



Q-Pulse Web Service APIs Technical Reference Guide

API Version: 02.000.590.081

Contents

Introduction	6
Version History	6
Core Service	7
GetQPulseWebServiceVersion	7
AvailableConnections	8
GetServerInfo	9
Authenticate, AuthenticateWindowsID and AuthenticateWindowsIDAndPassword	10
GetManagedList	14
SearchManagedList	16
GetServerAuthenticationType	18
HasServerLicenseExpired	18
GetLoggedInPersonDetails	19
SendEmail	20
Occurrence Service	22
ListAvailableReportTypes	23
GetDataFieldsForReportType	24
GetDataFields	26
GetReferenceList	29
CreateNewOccurrence	31
GetOccurrence	34
GetOccurrences	36
GetOccurrenceIDs	40
GetOccurrenceGotoUrl	42
RejectOccurrence	43

Incident Service	46
GetIncident.....	46
GetIncidentIDs.....	52
Attachment Service	55
PostAttachment.....	57
GetAttachment	58
Audit Service.....	61
GetAudit	61
GetAudits	66
GetAuditIDs.....	67
CreateNewAudit	69
Document Service	71
GetDocument	71
GetDocumentByNumber	72
GetDocumentByRevisionReference.....	73
GetDocumentByRevisionID.....	75
GetDocuments	76
GetDocumentIDs.....	77
GetDocumentsByIDs.....	79
GetDocumentSummaries	80
GetDocumentSummariesByIDs	83
GetDocumentAttachment	84
GetDocumentAndFolders.....	86
GetDocumentFolders	88
GetDocumentFoldersByCategory.....	89

GetDocumentFolderByID	90
GetDocumentsByFolder	92
GetDocumentIDsByFolder	93
GetDocumentSummariesByFolder	94
GetWorkloads.....	95
GetDocumentsByWorkload	96
GetDocumentIDsByWorkload	98
GetDocumentSummariesByWorkload	99
CreateNewDocument	100
AddNote	102
AcknowledgeDocument.....	103
ReviseDocumentToActive	104
AddUrlAttachmentItem	106
DistributeActiveDocument	107
DeactivateDocument	108
NonConformance Service.....	110
GetNonConformance	110
GetNonConformances.....	117
GetNonConformanceIDs	119
CreateNewNonConformance	121
Exceptions	124
Application Level Exceptions.....	124
Broken Rules.....	126
Unable to authenticate a User	127

Debugging Non-Application Exceptions	128
Communication Problems	128
XML problems	129
Other Considerations	129
Other Debugging Tools	129

Introduction

This document contains descriptions and example messages sent over HTTP between a client and the Q-Pulse Web Service endpoints.

They are examples of request and response messages for the purposes of understanding the working nature of each of the service calls.

The “live” documentation located at `http://<servicepath>/Docs/` contains the **WSDL**, **XSD** and generated documentation which define the formal contractual requirements for making requests of the services.

The Web Service Description Language (WSDL) can be obtained for each service endpoint by appending ‘?wsdl’ to the endpoint address e.g. `http:// <servicepath>/Services/Core.svc?wsdl`

The XML throughout this document was *generated* by WCF (both client and server side) and is for illustration purposes only.

It should be highlighted that the Q-Pulse Web Service uses two forms of error handling. The first is to address application level errors where a well-formed `Service FaultException` is returned when something undesirable happens in the Web Service call lifetime. The second is the concept of Broken Rules where a collection of `BrokenRule` entities are returned as part of a business object detailing the reasons why the saving of a business object has failed. Further detail is available in the section [Exceptions](#).

Version History

02.000.580.014 – Breaking Changes

To make the use of DataFields of Reference List type consistent throughout the Occurrence API a breaking change has been implemented which changes existing behaviour in the Occurrence service. For methods which use the DataField contract the service will expect the ‘Value’ field to reference the list items’s ID value. To make use of a list item’s display value it must be supplied using the ‘ListItemDisplayValue’ field. Further more a new DynamicFieldOperator called ‘IdEqualTo’ has been added for use within the DynamicFieldCondition contract. For all existing operators the service will treat the ‘Values’ field as a list item’s display value, however, if the new IdEqualTo operator is used the ‘Values’ field will be treated as a list items ID value.

