

The following document describes how to convert applications that used the old v4 OpenLM APIs and adapt them to OpenLM Server v5.

You can also check out [this archive](#) which contains a sample project.

Contents:

[Converting XML API consumption](#)

[New link](#)

[Changing the code](#)

[Converting SOAP API consumption](#)

[New link](#)

[Changing the code](#)

[Code example](#)

Converting XML API consumption

New link

[http://\[openlm_server\]:5015/api/easyadminapi/postmessage](http://[openlm_server]:5015/api/easyadminapi/postmessage)

Changing the code

No code changes are needed if you previously used the XML API on port 7014. Just replace the old link ([http://\[openlm_server\]:7014/OpenLMServer](http://[openlm_server]:7014/OpenLMServer)) to the new link and everything will work as usual.

Converting SOAP API consumption

New link

[http://\[openlm_server\]:5015/api/easyadminapi/web/\[method\]](http://[openlm_server]:5015/api/easyadminapi/web/[method])

Changing the code

SOAP methods are no longer supported on OpenLM Server v5.x. In order to consume those methods, a web request to the new API endpoint needs to be sent. The request needs to be in JSON format. The response will be returned in JSON format as well. You can see a code example here for converting the old SOAP method *GetDenialsChart*.

Code example

```
private readonly AdminAPIConnector connector =
AdminAPIConnector.Instance;

[TestMethod]
public void GetLicenses()
{
    /* Old code for OpenLM server version 4.x or older for
SOAP API call
    AdminAPIClient client = new AdminAPIClient();
    var request = new LicenseInfoRequest {
        BaseInfo = new RequestBaseInfo()
    };
    LicensesResponse response = client.GetLicenses(request);*/

    //Start code for converted API SOAP call for OpenLM server
```

```

version 5.x
    var request = new LicenseInfoRequest
    {
        BaseInfo = connector.CreateBaseInfo()
    };
    LicensesResponse response =
connector.Get<LicensesResponse, LicenseInfoRequest>(request,
"GetLicenses");
    //End code for converted API SOAP call for OpenLM server
version 5.x

```

```

    Assert.IsNotNull(response);
    Assert.IsNull(response.Error);
}

```

```

using System.Text;
using System.Net;
using System.IO;
using APISamples.AdminAPI;
using Newtonsoft.Json;
using System;

```

```

namespace APISamples
{
    public class AdminAPIConnector
    {
        private static AdminAPIConnector instance;

        private const int _port = 5015;
        private const string _server = "localhost";
        private const string UserName = "Please fill user name";
        private const string Password = "Please fill password";

        private AdminAPIConnector() { }

        public static AdminAPIConnector Instance
        {

```

```

get
{
    if (instance == null)
    {
        instance = new AdminAPIConnector();
    }
    return instance;
}

internal RequestBaseInfo CreateBaseInfo()
{
    return new RequestBaseInfo
    {
        PagingData = new PagingData
        {
            StartRecord = 0,
            NumOfRecord = 10,
            Sort = new string[] { "value" },
            Direction = "desc"
        },

        UserLocalSettings = new UserLocalSettings
        {
            TimezoneStandardName = "UTC",
            ThousandsSeparator = ",",
            DecimalSeparator = ".",
            TimeFormat = "dd/mm/yyyy hh:mm:ss"
        },
        SessionData = new SessionRefresh { Refresh = true,
SessionID = instance.GetSessionID() }
    };
}

#region Private methods
private string SendRequest(string json, string method)
{
    //string url =

```

```

"http://localhost:5015/OpenLM.Server.Services/AdminAPI/web/"
    string url =
$"http://{_server}:{_port}/OpenLM.Server.Services/AdminAPI/web/{method
}";
    HttpWebRequest req =
(HttpWebRequest)WebRequest.Create(url);

    req.UseDefaultCredentials = true;
    byte[] requestBytes =
System.Text.Encoding.ASCII.GetBytes(json);
    req.Method = "POST";
    req.ContentType = "text/xml;charset=utf-8";
    req.ContentLength = requestBytes.Length;
    Stream requestStream = req.GetRequestStream();
    requestStream.Write(requestBytes, 0, requestBytes.Length);
    requestStream.Close();

    HttpWebResponse res = (HttpWebResponse)req.GetResponse();
    StreamReader sr = new
StreamReader(res.GetResponseStream(), System.Text.Encoding.Default);

    string s =
System.Text.Encoding.Default.GetString(ASCIIEncoding.Default.GetBytes(
sr.ReadToEnd()));
    sr.Close();
    res.Close();

    return s;
}

private string GetSessionID()
{
    LoginFormSettingsResponse response =
instance.GetLoginFormSettings();
    if (response.UserAuthenticationRequired)
    {
        return AdminAPIAuthentication(response.ShowWinAuth);
    }
}

```

```

        else//No authentication required
        {
            return string.Empty;
        }
    }

    private static string AdminAPIAuthentication(bool
useWindowsAuthentication)
    {
        var userAuthenticationRequest = new
UserAuthenticationRequest { TrustedAuthentication =
useWindowsAuthentication };//Windows authentication
        if (!useWindowsAuthentication)//OpenLM server
authentication
        {
            userAuthenticationRequest.UserName = UserName;
            userAuthenticationRequest.Password = Password;
        }/* If OpenLM server authentication is required uncheck
this code
        else
        {
            userAuthenticationRequest.UserName = UserName;
            userAuthenticationRequest.Password = Password;
            userAuthenticationRequest.TrustedAuthentication =
false;
        }*/
        UserAuthenticationResponse userAuthenticationResponse =
Instance.PerformUserAuthentication(userAuthenticationRequest);
        if (userAuthenticationResponse.Error != null)
        {
            throw new
Exception(userAuthenticationResponse.Error.Message);
        }
        return userAuthenticationResponse.SessionID;
    }
#endregion

//Convert API SOAP call from OpenLM version 4.x or older to

```

OpenLM version 5.x

```
    internal T Get<T,S>(S request, string methodName)
    {
        return
        JsonConvert.DeserializeObject<T>(SendRequest(JsonConvert.SerializeObject(request), methodName));
    }

    public LoginFormSettingsResponse GetLoginFormSettings()
    {
        var request = new LoginFormSettingsRequest();
        return Get<LoginFormSettingsResponse,
        LoginFormSettingsRequest>(request, "GetLoginFormSettings");
    }

    public UserAuthenticationResponse
    PerformUserAuthentication(UserAuthenticationRequest request)
    {
        return Get<UserAuthenticationResponse,
        UserAuthenticationRequest>(request, "PerformUserAuthentication");
    }
}
```