us some basic data about the PDF in question.

If the built in functions for manipulating PDFs don't fit all your needs, DDX XML syntax can be used to do more advanced PDF creation and editing. The DDX Reference can be found here: http://www.adobe.com/go/learn_lc_ddx_9

SENDING EMAIL THROUGH COLD FUSION

Send email is something that ColdFusion does very well.

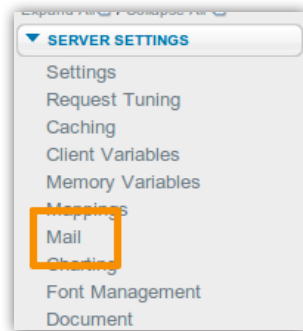
It comes built with an integrated mail system with logging capabilities, unsent email management and much more, which can be administered through the ColdFusion Administrator that was looked at earlier.

Before you can use the Mail Web Service, we will need to set up a SMTP server in the ColdFusion Administrator.

To configure the SMTP server, go to the ColdFusion Administrator and login as you did before.

<http://localhost/CFIDE/administrator>

Navigate to Server Settings > Mail through the left hand side menu.



Normally ColdFusion will send email asynchronously, however the Mail Web Service does not currently expose that functionality. Therefore enter a valid SMTP server for the "Mail Server", as the Mail ColdFusion Web Service will throw an exception if the SMTP server cannot be reached.

For development and/or testing, you can use a product like fakemail (<http://sourceforge.net/projects/fakemail/>), which is a SMTP server that will write incoming emails to a file.

To generate the required Java code to consume the Mail Web Service, run the following wsimport command:

```
wsimport http://localhost/CFIDE/services/mail.cfc?wsdl -d bin/ -s src/ -p com.coldfusion.mail
```

Browsing to <http://localhost/CFIDE/services/mail.cfc> you will see there is only one method, "send", that enables you to send an email.

This operation corresponds directly with the "cfmail" custom tag in ColdFusion, for which the documentation can be found here:

http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7f8e.html

Once the Java Code has been generated, we create a MailService like we would normally:

```
MailService service = new MailService();
Mail mail = service.getMailCfc();
```

From there, we call the "send()" command on Mail object:

```
mail.send("cfaas", "cfaas", "", null, null, null, "mark.mandel@gmail.com", "mark.mandel@gmail.com", null, null, "This is my test email", "Hello from ColdFusion!\n\nFrom Mark.", null, null, null, null, null, null, null, null, null, null, null);
System.out.println("Email Sent!");
```

You will notice that the third argument is "", rather than null. There is an issue in the ColdFusion Mail Web Service that throws an error when "null" is passed into the SMTP server argument. To use the SMTP server as it is defined in the ColdFusion Administrator, we can pass in "".

An email should now have been sent to mark.mandel@gmail.com.

RETRIEVING AND MANAGING POP EMAIL

ColdFusion also provides an easy way to use Web Service to check POP mail in-boxes, as well as delete email from them.

To generate the Java Code for the POP Web Service, run the following wsimport command:

```
wsimport http://localhost/CFIDE/services/pop.cfc?wsdl -d bin/ -s src/ -p com.coldfusion.pop
```

Browsing to <http://localhost/CFIDE/services/pop.cfc>, we can see that there are several methods for interacting with POP mail boxes.

To get a list of all the headers that exist within a single POP mailbox, the method "getHeaderOnly" is available.

To get all the headers, we pass in the POP mail details, as well as the POP username and password.

```
List<Popheader> records = pop.getHeaderOnly("cfaas", "cfaas", "mail.server.com", "110", "name@server.com", "password", null, null, null, null, null);
for(Popheader header : records)
{
    System.out.println("Subject: " + header.getSubject() + ", From: " + header.getFrom());
}
```

This will output all the email subjects and sender email addresses that are currently in the POP mailbox.

This method is useful if you wish to simply list all the emails that are currently in a POP mailbox, without wasting the bandwidth to retrieve the full message contents, attachments and the other data that can be sent along with an email.

To retrieve all the details of an email, the "getAll" method is also available.

