



**Q-Pulse Web Service APIs
Development Scenarios**

Contents

Introduction.....	3
Scenario 1 – Creating a Non-conformance.....	4
Scenario 2 – Raising an Occurrence	5
Scenario 3 – Adding an attachment to an existing Audit	6
Scenario 4 – Searching for a Document	7
Scenario 5 - Searching on Occurrence Dynamic Fields.....	8

Introduction

The purpose of this document is to model several scenarios, showing which calls to the Q-Pulse Web Service would be required to achieve certain functionality. Code snippets have been used to illustrate the calls required.

Scenario 1 – Creating a Non-conformance

Description: Create a new Non-conformance in Q-Pulse via the Web Service

Steps

1. Get a database name:

```
String databaseName = CoreSvc.AvailableConnections()[0]
```

2. Using a database name, get an authentication token by authenticating a user:

```
String token = CoreSvc.Authenticate(userName, password, databaseName)
```

3. Using the authentication token, create a new NonConformance:

```
NonConformance nc = NonConformanceSvc.CreateNewNonConformance(token,  
    NCToSend)
```

4. Check that the creation succeeded:

```
Bool success = nc.ID > 0 && nc.BrokenRules.Count == 0
```

Scenario 2 – Raising an Occurrence

Description: Raise an Occurrence for a particular user.

Steps

1. Get a database name:

```
String databaseName = CoreSvc.AvailableConnections()[0]
```

2. Using a database name, get an authentication token by authenticating a user:

```
String token = CoreSvc.Authenticate(userName, password, databaseName)
```

3. Get the Occurrence Report Types available in Q-Pulse and choose one:

```
String reportTypeName =  
    OccurrenceSvc.ListAvailableReportTypes(databaseName)[0]
```

4. Get the DataFields available for the Report Type by passing in the database name and the Report Type, and choose one to populate with data:

```
DataField dataField =  
    OccurrenceSvc.GetDataFieldsForReportType(reportTypeName,  
        databaseName)[0]
```

5. Create a new Occurrence and send it to the Web Service:

```
Occurrence occToSend = new Occurrence()  
occToSend.ReportTypeName = reportTypeName  
dataField.Value = "some value"  
occToSend.DataFields = dataField  
Occurrence occ = Occurrence.CreateNewOccurrence(token, occToSend )
```

6. Check that the creation succeeded:

```
bool success = occ.ID > 0 && occ.BrokenRules.Count == 0
```

Scenario 3 – Adding an attachment to an existing Audit

Description: Find an Audit where the Auditor is “Joe Bloggs” and add an attachment to it.

Steps

1. Get a database name:

```
String databaseName = CoreSvc.AvailableConnections()[0]
```

2. Using a database name, get an authentication token by authenticating a user:

```
String token = CoreSvc.Authenticate(userName, password, databaseName)
```

3. Using the authentication token and search criteria, get the ID of an Audit:

```
int auditID = Audit.GetAuditIDs(token, new AuditQuery{ AuditorName =  
    "Bloggs, Joe" })[0]
```

4. Upload a file to the Attachments endpoint by manually creating a request message:

```
WebClient client = new WebClient()  
client.Headers.Add("Operation", "Upload")  
client.Headers.Add("Token", token)  
client.Headers.Add("ObjectName", "Audit")  
client.Headers.Add("ObjectID", auditID)  
byte[] response = WebClient.UploadFile(locationOfAttachmentSvc,  
    locationOfFile)
```

Scenario 4 – Searching for a Document

Description: Search for a Document in Q-Pulse via the Web Service

Steps

1. Get a database name:

```
String databaseName = CoreSvc.AvailableConnections()[0]
```

2. Using a database name, get an authentication token by authenticating a user:

```
String token = CoreSvc.Authenticate(userName, password, databaseName)
```

3. Get a list of current Document Types in Q-Pulse and choose one:

```
int docTypeID = CoreSvc.GetManagedList(token,
ManagedListType.DocumentType)[0].ID
```

4. Create a DocumentQuery and send it to the Web Service to get a collection of matching documents:

```
Document query = new DocumentQuery()
{
    DocumentStatus = DocumentStatusType.Active
    Keywords = "Joe"
}
Document [ ] documents = DocumentSvc.GetDocuments(token, query)
```

Scenario 5 - Searching on Occurrence Dynamic Fields

Steps

1. Get a database name:

```
String databaseName = CoreSvc.AvailableConnections()[0]
```

2. Using a database name, get an authentication token by authenticating a user:

```
String token = CoreSvc.Authenticate(userName, password, databaseName)
```

3. Get the Occurrence Report Types available in Q-Pulse and choose one:

```
String reportTypeName =  
OccurrenceSvc.ListAvailableReportTypes(databaseName)[0]
```

4. Get the DataFields available for the Report Type by passing in the database name and the Report Type, and choose one to search on:

```
DataField dataField =  
OccurrenceSvc.GetDataFieldsForReportType(reportTypeName,  
databaseName)[0]
```

5. Build up an Occurrence Query and send it to the Web Service to retrieve a collection of matching Occurrences:

```
DynamicFieldCondition condition = new DynamicFieldCondition()  
{  
    FieldName = dataField.Name  
    Operator = DynamicFieldOperator.Containing  
    Values = new String [] { "spillage" }  
}  
OccurrenceQuery query = new OccurrenceQuery()  
{  
    ReportTypeName = reportTypeName,  
    DynamicFieldConditions = new DynamicFieldCondition[]  
    {condition}  
}  
Occurrence [ ] occurrences = Occurrence.GetOccurrences(token, query)
```



Gael Ltd.

Orion House,
S. E. Technology Park,
East Kilbride,
Scotland. G75 0RD

t: +44 1355 593400

f: +44 1355 579191

e: info@gaelquality.com

w: www.gaelquality.com

Q-Pulse is a registered trademark of Gael Products Ltd. All rights reserved worldwide.
Copyright © 2010 Gael Products Ltd. Gael Quality is a trading division of Gael Ltd.