Purge an item from Wowza Streaming Engine
Media Cache
Originally Published on 10/06/2015 | Updated on 01/10/2019 1:37 pm PST

Use the Wowza Streaming Engine™ media server software Java API to purge assets from Media Cache.

Purge assets not in use

Use the following Java API methods to purge an asset from the cache if it's not being referenced or played by clients.

To purge an asset from the cache based on the asset `mediaName`, which is the asset stream name minus the prefix and after any play aliasing, use the following call:

```
mediaCache = MediaCacheImpl.getMediaCache();
mediaCache.flushItemFromCache(mediaName);
```

You can also use the following call that includes the asset stream name with the prefix. This will take the stream name (with the prefix) and run it through an aliasing system to resolve it to the proper media name, and then purge the asset from the cache:

```
mediaCache = MediaCacheImpl.getMediaCache();
mediaCache.flushStreamNameFromCache(appInstance, streamName);
```

Purge assets in use

**Note**: Purging assets that are in use requires Wowza Streaming Engine™ 4.3 or later.

Use the following Java API methods to purge an asset from the cache even if it's still being referenced or played by clients.

To purge an asset from the cache based on the asset `mediaName`, which is the asset stream name minus the prefix and after any play aliasing, use the following call:
You can also use the following call that includes the asset stream name with the prefix. This will take the stream name (with the prefix) and run it through an aliasing system to resolve it to the proper media name, and then purge the asset from the cache:

```java
MediaCache mediaCache = MediaCacheImpl.getMediaCache();
mediaCache.flushAndForceStreamNameFromCache(appInstance, streamName);
```

These methods purge the specified asset 30 seconds after the last currently connected client is disconnected. After the call is made, Media Cache will disconnect all currently connected clients, purge the asset from the cache, and then, if a new request is made for the asset, get new copy of the asset from the source. These calls are useful if you want to remove a specific asset and/or remove an asset so it can be refreshed when a new client requests it.

The following example HTTP provider uses the `flushAndForceItemFromCache` method:
while (itemNamesI.hasNext()) {
    String thisItem = itemNamesI.next();
    WMSLoggerFactory.getLogger(HTTPForceDelete.class).info("Item in list was called "+thisItem);
    if (thisItem.endsWith(itemName)) {
        foundItem = mediaCache.flushAndForceItemFromCache(thisItem);
    }
}

String retStr  = "Item "+itemName+" ";
if ( foundItem == true )
    retStr += "found."
else
    retStr += "NOT FOUND."

try {
    OutputStream out = resp.getOutputStream();
    byte[] outBytes = retStr.getBytes();
    out.write(outBytes);
} catch (Exception e) {
    WMSLoggerFactory.getLogger(null).error("HTTPForceDelete: "+e.toString());
}
}