Control aliasing of stream names with the
Wowza Streaming Engine Java API

This example demonstrates the basic framework for implementing the Wowza Streaming Engine™ Java API `IMediaStreamNameAliasProvider3` interface, which is used for aliasing stream names. Use this interface instead of the `StreamNameAlias AddOn` or `.stream` files when you need programmatic control of aliasing.

This interface provides several `resolvePlayAlias` methods, one of which is triggered each time a client plays a stream. The stream name that is sent to the Wowza media server by a client is run through the `IMediaStreamNameAliasProvider3` interface. If a match is found it’s transformed. If no match is found, an error is returned to the client. There several overloads of `resolvePlayAlias`, one for each client type.

The `resolveStreamAlias` methods are executed by the WebRTC subsystem for WebRTC streams or the MediaCaster subsystem. When you start a stream using `StreamManager`, `StartUpStreams.xml` or `MediaCaster API`, the MediaCaster subsystem is used to provide the following streaming functions; live stream repeater, SHOUTcast/IceCast re-streaming and RTSP/RTP re-streaming. These types of alias are generally used to turn complex url based stream names into simple names. For example, if you’re re-streaming a SHOUTcast stream that is hosted at the URL `http://192.168.1.7/myradiostation` then you can create a stream alias such as `myradiostation`. You can then use this alias name to refer to the SHOUTcast stream.

In multi-bitrate streaming, using a smil file (or client-side playlist for Flash dynamic streaming), the smil or playlist is resolved first then each item in the smil is resolved.

Notes:

- You can’t use this interface in the same application with the `StreamNameAlias AddOn`, and you can’t use `.stream` files if you’re using this interface.
- Aliasing only changes a stream name. You can’t use aliasing to change the IP address, domain name, or application name.
package com.wowza.wms.example.module;

import com.wowza.wms.application.*;
import com.wowza.wms.client.IClient;
import com.wowza.wms.httpstreamer.model.IHTTPStreamerSession;
import com.wowza.wms.mediacaster.IMediaCaster;
import com.wowza.wms.module.*;
import com.wowza.wms.rtp.model.RTPSession;
import com.wowza.wms.stream.IMediaStream;
import com.wowza.wms.stream.IMediaStreamNameAliasProvider3;
import com.wowza.wms.stream.livepacketizer.ILiveStreamPacketizer;

public class ModuleStreamNameAliasExample extends ModuleBase implements IMediaStreamNameAliasProvider3 {

  public void onAppStart(IApplicationInstance appInstance) {
    String fullname = appInstance.getApplication().getName() + "/
    + appInstance.getName();
    getLogger().info("onAppStart: " + fullname);

    appInstance.setStreamNameAliasProvider(this);
  }

  public String resolvePlayAlias(IApplicationInstance appInstance, String name, IClient client) {
    getLogger().info("Resolve Play Flash: " + name);
    return name;
  }

  public String resolvePlayAlias(IApplicationInstance appInstance, String name, IHTTPStreamerSession httpSession) {
    getLogger().info("Resolve Play HTTPSession: " + name);
    return name;
  }

  public String resolvePlayAlias(IApplicationInstance appInstance, String name, RTPSession rtpSession) {
    getLogger().info("Resolve Play RTPSession: " + name);
    return name;
  }

  public String resolvePlayAlias(IApplicationInstance appInstance, String name, ILiveStreamPacketizer liveStreamPacketizer) {
    getLogger().info("Resolve Play LiveStreamPacketizer: " + name);
    return name;
  }

  public String resolveStreamAlias(IApplicationInstance appInstance, String name, IMediaCaster mediaCaster) {
    getLogger().info("Resolve Stream Mediacaster: " + name);
  }
}
Add the following module last in the list in `/conf/[app-name]/Application.xml`:

```
ModuleStreamNameAliasExample
ModuleStreamNameAliasExample
com.wowza.wms.example.module.ModuleStreamNameAliasExample
```