

**May  
2025**



**COLD FUSION**  
user group **Seattle**



# Agenda

- Introductions
- Adobe ColdFusion Summit 2025
- Adding eCommerce to your ColdFusion app using the Authorize.net API
- Tag/Function O'Month
- Questions/Answers/Help Needed
- Next Meeting



# Adobe ColdFusion Summit 2025

- Location: Resorts World – Las Vegas
- Conference Dates: September 22<sup>nd</sup> and 23<sup>rd</sup>
- Certification Dates: September 21<sup>st</sup> and 24<sup>th</sup>
- More Info and Register: <https://cfsummit.adobeevents.com/>



# Adding eCommerce to your ColdFusion app using the Authorize.net API

**Presenter:** Leon O'Daniel

## **Current Roles:**

- Senior Web Application Developer – Boeing Commercial Airplanes Digital Engineering
- Vice President – O'Daniel Designs

## **ColdFusion Experience**

- Developing Web Applications using Adobe ColdFusion since 1997
- Started with Adobe ColdFusion 3.1



# Adding eCommerce to your ColdFusion app using the Authorize.net API

What I will cover:

- Setting up a developer sandbox account
- Obtaining an Authorize.NET API Login ID and Transaction Key
- Safeguarding your Authorize.NET API Login ID and Transaction Key
- Getting test credit card numbers
- Passing Shopping Cart Information to Authorize.net
- Processing the Authorize.net response
- Transitioning from Authorize.net AIM to Authorize.NET API
- Other payment processing considerations



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Setting up an Authorize.NET developer sandbox account

- Go to [https://developer.authorize.net/hello\\_world/sandbox.html](https://developer.authorize.net/hello_world/sandbox.html)
- Complete the form and submit
- Your account will be created



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Obtaining an Authorize.NET API Login ID and Transaction Key

- Login to your Sandbox account at <https://logintest.authorize.net/>
- In the left side-bar, select **Settings** (under the *Account* heading)
- In the Security Settings section, select **API Credentials & Keys**
- Your Sandbox account's *API Login ID* will never change, but your *Transaction Key* will
- Create your Transaction Key in the *Create New Keys* section of this form
  - Click the **New Transaction Key** radio button
  - Click the **Submit** button
  - You will be presented with the option to *Disable Old Transaction Key Immediately*
  - Make sure to copy the new transaction key to somewhere safe – it won't be presented again



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Safeguarding your API Login ID and Transaction Key

Storing your API Login ID and Transaction Key in your source code is a BAD thing:

- It's a violation of PCI (payment card industry) compliance requirements
- The credentials can more easily be compromised
  - Source Control Compromised
  - Server Compromised



# Adding eCommerce to your ColdFusion app using the Authorize.net API

Safeguarding your API Login ID and Transaction Key

*The solution:*

- Use a secrets manager – or –
- **Store the credentials encrypted in the database**



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Safeguarding your API Login ID and Transaction Key

*Store the credentials encrypted in the database*

```
<cfscript>
    VARIABLES.APILoginId = "Your API Login ID";
    VARIABLES.TransactionKey = "Your Transaction Key";
    VARIABLES.EncryptionKey = generateSecretKey('AES');

    // Encrypt Using AES Cipher Block Chaining (CBC) mode
    VARIABLES.encAPILoginId = Encrypt(VARIABLES.APILoginId, VARIABLES.EncryptionKey, 'AES/CBC/PKCS5Padding', 'HEX');
    VARIABLES.encTransactionKey = Encrypt(VARIABLES.TransactionKey, VARIABLES.EncryptionKey, 'AES/CBC/PKCS5Padding', 'HEX');

    // Decrypt Using AES Cipher Block Chaining (CBC) mode
    VARIABLES.decAPILoginId = decrypt(VARIABLES.encAPILoginId, VARIABLES.EncryptionKey, 'AES/CBC/PKCS5Padding', 'HEX');
    VARIABLES.decTransactionKey = decrypt(VARIABLES.encTransactionKey, VARIABLES.EncryptionKey, 'AES/CBC/PKCS5Padding', 'HEX');
</cfscript>
```



# Adding eCommerce to your ColdFusion app using the Authorize.net API

Safeguarding your API Login ID and Transaction Key

*Store the credentials demo*

- *authnet/generateEncryptedCredentials.cfm*
- *authnet/getAuthNetCredentials.cfm*
- *AuthorizeNetService.cfc – getCredentials method*
- *Stored Proc: [dbo].[getAuthNetCredentials]*



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Getting Test Credit Card Numbers

*Yes! They do Exist.*

*Go to the Authorize.NET testing guide page at*

*[https://developer.authorize.net/hello\\_world/testing\\_guide.html](https://developer.authorize.net/hello_world/testing_guide.html)*



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Compare/Contrast Authorize.NET AIM vs Authorize.NET API

- *Endpoints are different:*
  - *Production*
    - AIM: <https://secure2.authorize.net/gateway/transact.dll>
    - API: <https://api.authorize.net/xml/v1/request.api>
  - *Sandbox*
    - AIM: <https://test.authorize.net/gateway/transact.dll>
    - API: <https://apitest.authorize.net/xml/v1/request.api>
- *Data is sent to Authorize.NET differently*



