Qlik | Qonnections
Global Partner Summit 2015

10th Annual

Advanced QlikView Management Service API Usage

Jeff Goldberg & Jacob Vinzent

28 April, 2015

Legal Disclaimer

This Presentation contains forward-looking statements, including, but not limited to, statements regarding the value and effectiveness of Qlik's products, the introduction of product enhancements or additional products, Qlik's partner and customer relationships, and Qlik's growth, expansion and market leadership, that involve risks, uncertainties, assumptions and other factors which, if they do not materialize or prove correct, could cause Qlik's results to differ materially from those expressed or implied by such forward-looking statements. All statements, other than statements of historical fact, are statements that could be deemed forward-looking statements, including statements containing the words "predicts," "plan," "expects," "anticipates,“ “see,” "believes," "goal," "target," "estimate," "potential," "may", "will," "might," "could," and similar words. Qlik intends all such forward-looking statements to be covered by the safe harbor provisions for forward-looking statements contained in Section 21E of the Exchange Act and the Private Securities Litigation Reform Act of 1995. Actual results may differ materially from those projected in such statements due to various factors, including but not limited to: risks and uncertainties inherent in our business; our ability to attract new customers and retain existing customers; our ability to effectively sell, service and support our products; our ability to manage our international operations; our ability to compete effectively; our ability to develop and introduce new products and add-ons or enhancements to existing products; our ability to continue to promote and maintain our brand in a cost-effective manner; our ability to manage growth; our ability to attract and retain key personnel; the scope and validity of intellectual property rights applicable to our products; adverse economic conditions in general and adverse economic conditions specifically affecting the markets in which we operate; and other risks and uncertainties more fully described in Qlik's publicly available filings with the Securities and Exchange Commission. Past performance is not necessarily indicative of future results. The forward-looking statements included in this presentation represent Qlik's views as of the date of this presentation. Qlik anticipates that subsequent events and developments will cause its views to change. Qlik undertakes no intention or obligation to update or revise any forward-looking statements, whether as a result of new information, future events or otherwise. These forward-looking statements should not be relied upon as representing Qlik's views as of any date subsequent to the date of this presentation.

This Presentation should be read in conjunction with Qlik's periodic reports filed with the SEC (SEC Information), including the disclosures therein of certain factors which may affect Qlik's future performance. Individual statements appearing in this Presentation are intended to be read in conjunction with and in the context of the complete SEC Information documents in which they appear, rather than as stand-alone statements. **This presentation is intended to outline our general product direction and should not be relied on in making a purchase decision, as the development, release, and timing of any features or functionality described for our products remains at our sole discretion.**

Your Hosts for Today

Jeff Goldberg

- Senior Enterprise Architect
- Integration, Security, Deployment
- Qlik OEM Solution Architect 2013-2015
- I like to figure out how to make software do what I want.

Jacob Vinzent

- Senior Enterprise Architect
- Integration, Security, APIs
- Been 2 years with Qlik
- Came from the cloud industry
- Like to understand what Jeff tries to do with the software

Agenda

- A story about an admin and his opportunity to automate
- Background
  - What is it and what does it do?
- General use cases
- How our hero achieves his goals
- Conclusion

So there's this admin…

- He's constantly going into the QlikView® Management Console.
  - He's always provisioning users
  - Kicking off tasks when data warehouse jobs finish
  - Manually creating distributions of smaller data set apps for his customers
- What he wants to do…
  - Pass along QlikView user provisioning to the helpdesk
  - Sleep late instead of waking early to kick off reloads
  - Enable users to create new application data sets on the fly
  - Not being a bottleneck
  - Avoid Shadow IT in his organization

QRS API!

Background

- What is the QMS API?
  - QMS stands for QlikView Management Service
  - A web service to control all aspects of the QlikView Management Console (QMC)
- What can you do with the QMS API?
  - Use it to provision users to a QlikView document
  - Create a new tasks
  - Set up tenants in a QlikView deployment
  - Execute QlikView tasks using event driven execution (EDX)
  - And a whole lot more!
- What does QMS API require?
  - .Net 4.0 & Visual Studio
  - Connectivity to the server running the QMC

Use Cases

- In the enterprise
  - User and group provisioning on documents and distribution tasks
  - Triggered execution based on an event (system or user driven)
  - Provide an internal SaaS service to the organization

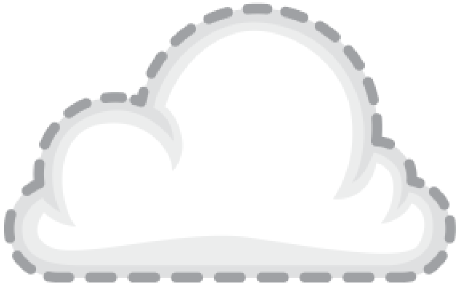
- In OEM
  - Host application integration to drive tenant administration of users and tasks
  - Provision the reporting capabilities together with the base solution

Back to Our Admin

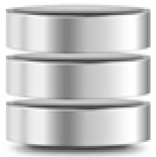
- Build a .Net app for an OEM solution provisioning
  - One app for each customer
  - Control the security via Ticket login
  - Make sure that the app is reloaded just after it is provisioned and after there on a regular basis