As a result of this breaking change the service namespace has changed to

`“http://www.qpulse.com/QPulseWebServices/v1.2/”`

Core Service

Description: This service provides access to common cross-module operations, including the ability to perform authentication and retrieve (non-user-created) managed lists. The Q-Pulse Web Service uses the same user authentication model as the Q-Pulse application, so it is necessary to authenticate a user before performing most of the Web Service operations. Upon successful authentication, a token is returned to the caller which should be subsequently passed into the other Web Service operations that require it.

Endpoint: `http://<servicepath>/Services/Core.svc`

GetQPulseWebServiceVersion

Description: Provides the version number of this particular release of the QPulse Web Service product in the format of aa.bbb.cc.dd. This is not to be confused with the web service contract version.

Endpoint: `http://<servicepath>/Services/Core.svc/GetQPulseWebServiceVersion`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetQPulseWebServiceVersion/>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetQPulseWebServiceVersionResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetQPulseWebServiceVersionResult> Q-PULSE WEB SERVICES VERSION
    </GetQPulseWebServiceVersionResult>

    </GetQPulseWebServiceVersionResponse>

  </s:Body>

</s:Envelope>
```

```

</s:Body>
</s:Envelope>

```

AvailableConnections

Description: Provides a list of available Q-Pulse Databases. This is used to determine which databases are available for the purposes of specifying which database to use in some of the other Web Service operations e.g. Authenticate.

Endpoint: `http://<servicepath>/Services/Core.svc/AvailableConnections`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:AvailableConnections/>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <AvailableConnectionsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <AvailableConnectionsResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <a:string>Q-PULSE DATABASE NAME</a:string>

      </AvailableConnectionsResult>

    </AvailableConnectionsResponse>

  </s:Body>

```

```
</s:Envelope>
```

GetServerInfo

Description: Provides information related to Q-Pulse server including information about the licence, the version of web services and a list of available Q-Pulse Databases which can be logged into.

Endpoint: `http://<servicepath>/Services/Core.svc/GetServerInfo`

Availability: Q-Pulse Web Services API version 02.000.580.010 and above

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetServerInfo/>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetServerInfoResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetServerInfoResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <AvailableDatabases>

          <DatabaseInfo>

            <AuthenticationType>QPulseAuthentication</AuthenticationType>

            <Name> Q-PULSE DATABASE NAME</Name>

          </DatabaseInfo>

        </AvailableDatabases>

      </GetServerInfoResult>

    </GetServerInfoResponse>

  </s:Body>

</s:Envelope>
```

```

        </AvailableDatabases>

        <LicenceGNumber>LICENCE GNUMBER</LicenceGNumber>

        <LicenceeName>LICENCEE NAME</LicenceeName>

        <WebServicesVersion>Q-PULSE WEB SERVICES
VERSION</WebServicesVersion>

    </GetServerInfoResult>

</GetServerInfoResponse>

</s:Body>
</s:Envelope>

```

Authenticate, AuthenticateWindowsID and AuthenticateWindowsIDAndPassword

Description: Returns an authentication token that can be used to access services requiring authentication. The operations require a user who has permission to log in. Authenticate requires a username, password and database name. The username and password should correspond with a user in the Q-Pulse database specified. AuthenticateWindowsID requires a Windows user name and a database name. It will only be successful if Single Sign-On is enabled in Q-Pulse and the user is allowed to login using Single Sign-On. AuthenticateWindowsID essentially allows a client application to be developed which uses the user's Windows identity in order to avoid asking them to enter a username and password. If a Windows application is being developed, this information can be obtained from the Thread Principal and Identity (see <http://msdn.microsoft.com/en-us/library/t6547wf1.aspx>). The same approach can be used if developing an ASP.NET application (Windows Authentication must be enabled in the Web.config, see <http://msdn.microsoft.com/en-us/library/aa480475.aspx> and the impersonation section of <http://msdn.microsoft.com/en-us/library/ms998358.aspx>). AuthenticateWindowsIDAndPassword requires a Windows user name and password and a database name. If a Windows application is being developed, the user would need to be prompted to enter their Windows user name and password. It will only be successful if Windows Authentication prompt for credentials is enabled in Q-Pulse.

Endpoints:

- `http://<servicepath>/Services/Core.svc/Authenticate`
- `http://<servicepath>/Services/Core.svc/AuthenticateWindowsID`
- `http://<servicepath>/Services/Core.svc/AuthenticateWindowsIDAndPassword`

IMPORTANT NOTE: As the XML messages involved in authentication contain sensitive information it is important to secure these messages. See the Section “*Security Considerations*” in the Developer Guide.

Sample Requests:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:Authenticate>

      <v1:userName>Q-Pulse User Name</v1:userName>

      <v1:password>Q-Pulse Password</v1:password>

      <v1:database>Q-Pulse Database</v1:database>

    </v1:Authenticate>

  </s:Body>

</s:Envelope>
```

...and for Windows ID Authentication...

