**DwBP - 6 - Using the Player Catalog**

In the last video, we built a great foundation on which to continue our development with Brightcove Player. In this video, we're going to use the Player Catalog to change the video that is currently playing in the Player based upon a user interaction-- in this case, a user clicking on a button. We will use a couple of methods to get a video and then load that video into the Player. Let's go ahead and get started.

One of the things we're going to need to do is access the Player throughout all of our JavaScript code. So the first thing to change is how the myPlayer variable is defined. We’ll move the variable declaration and add a var statement. No assignment to the variable is made because it will be assigned a value later. Now myPlayer is going to have meaning throughout our whole script block.

The video change will happen on a user interaction. We’ll put in a button for the user to click. The clickable text on the button is going to say change video. And it's going to call an Event Handler called changeVideo.

Now the function changeVideo needs to be defined. What we want to do is use the Player Catalog and call the getVideo() method. We know that myPlayer now has meaning throughout the whole script block, so we can use the catalog and the method is getVideo.

The getVideo method has two arguments. The first one is the videoID for the video to be retrieved, and the second is a callback function which will be executed only after the video is retrieved. Let's put in a videoID from the proper account.

Now the anonymous callback function needs to be built. Two parameters are passed to this function automatically. The first one is an error, which I'm hoping will just see that it's null. And the second is a videoObject returned by the catalog.

And now let’s code what we want the callback function to do. Remember, the callback function is only called once the video is returned to the client. If there is an error, we're going to display the error to the console using a log method. And hopefully, again, that'll be null.

Next we're going to load that videoObject that's returned by getVideo into the Player. To do this the load() method is used.

Then we will mute the player, and lastly play the video.

Save that, and test by browsing the page.

So here's one video playing. Now click the Change Video button. And there is the new video playing. Super, it is working.

Let’s take a moment to review what we did with a couple of slides. And here we see the explanation of the Player Catalog. It's basically a helper library for making requests to the Video Cloud Catalog.

And you see at the bottom there, we've used two of those methods. We used getVideo, which uses a video ID and a callback function. The Load method was also used and it puts the returned video object in Brightcove Player.

Here we see the video object that's actually returned by the getVideo method. Note the response and the response text. That's actually the information about the video that's brought in. We see the actual type is an XML HTTP request object that is returned.

We've now completed our second task. And what we did is we used the Catalog object to be able to dynamically load and play a video that was not originally in the Player.

In our next video, we're going to use metadata-- information about the video-- in our HTML page. Let's get started and see you there.