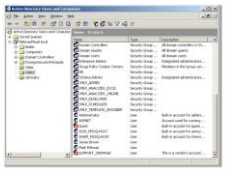
Image source. Google

QlikView in the Cloud Sign-up



End user sign-up





Customer DB

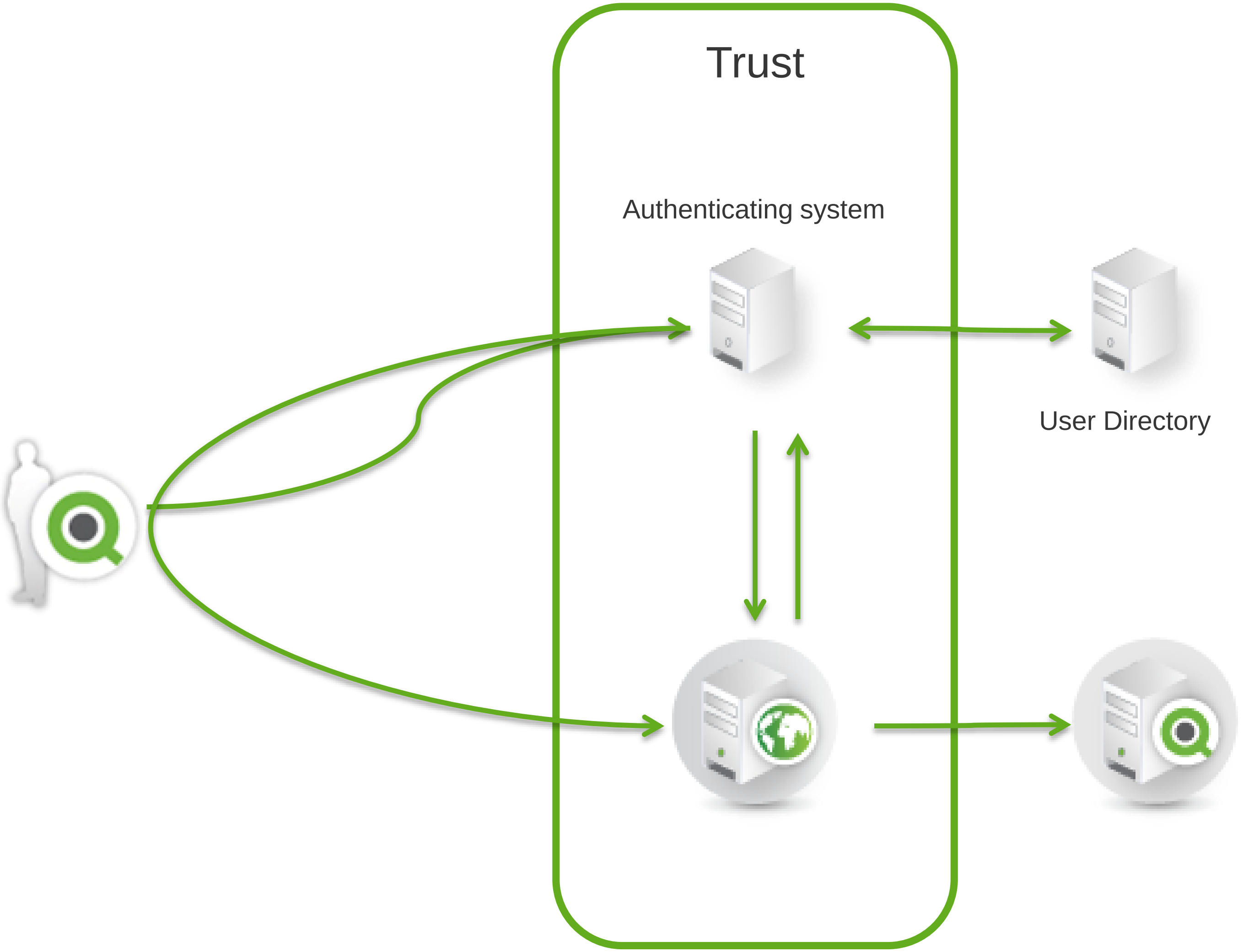
Management

AD

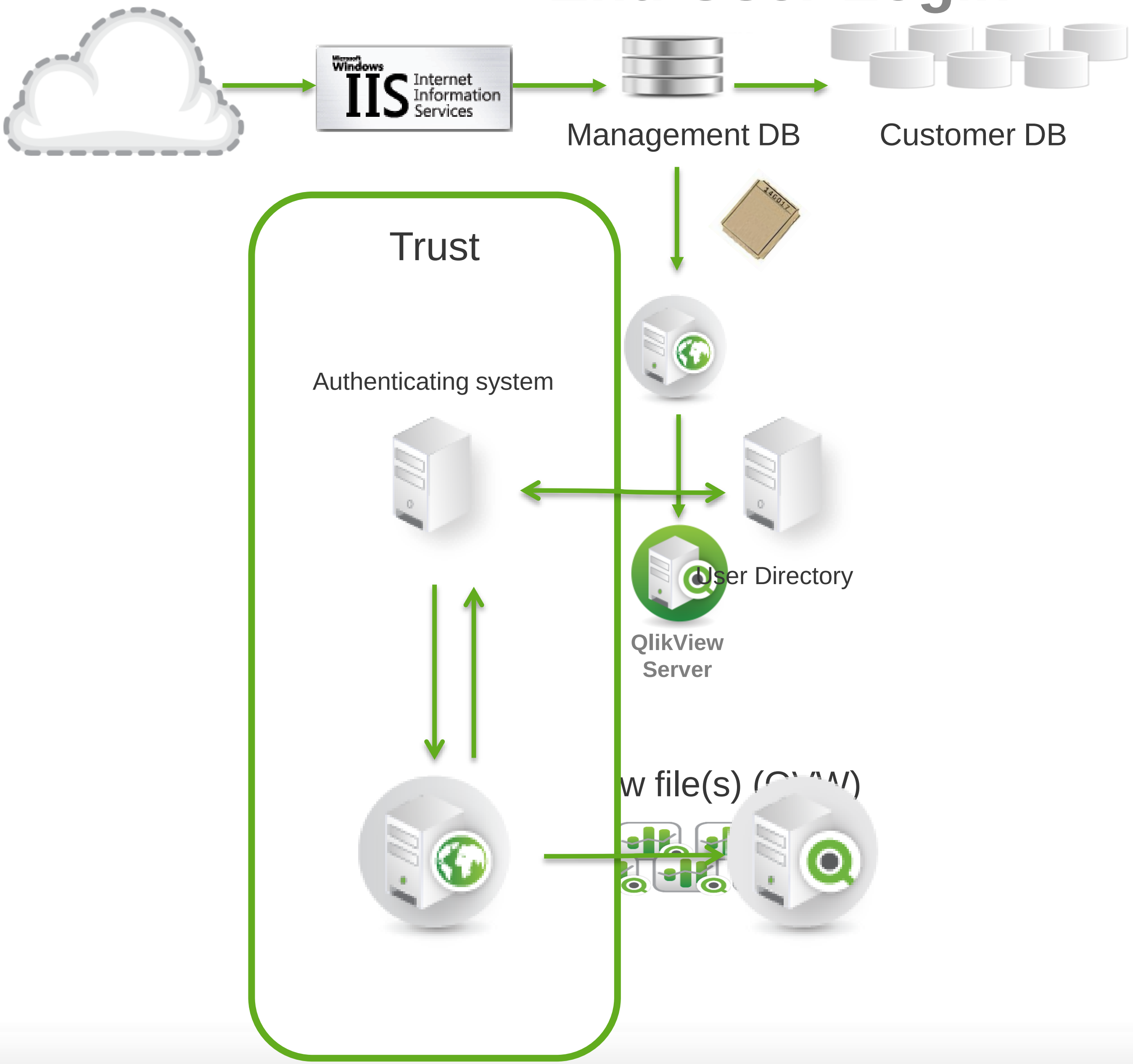
QlikView file (QVW)

QlikView
Publisher

Authorization via Web Tickets for Login



End User Login



Management DB

Customer DB

Trust

Authenticating system

User Directory

QlikView Server

w file(s) (QVW)

The Manual Process

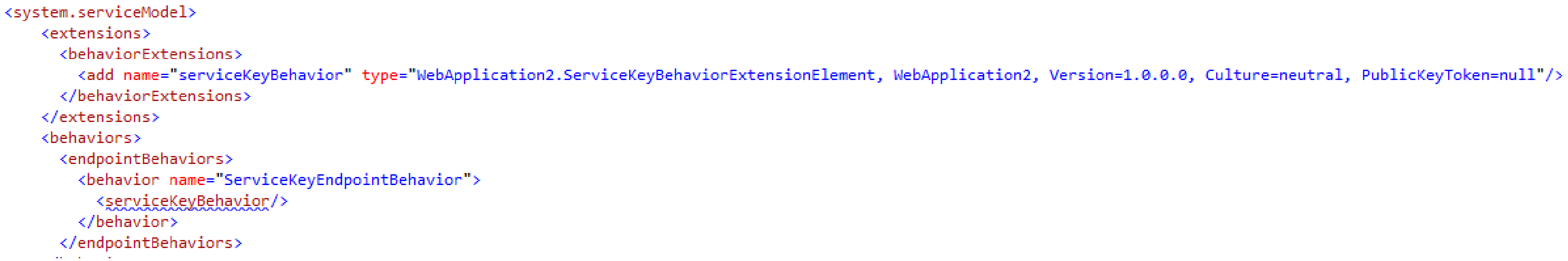
- Create a new instance of our OEM solution
- Copy a QlikView Doc for a new customer
- Make sure the load script point to the right database
- Create a new group in the user directory
- Create a new task and configure the settings

Building A QMS App

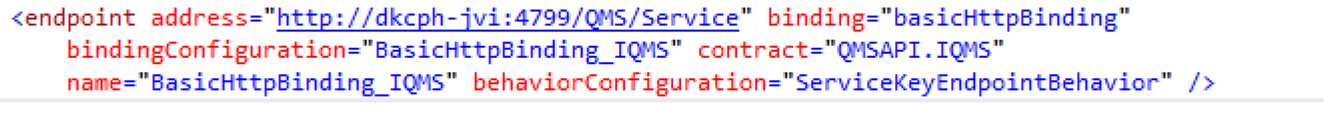
- What you need before you start:
  - Visual Studio
  - An administrator account that is a member of the QlikView Management API group (if not exist, then create group)
  - Access to the QlikView server through port 4799

- Setup Prerequisites:
  - A service reference connection to the QlikView Server (http://localhost:4799/QMS/service)
  - Change Web.Config

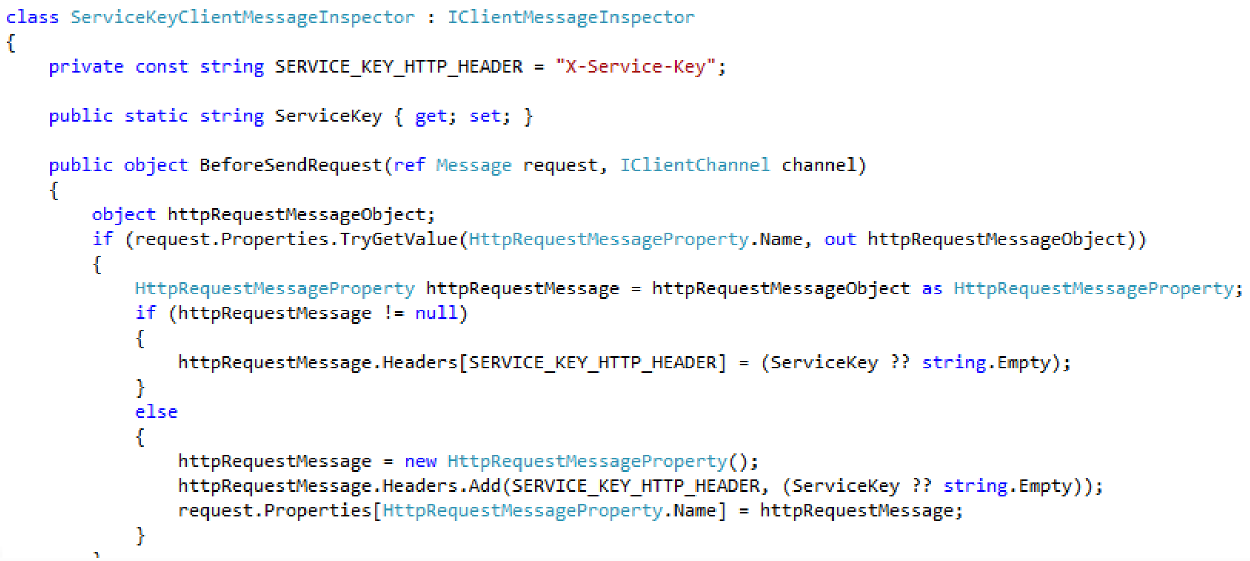
Change Web.Config and Add Class



```
<system.serviceModel>
    <extensions>
      <behaviorExtensions>
        <add name="serviceKeyBehavior" type="WebApplication2.ServiceKeyBehaviorExtensionElement, WebApplication2, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null"/>
      </behaviorExtensions>
    </extensions>
    <behaviors>
      <endpointBehaviors>
        <behavior name="ServiceKeyEndpointBehavior">
          <serviceKeyBehavior/>
        </behavior>
      </endpointBehaviors>
```