In this instance we are going to pass along the UID of an Email to only retrieve a single email, which is the last parameter of this method:

```
List<Poprecord> results = pop.getAll("cfaas", "cfaas", "mail.server.com", "110", "name@server.com", "password", null, null, null, null, "757CC158D5154C6B97CC798C68E15EE3");
Poprecord result = results.get(0);
System.out.println(result.getSubject());
System.out.println(result.getBody());
```

Which will output both the subject and the full body of the Email. From the "Poprecord" object details such as attachments are also available, as well as much more.

Finally, if there is a need to delete a Mail record from the POP mailbox, the "delete" command is also available.

To delete the Mail message that was being read earlier, we can pass across the UID of that Email message.

```
pop.delete("cfaas", "cfaas", "mail.server.com", "110", "name@server.com", "password", null, null, null, null, "757CC158D5154C6B97CC798C68E15EE3");
```

For more details on this Web Service, it corresponds to the "cfpop" ColdFusion Custom tag, for which the documentation can be found here: http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7f96.html

IMAGE MANIPULATION

The Image ColdFusion Web Service provides a variety of methods for easily manipulating existing images, as well as retrieving meta information about images.

By browsing to <http://localhost/CFIDE/services/image.cfc>, you can view the full number of methods. The details of each of these methods can be viewed via the documentation for the "cfimage" custom tag that can be found here http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7945.html

For sample data, we will use a picture of an Australian Terrier I took:



As usual, the Java code is generated through the wsimport tool in the package "com.coldfusion.image":

```
wsimport http://localhost/CFIDE/services/image.cfc?wsdl -d bin/ -s src/ -p com.coldfusion.image
```

Once that is completed, we can retrieve the EXIF data from this image using the "GetEXIFMetaData" method that is available on the Image Web Service, like so:

```
ImageService service = new ImageService();
Image image = service.getImageCfc();
List<Element> results = image.getEXIFMetaData("cfaas", "cfaas", "/home/mark/00027.jpg");
for(Element result : results)
{
    System.out.println(result.getKey() + ": " + result.getValue());
}
```



Much like the PDF Web Service, the Imaging Web Service can only operate on Image files that are accessible from its local file path.

This will output a variety of EXIF information, such as:

```
White Balance: Auto white balance
Exposure Mode: Auto exposur
Focal Plane X Resolution: 73/912000 inches
Exposure Time: 0.02 sec
File Source: Digital Still Camera (DSC)
Shutter Speed Value: 1/50 sec
Make: Canon
Focal Length: 22.5 mm
Sensing Method: One-chip color area sensor
Compressed Bits Per Pixel: 3 bits/pixel
Y Resolution: 180 dots per inch
Compression: JPEG (old-style)
ISO Speed Ratings: 800
Color Space: sRGB
X Resolution: 180 dots per inch
F-Number: F4.9
Exif Version: 2.21
Exif Image Width: 2736 pixels
Model: Canon PowerShot S90
Aperture Value: F4.9
Date/Time: 2009:12:25 14:48:25
```

For an example of image manipulation, the original image will be converted to gray scale.

```
ImageService service = new ImageService();
Image image = service.getImageCfc();
String result = image.grayScale("cfaas", "cfaas", "/home/mark/00027.jpg");
System.out.println(result);
```

A URL is returned from this method, which provides this image converted to gray scale.



There are many options for manipulating images via this Web Service, and this is just one example of the capabilities that are available.



There is a "batchOperation" method for executing multiple image manipulation methods together.

GRAPH AND CHART CREATION

ColdFusion also has an integrated Chart and Graph creation library that can be accessed via the ColdFusion as a Service Web Services.

There is only one method "generate" that is available on this Web Service, which can be viewed by browsing to: <http://localhost/CFIDE/services/chart.cfc>

There is a wide variety of parameters that can be passed through to this service, but we will concentrate on the most used ones:

- format – Charts can be generated either in Flash, Jpg

or PNG formats. Of the three, Flash tends to be the most aesthetically pleasing.

- show3d – This parameter does nothing more than render the chart in 3d, which tends to be a nicer looking chart.
- title – The title for the chart that is being created.
- Chartseries – The data for the chart that is being created.