# Adding eCommerce to your ColdFusion app using the Authorize.net API

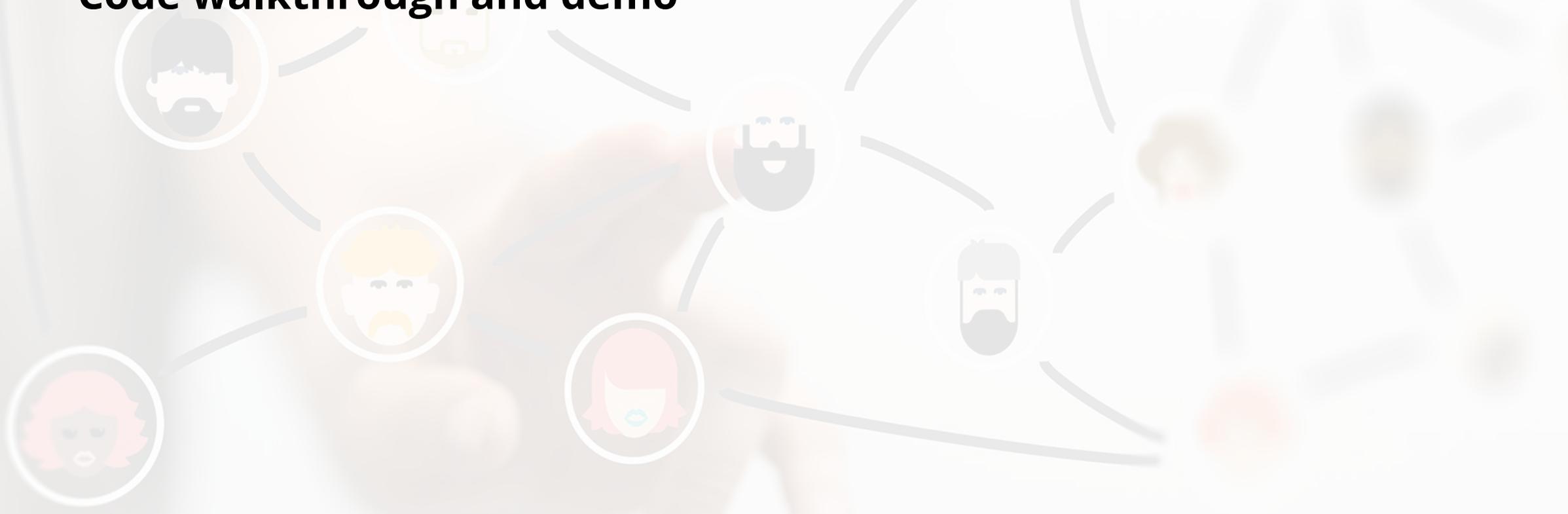
## Compare/Contrast Authorize.NET AIM vs Authorize.NET API

- *Data is sent to Authorize.NET differently*
  - *Authorize.NET AIM – via HTTP POST as a series of form fields*
  - *Authorize.NET API – via HTTP POST using XML or JSON*
- *The error text is included with the error code with Authorize.NET API*
- *Data is returned differently*
  - *Authorize.NET AIM: A string of data is returned in pipe-delimited format*
  - *Authorize.NET API: XML or JSON is returned*



# Adding eCommerce to your ColdFusion app using the Authorize.net API

Code walkthrough and demo





# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Including Shopping Cart Items in the Authorize.NET API Request

*Make sure to include this between the <Order> and <Tax> XML tags:*

```
<lineltems>  
  <lineltem>  
    <itemld>1</itemld>  
    <name>vase</name>  
    <description>Cannes logo</description>  
    <quantity>18</quantity>  
    <unitPrice>45.00</unitPrice>  
  </lineltem>  
</lineltems>
```



# Adding eCommerce to your ColdFusion app using the Authorize.net API

## Resources:

- **Authorize.NET AIM Guide**

([https://www.authorize.net/content/dam/anet-redesign/documents/AIM\\_guide.pdf](https://www.authorize.net/content/dam/anet-redesign/documents/AIM_guide.pdf))

This has the best description of the values for the Authorize.Net Response variables starting on page 50

- **Authorize.NET API Reference**

(<https://developer.authorize.net/api/reference/index.html>)



# Tag/Function O'Month

Presenter: Will Frankhouser – Adobe Certified Professional – ColdFusion





# Tag/Function O'Month

Time to REview with some **REFind**, **REMatch** and **REReplace**!

- Adobe ColdFusion by default uses [Perl-regex engine](#)
  - - Starting in CF2018 Update 5, useJavaAsRegexEngine flag added
- **NoCase** versions of the functions too
- [REFind](#)(reg\_expression, string [, start, returnsubexpressions, scope ] )
- [REMatch](#)(reg\_expression, string)
- [REReplace](#)(string, regex, substring, [scope])
- ✦ Can use regex with [Character Classes](#) [[:space:]]



# Questions/Answers/ Help Needed

This is your opportunity to ask any ColdFusion question or request help for a ColdFusion-related issue.



# Next Meeting

**June 11, 2025 – 5:00 PM PDT (on-line via Zoom).**

*How to use OAuth 2.0 to send mail using Microsoft 365*

Michael Hayes – media3

- Basic OAuth 2.0 Overview
- How to send mail using Microsoft 365 and CFMail

*ColdFusion Tag/Function O' Month*