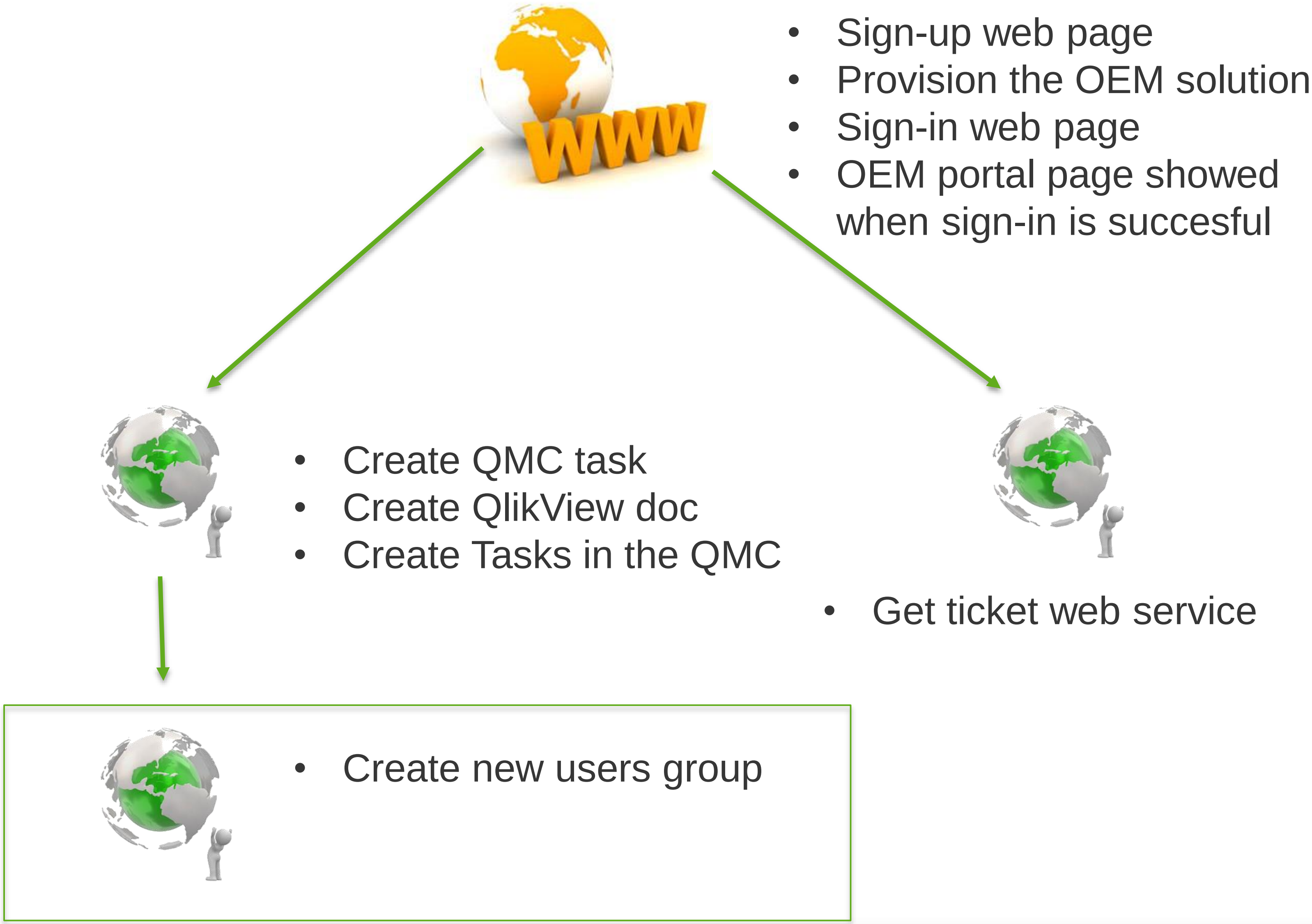


```
<endpoint address="http://dkcph-jvi:4799/QMS/Service" binding="basicHttpBinding"
    bindingConfiguration="BasicHttpBinding_IQMS" contract="QMSAPI.IQMS"
    name="BasicHttpBinding_IQMS" behaviorConfiguration="ServiceKeyEndpointBehavior" />
```



```
class ServiceKeyClientMessageInspector : IClientMessageInspector
{
    private const string SERVICE_KEY_HTTP_HEADER = "X-Service-Key";

    public static string ServiceKey { get; set; }

    public object BeforeSendRequest(ref Message request, IClientChannel channel)
    {
        object httpRequestMessageObject;
        if (request.Properties.TryGetValue(HttpRequestMessageProperty.Name, out httpRequestMessageObject))
        {
            HttpRequestMessageProperty httpRequestMessage = httpRequestMessageObject as HttpRequestMessageProperty;
            if (httpRequestMessage != null)
            {
                httpRequestMessage.Headers[SERVICE_KEY_HTTP_HEADER] = (ServiceKey ?? string.Empty);
            }
            else
            {
                httpRequestMessage = new HttpRequestMessageProperty();
                httpRequestMessage.Headers.Add(SERVICE_KEY_HTTP_HEADER, (ServiceKey ?? string.Empty));
                request.Properties[HttpRequestMessageProperty.Name] = httpRequestMessage;
            }
```

What We Need To Automate



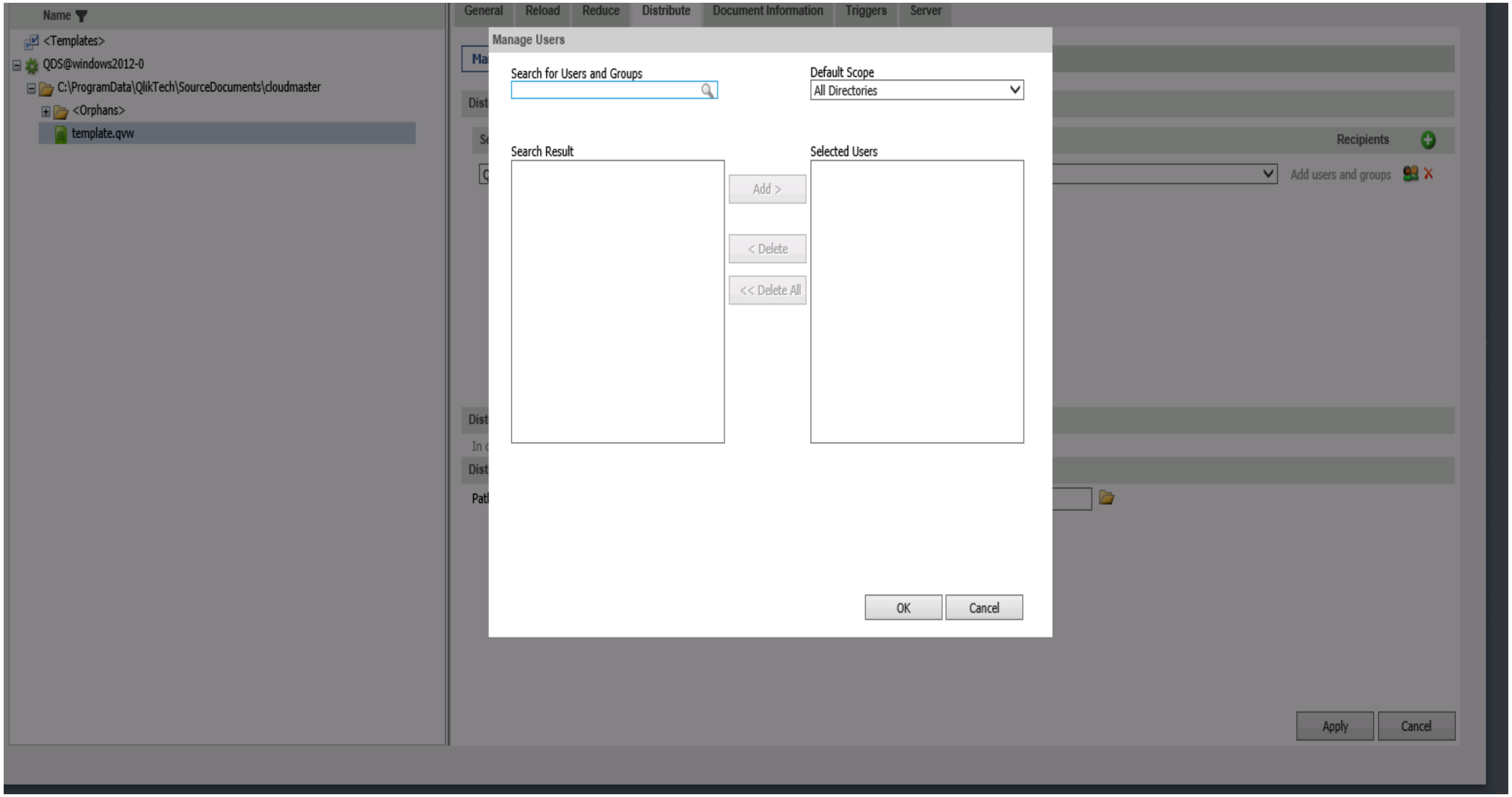
- Sign-up web page
- Provision the OEM solution
- Sign-in web page
- OEM portal page showed when sign-in is succesful