To generate a simple bar chart, we must first generate the Java source code for the Web Service:

```
wsimport http://localhost/CFIDE/services/chart.cfc?wsdl -d bin/ -s src/ -p com.coldfusion.chart
```

To access the Chart Web Service, we create a "ChartService" like normal:

```
ChartService service = new ChartService();
Chart chart = service.getChartCfc();
```

However, providing the data for the "chart" object is a bit more complicated than usual.

First we need to create an instance of the Class "ArrayOfXsdAnyType", which is essentially the List of "Chartdata" items that are used to store the data that is used in the chart.

The "Chartdata" is created and added to our data collection like so:

```
Chartdata data = new Chartdata();
data.setItem("Dogs");
data.setValue("15");
array.getItem().add(data);

data = new Chartdata();
data.setItem("Cats");
data.setValue("5");
array.getItem().add(data);

data = new Chartdata();
data.setItem("Goats");
data.setValue("10");
array.getItem().add(data);
```

Next, this data needs to be stored in a "Chartseries" object. The job of this object is to define what sort of chart is to be created. In this instance, a bar chart will be created.

```
Chartseries series = new Chartseries();
series.setChartdata(array);
series.setType("bar");
```

To pass the "Chartseries" into the Web Service "chart" object, we have to put it inside a "List<Chartseries>".

```
List<Chartseries> chartSeries = new ArrayList<Chartseries>();
chartSeries.add(series);
```



Multiple Chart Series can be used for charts such as line graphs to display multiple sets of data on the same chart.

Finally we pass this through to the "generate" function, also telling it to generate a Flash chart, to turn on 3D and add a title of "My Chart Title".

```
String result = chart.generate("cfaas", "cfaas", "", "", "", "", "", "", "", "", "flash", "", "", "", "", "true", "", "", "", "", "My Chart Title", "", "", "", "", "", "", chartSeries);
System.out.println(result);
```

This will return a URL in which the Chart can be viewed.

My Chart Title

Category	Value
Dogs	16
Cats	6
Goats	11

Hot Tip Passing "null" values into the generate() function causes errors to be thrown. Use "" instead.

It is very easy to switch this to an alternate chart type. For example, if the data above was to be displayed as a pie chart, the "type" attribute on the "Chartseries" object only need be changed to "pie":

My Chart Title

■ Dogs ■ Cats ■ Goats

```
series.setType("pie");
```

Details on the CFCHART custom tag that this Web Service corresponds to can be found here:
http://help.adobe.com/en_US/ColdFusion/9.0/CFMLRef/WSc3ff6d0ea77859461172e0811cbec22c24-7930.html

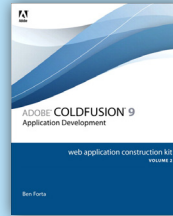
ABOUT THE AUTHOR



Mark Mandel is a full time consultant and lead developer on several open source projects, most notably ColdSpring, Transfer ORM, JavaLoader, and ColdDoc. He has been working with ColdFusion for a number of years, including at his very own dot com back in the late '90s.

Mark can often be found blogging at www.compoundtheory.com, which has housed his thoughts on ColdFusion, Java, and various other aspects of software development for several years. He can also be found as a regular poster on ColdFusion mailing lists as well as generally causing havoc in the #coldfusion channel on Dalnet IRC network.

RECOMMENDED BOOKS



Written by the best known and most trusted name in the ColdFusion community, Ben Forta, The ColdFusion Web Application Construction Kit is the best-selling ColdFusion series of all time - the books that most ColdFusion developers used to learn the product.

BUY NOW

books.dzone.com/books/adobe-coldfusion-9

Browse our collection of 100 Free Cheat Sheets

Free PDF

Upcoming Refcardz

- Apache Ant
- Hadoop
- Spring Security
- Subversion



DZone communities deliver over 6 million pages each month to more than 3.3 million software developers, architects and decision makers. DZone offers something for everyone, including news, tutorials, cheatsheets, blogs, feature articles, source code and more. **"DZone is a developer's dream,"** says PC Magazine.

DZone, Inc.
 140 Preston Executive Dr.
 Suite 100
 Cary, NC 27513
 888.678.0399
 919.678.0300
Refcardz Feedback Welcome
refcardz@dzone.com
Sponsorship Opportunities
sales@dzone.com

ISBN-13: 978-1-934238-75-2
 ISBN-10: 1-934238-75-9

50795

\$7.95