- Create QMC task
- Create QlikView doc
- Create Tasks in the QMC

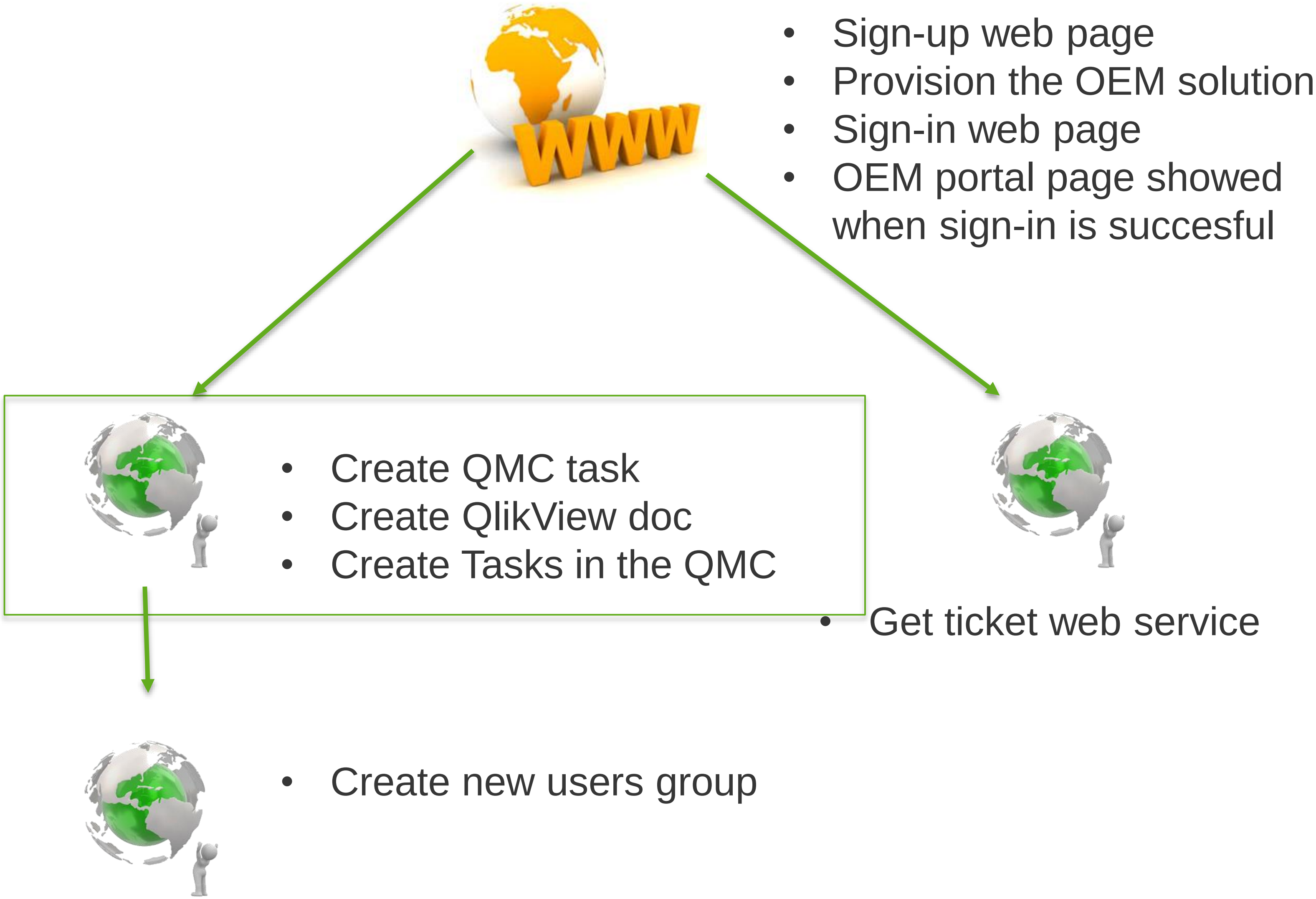
- Get ticket web service

- Create new users group

User or User Group Provisioning



What We Need To Automate That



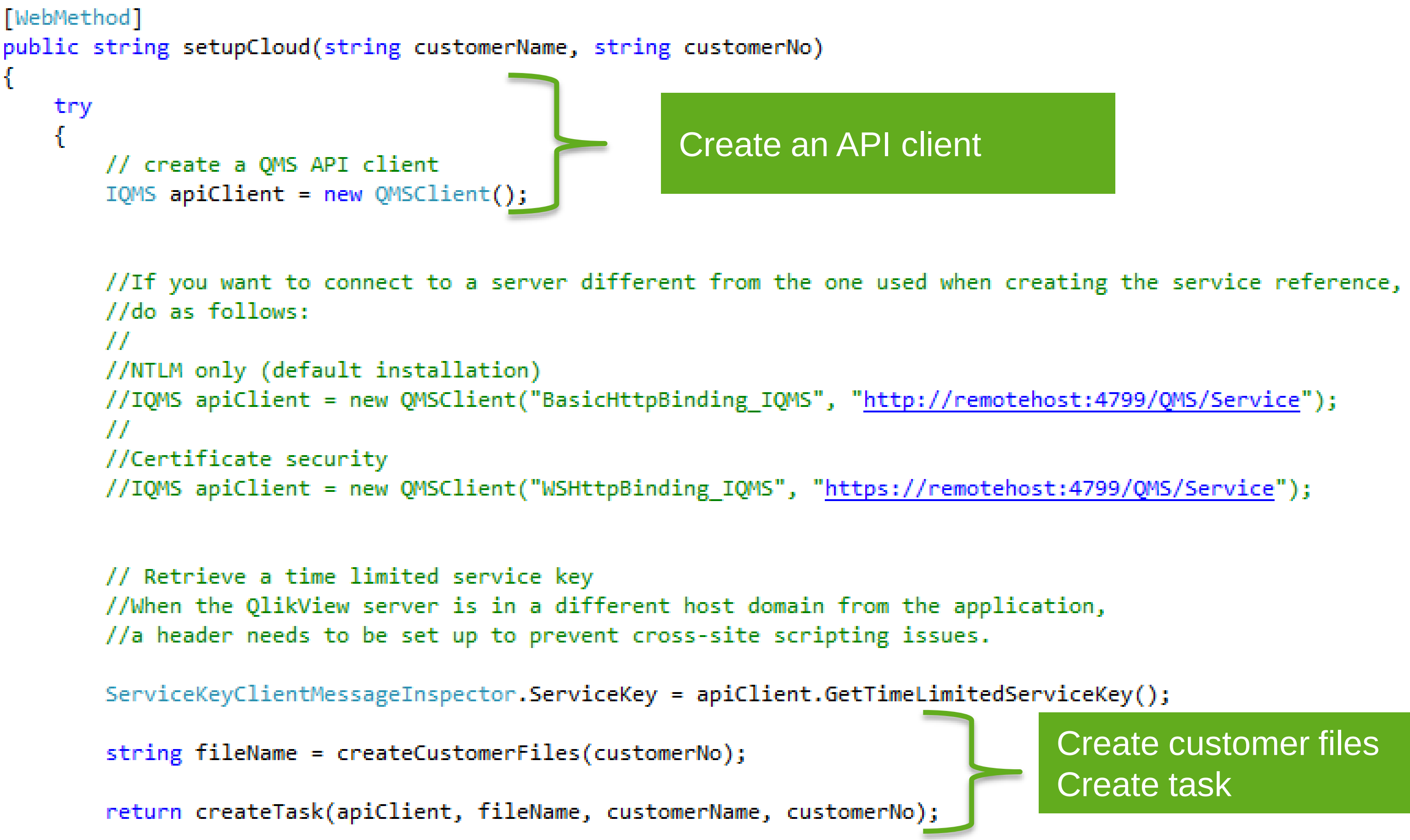
- Sign-up web page
- Provision the OEM solution
- Sign-in web page
- OEM portal page showed when sign-in is succesful

- Create QMC task
- Create QlikView doc
- Create Tasks in the QMC

- Get ticket web service

- Create new users group

Let’s Look At The Code



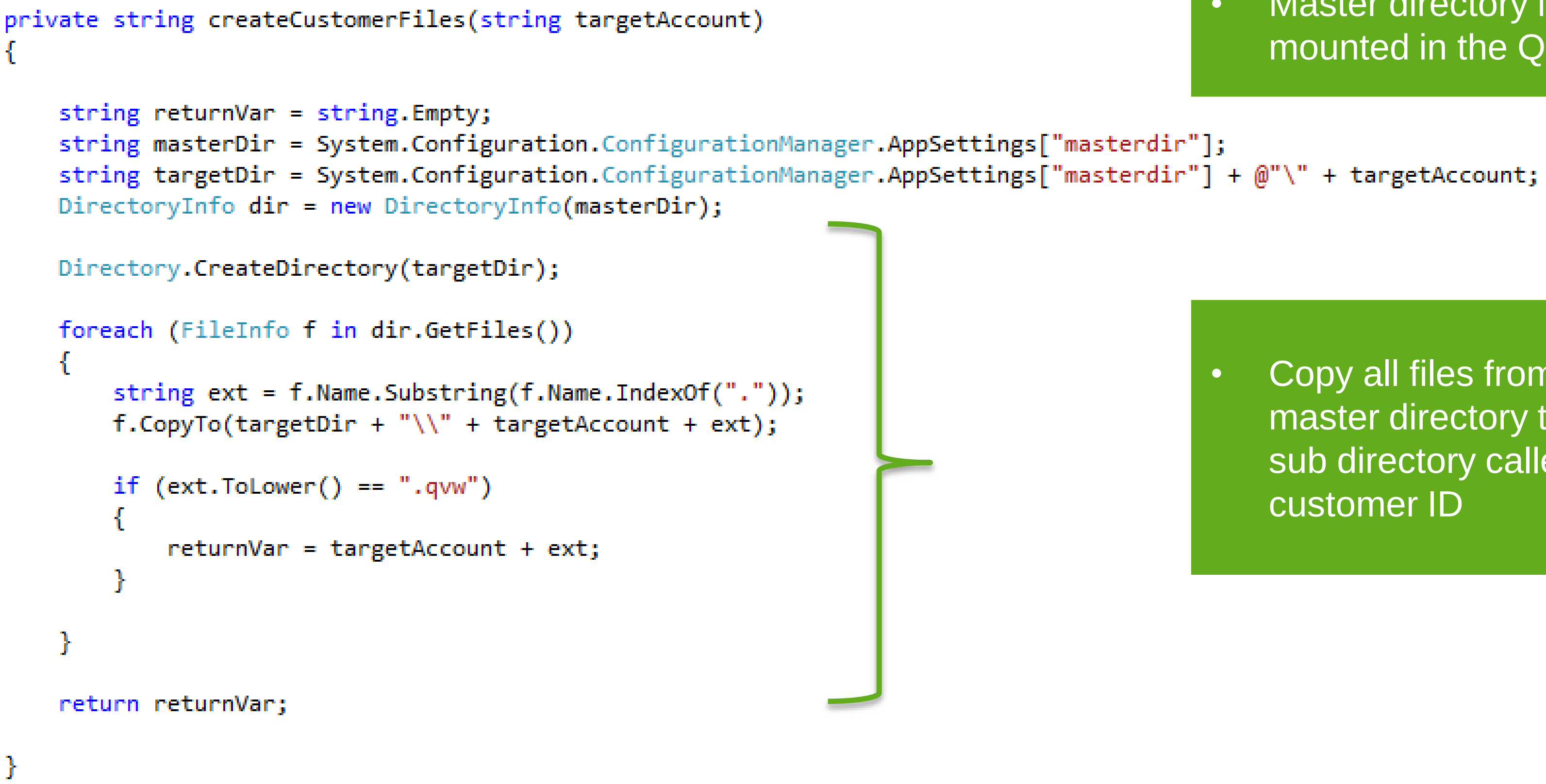
```
[WebMethod]
public string setupCloud(string customerName, string customerNo)
{
    try
    {
        // create a QMS API client
        IQMS apiClient = new QMSClient();


        //If you want to connect to a server different from the one used when creating the service reference,
        //do as follows:
        //
        //NTLM only (default installation)
        //IQMS apiClient = new QMSClient("BasicHttpBinding_IQMS", "http://remotehost:4799/QMS/Service");
        //
        //Certificate security
        //IQMS apiClient = new QMSClient("WSHttpBinding_IQMS", "https://remotehost:4799/QMS/Service");


        // Retrieve a time limited service key
        //When the QlikView server is in a different host domain from the application,
        //a header needs to be set up to prevent cross-site scripting issues.

        ServiceKeyClientMessageInspector.ServiceKey = apiClient.GetTimeLimitedServiceKey();

        string fileName = createCustomerFiles(customerNo);

        return createTask(apiClient, fileName, customerName, customerNo);
```

Create an API client

Create customer files
Create task

Let’s Look At The Code

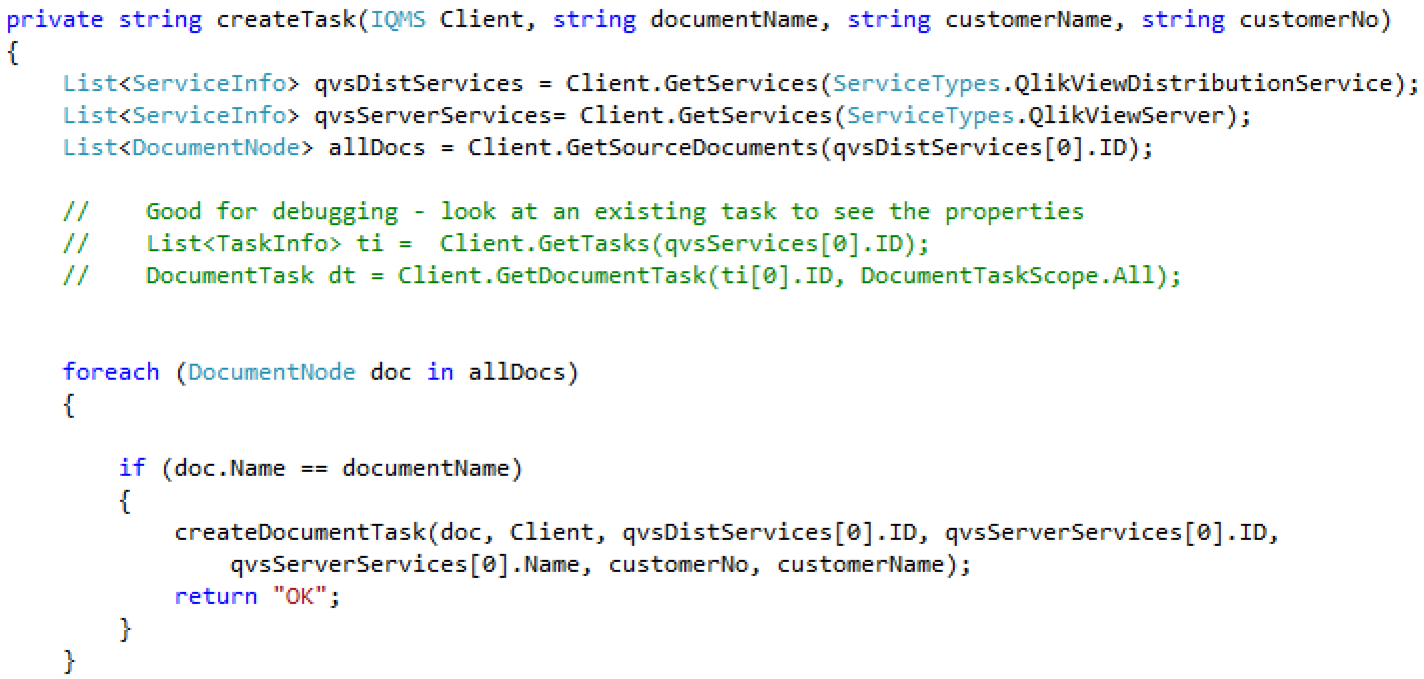
- Create a new directory under the master directory
- Master directory is mounted in the QMC



```
private string createCustomerFiles(string targetAccount)
{

    string returnVar = string.Empty;
    string masterDir = System.Configuration.ConfigurationManager.AppSettings["masterdir"];
    string targetDir = System.Configuration.ConfigurationManager.AppSettings["masterdir"] + @"\" + targetAccount;
    DirectoryInfo dir = new DirectoryInfo(masterDir);

    Directory.CreateDirectory(targetDir);

    foreach (FileInfo f in dir.GetFiles())
    {
        string ext = f.Name.Substring(f.Name.IndexOf("."));
        f.CopyTo(targetDir + "\\" + targetAccount + ext);

        if (ext.ToLower() == ".qvw")
        {
            returnVar = targetAccount + ext;
        }

    }

    return returnVar;

}
```

- Copy all files from the master directory to a sub directory called the customer ID

Let’s Look At The Code

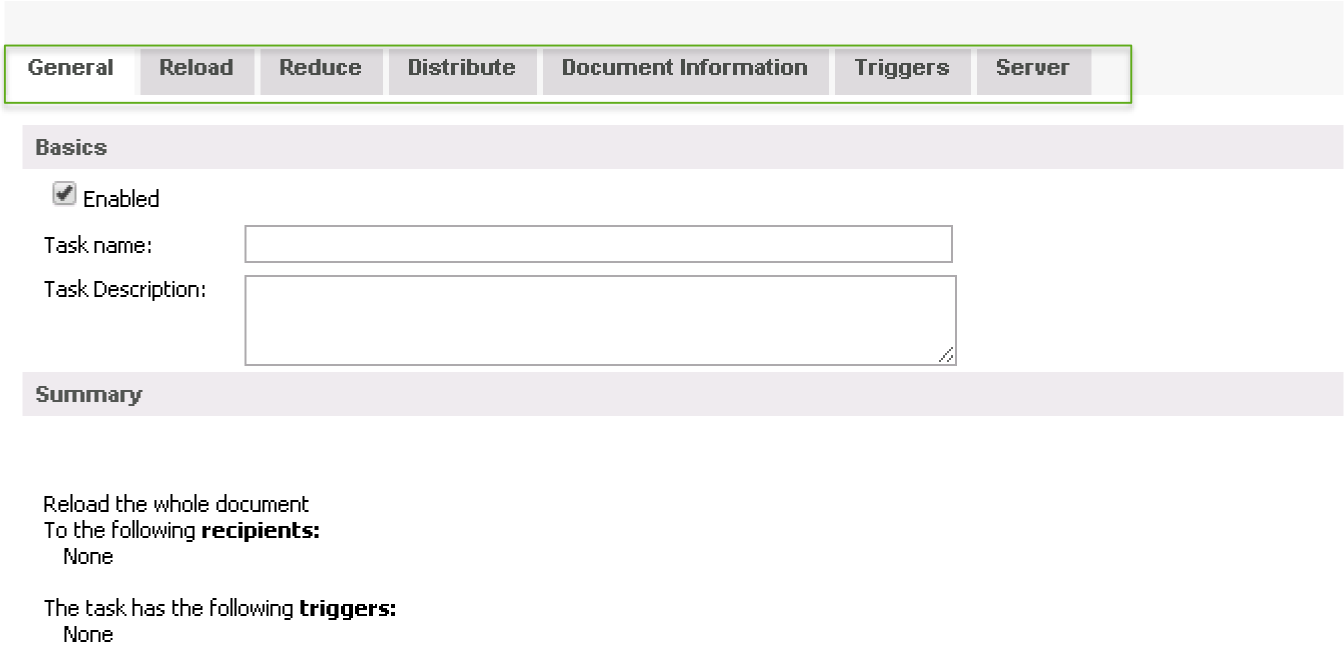


```
private string createTask(IQMS Client, string documentName, string customerName, string customerNo)
{
    List<ServiceInfo> qvsDistServices = Client.GetServices(ServiceTypes.QlikViewDistributionService);
    List<ServiceInfo> qvsServerServices= Client.GetServices(ServiceTypes.QlikViewServer);
    List<DocumentNode> allDocs = Client.GetSourceDocuments(qvsDistServices[0].ID);

    //    Good for debugging - look at an existing task to see the properties
    //    List<TaskInfo> ti =  Client.GetTasks(qvsServices[0].ID);
    //    DocumentTask dt = Client.GetDocumentTask(ti[0].ID, DocumentTaskScope.All);


    foreach (DocumentNode doc in allDocs)
    {

        if (doc.Name == documentName)
        {
            createDocumentTask(doc, Client, qvsDistServices[0].ID, qvsServerServices[0].ID,
                qvsServerServices[0].Name, customerNo, customerName);
            return "OK";
        }
    }
```

Get document list from the publisher

Find the new created document by name and create task

Let’s Look At The Code – Think About The Tabs



Basics

☑ Enabled

Task name:

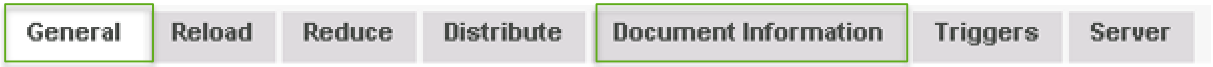
Task Description:

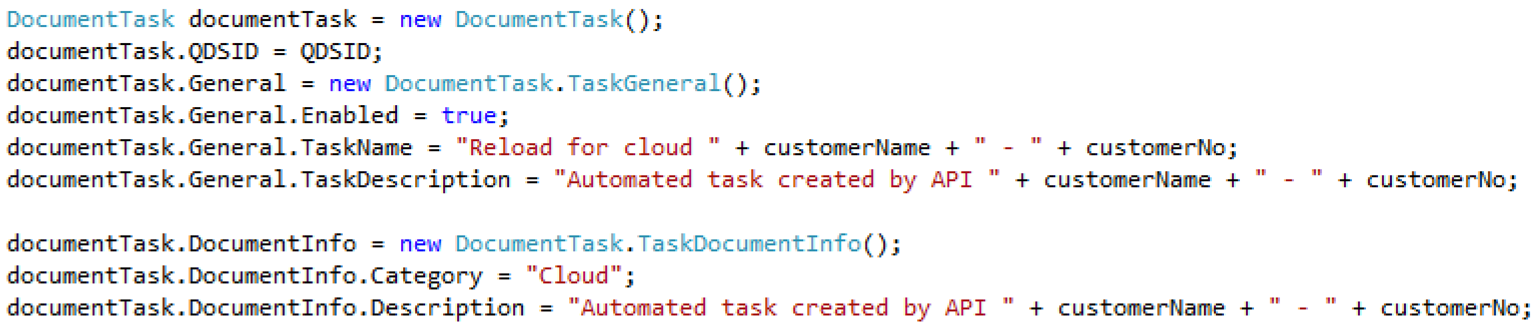
Summary

Reload the whole document
To the following **recipients:**
None

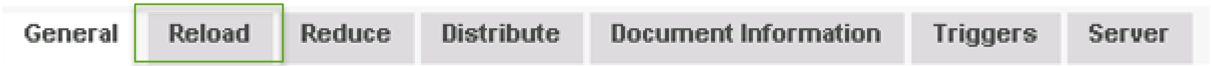
The task has the following **triggers:**
None

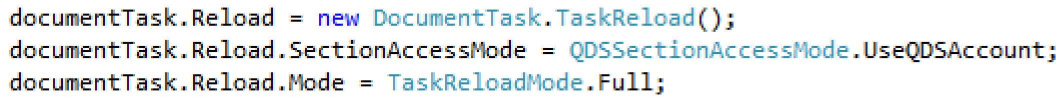
Let’s Look At The Code – Think About The Tabs



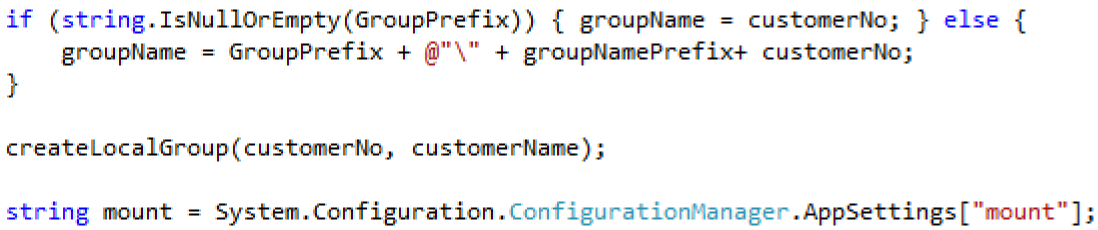


```
DocumentTask documentTask = new DocumentTask();
documentTask.QDSID = QDSID;
documentTask.General = new DocumentTask.TaskGeneral();
documentTask.General.Enabled = true;
documentTask.General.TaskName = "Reload for cloud " + customerName + " - " + customerNo;
documentTask.General.TaskDescription = "Automated task created by API " + customerName + " - " + customerNo;

documentTask.DocumentInfo = new DocumentTask.TaskDocumentInfo();
documentTask.DocumentInfo.Category = "Cloud";
documentTask.DocumentInfo.Description = "Automated task created by API " + customerName + " - " + customerNo;
```





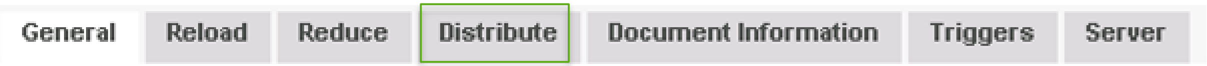
```
documentTask.Reload = new DocumentTask.TaskReload();
documentTask.Reload.SectionAccessMode = QDSSectionAccessMode.UseQDSAccount;
documentTask.Reload.Mode = TaskReloadMode.Full;
```

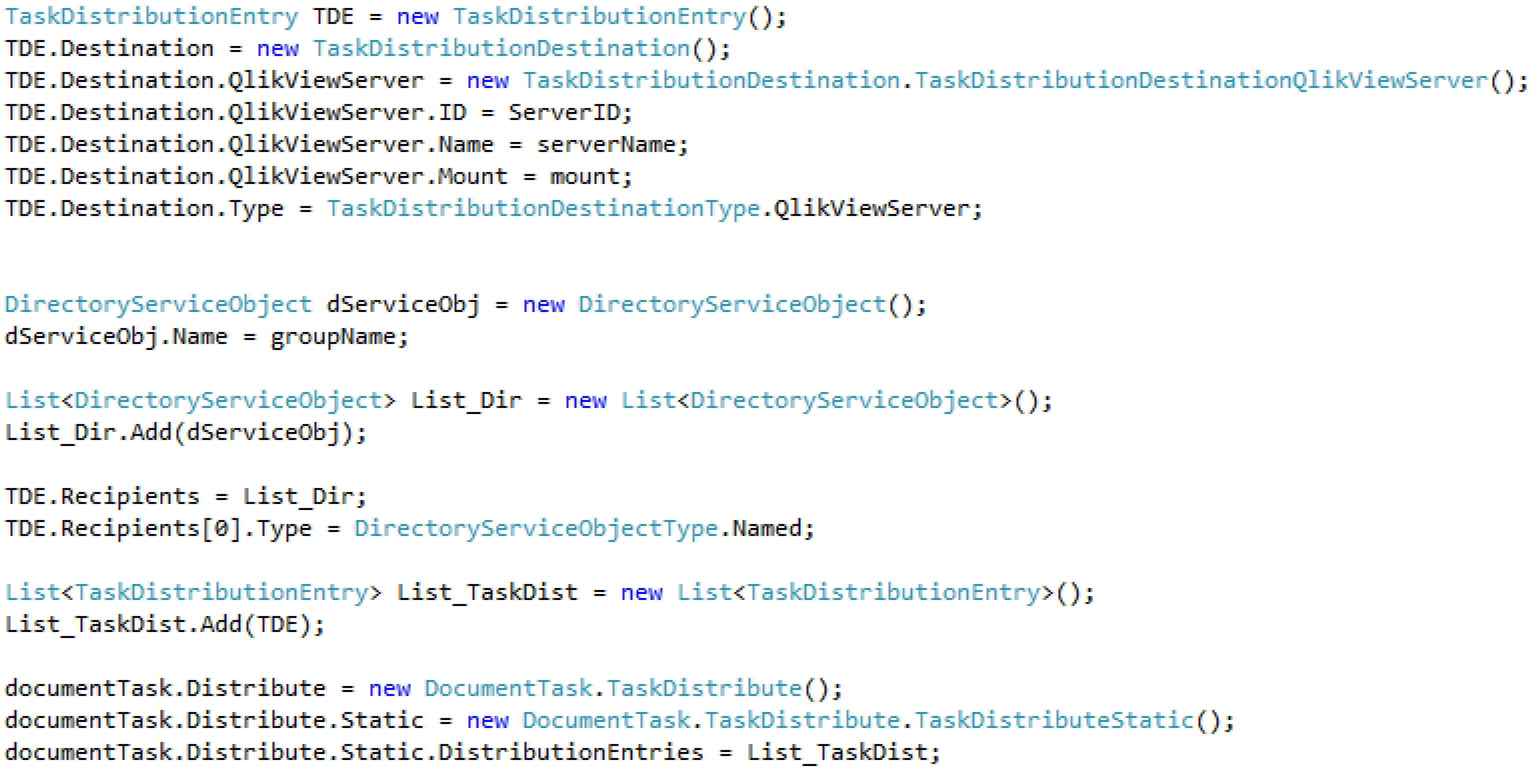


```
if (string.IsNullOrEmpty(GroupPrefix)) { groupName = customerNo; } else {
    groupName = GroupPrefix + @"\" + groupNamePrefix+ customerNo;
}

createLocalGroup(customerNo, customerName);

string mount = System.Configuration.ConfigurationManager.AppSettings["mount"];
```

- Create the group in the user directory
- Get the mount name related to the master directory

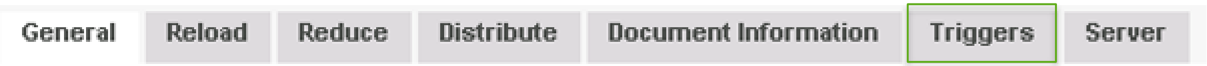
Let’s Look At The Code – Think About The Tabs

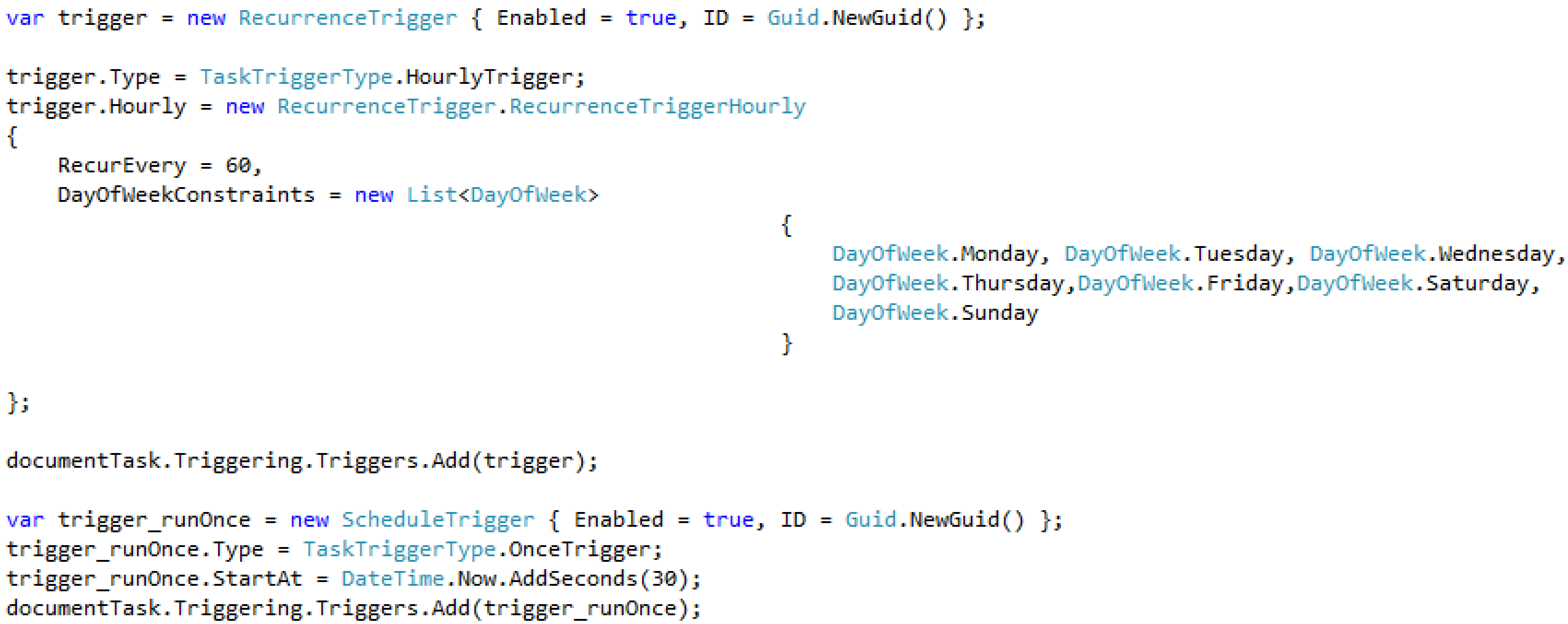




```
TaskDistributionEntry TDE = new TaskDistributionEntry();
TDE.Destination = new TaskDistributionDestination();
TDE.Destination.QlikViewServer = new TaskDistributionDestination.TaskDistributionDestinationQlikViewServer();
TDE.Destination.QlikViewServer.ID = ServerID;
TDE.Destination.QlikViewServer.Name = serverName;
TDE.Destination.QlikViewServer.Mount = mount;
TDE.Destination.Type = TaskDistributionDestinationType.QlikViewServer;


DirectoryServiceObject dServiceObj = new DirectoryServiceObject();
dServiceObj.Name = groupName;

List<DirectoryServiceObject> List_Dir = new List<DirectoryServiceObject>();
List_Dir.Add(dServiceObj);

TDE.Recipients = List_Dir;
TDE.Recipients[0].Type = DirectoryServiceObjectType.Named;

List<TaskDistributionEntry> List_TaskDist = new List<TaskDistributionEntry>();
List_TaskDist.Add(TDE);

documentTask.Distribute = new DocumentTask.TaskDistribute();
documentTask.Distribute.Static = new DocumentTask.TaskDistribute.TaskDistributeStatic();
documentTask.Distribute.Static.DistributionEntries = List_TaskDist;
```

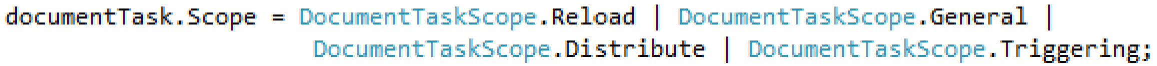
Let’s Look At The Code – Think About The Tabs



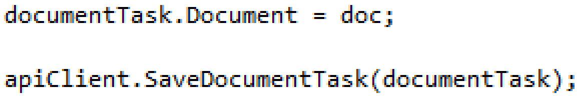


```
var trigger = new RecurrenceTrigger { Enabled = true, ID = Guid.NewGuid() };

trigger.Type = TaskTriggerType.HourlyTrigger;
trigger.Hourly = new RecurrenceTrigger.RecurrenceTriggerHourly
{
    RecurEvery = 60,
    DayOfWeekConstraints = new List<DayOfWeek>
                                                            {
                                                                DayOfWeek.Monday, DayOfWeek.Tuesday, DayOfWeek.Wednesday,
                                                                DayOfWeek.Thursday,DayOfWeek.Friday,DayOfWeek.Saturday,
                                                                DayOfWeek.Sunday
                                                            }

};

documentTask.Triggering.Triggers.Add(trigger);

var trigger_runOnce = new ScheduleTrigger { Enabled = true, ID = Guid.NewGuid() };
trigger_runOnce.Type = TaskTriggerType.OnceTrigger;
trigger_runOnce.StartAt = DateTime.Now.AddSeconds(30);
documentTask.Triggering.Triggers.Add(trigger_runOnce);
```

Let’s Look At The Code – Save The Work



```
documentTask.Scope = DocumentTaskScope.Reload | DocumentTaskScope.General |
                      DocumentTaskScope.Distribute | DocumentTaskScope.Triggering;
```

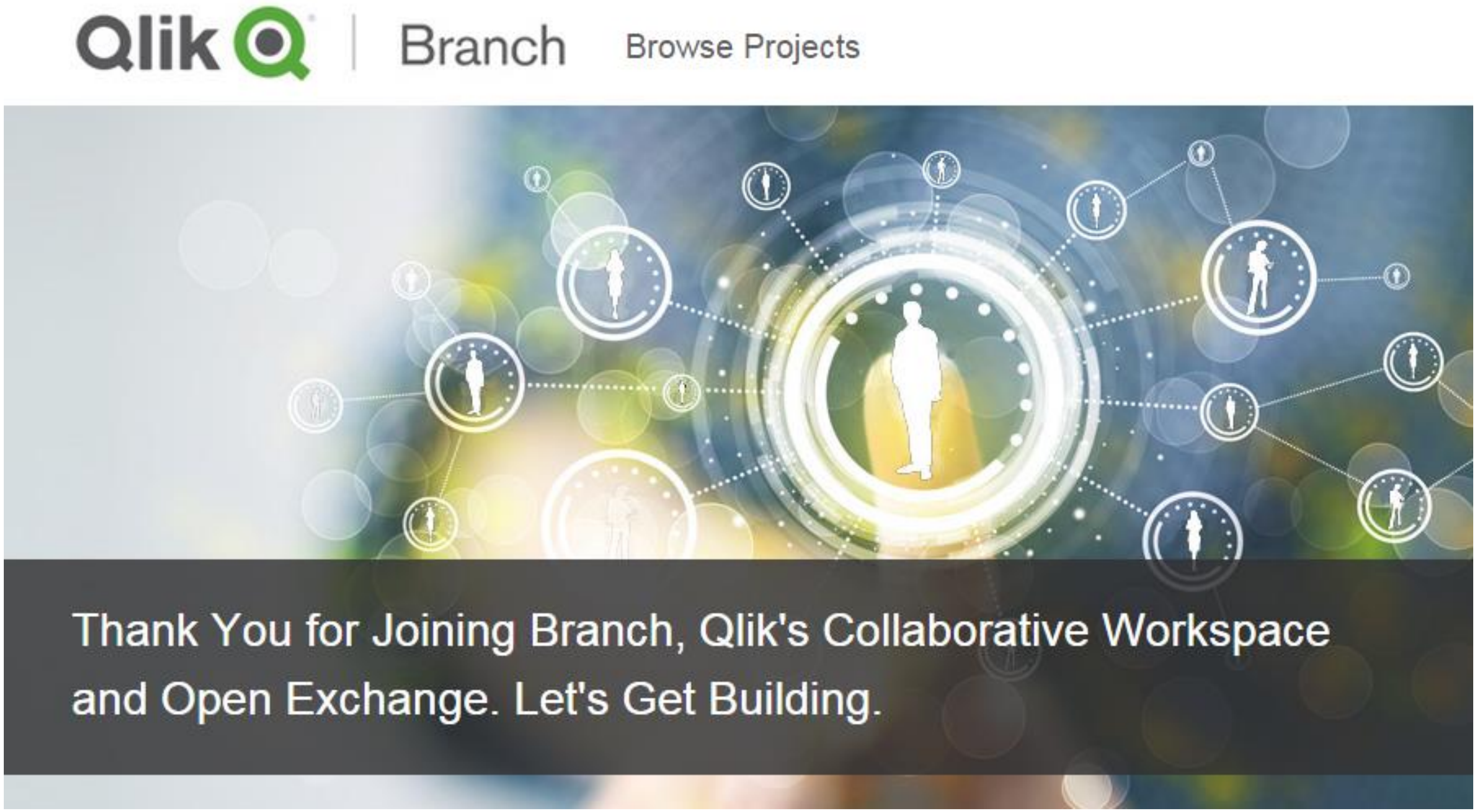


```
documentTask.Document = doc;

apiClient.SaveDocumentTask(documentTask);
```

Qlik Qonnections Global Partner Summit 2015

Let’s Look At The Result

Remember to Share



Qlik | Branch Browse Projects

Thank You for Joining Branch, Qlik's Collaborative Workspace and Open Exchange. Let's Get Building.

http://branch.qlik.com/

Qlik | Qonnections

Global Partner Summit 2015

Question and Answer

Feedback survey

- Please complete the track session survey via the mobile app

- Access the track session survey through the mobile app
- Enter track session code T55
- Provide your feedback

Qlik | Qonnections
Global Partner Summit 2015

10th Annual

Thank You