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:AuthenticateWindowsID>

      <v1:windowsUserName>WINDOWS DOMAIN\Windows User
      Name</v1:windowsUserName>

      <v1:database>Q-PULSE DATABASE NAME</v1:database>

    </v1:AuthenticateWindowsID>

  </s:Body>

</s:Envelope>
```

...and for Windows ID and Password Authentication...

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:AuthenticateWindowsIDAndPassword>

      <v1:windowsUserName> WINDOWS_DOMAIN\Windows User
Name</v1:windowsUserName>

      <v1:password>WINDOWS PASSWORD</v1:password>

      <v1:database>Q-PULSE DATABASE NAME</v1:database>

    </v1:AuthenticateWindowsIDAndPassword>

  </s:Body>

</s:Envelope>

```

Sample Responses:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <AuthenticateResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <AuthenticateResult>AUTHENTICATION TOKEN</AuthenticateResult>

    </AuthenticateResponse>

  </s:Body>

</s:Envelope>

```

...and again for Windows ID Authentication...

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <AuthenticateWindowsIDResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

```

```
        <AuthenticateWindowsIDResult>AUTHENTICATION  
        TOKEN</AuthenticateWindowsIDResult>  
  
    </AuthenticateWindowsIDResponse>  
  
</s:Body>  
  
</s:Envelope>
```

...and again for Windows ID And Password Authentication...

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  
  
    <s:Body>  
  
        <AuthenticateWindowsIDAndPasswordResponse  
        xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">  
  
            <AuthenticateWindowsIDAndPasswordResult>AUTHENTICATION  
            TOKEN</AuthenticateWindowsIDAndPasswordResult>  
  
        </AuthenticateWindowsIDAndPasswordResponse>  
  
    </s:Body>  
  
</s:Envelope>
```

GetManagedList

Description: Retrieves a list of items from a named (non-user- created) Q-Pulse managed list for use in completing forms within Q-Pulse. The Managed List items returned include those that are now considered 'archived' in Q-Pulse. To indicate whether an item is archived, each item has a boolean 'IsArchived' flag. In the XML returned, if the IsArchived value is false, the tag is simply not declared. Person, Supplier, Department, Customer, Country, Asset, OccurrenceReportType, ReferenceList, IncidentPriority, IncidentStatus, NonConformanceSource, NonConformanceStatus, NonConformanceSeverity

Endpoint: `http://<servicepath>/Services/Core.svc/GetManagedList`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetManagedList>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:type>NonConformanceSource</v1:type>

    </v1:GetManagedList>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetManagedListResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetManagedListResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <Items>

          <ManagedListItem>

            <FullPath>Customer Complaint</FullPath>
```

```

        <ID>1</ID>

        <Name>Customer Complaint</Name>

        <ParentID>-1</ParentID>

    </ManagedListItem>

    <ManagedListItem>

        <FullPath>Customer Return</FullPath>

        <ID>2</ID>

        <Name>Customer Return</Name>

        <ParentID>-1</ParentID>

    </ManagedListItem>

    <ManagedListItem>

        <FullPath>Supplier</FullPath>

        <ID>3</ID>

        <Name>Supplier</Name>

        <ParentID>-1</ParentID>

    </ManagedListItem>

    <ManagedListItem>

        <FullPath>Employee Suggestion</FullPath>

        <ID>4</ID>

        <Name>Employee Suggestion</Name>

        <ParentID>-1</ParentID>

    </ManagedListItem>

    and so on .....

    </Items>

    <ListType>NonConformanceSource</ListType>

</GetManagedListResult>

```

```

    </GetManagedListResponse>

  </s:Body>

</s:Envelope>

```

SearchManagedList

Description: Enables searching for items within managed lists (except user-created lists).

Endpoint: `http://<servicepath>/Services/Core.svc/SearchForManagedListItem`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:SearchManagedList>

      <v1:token>ng0veA7xqebw1Ui0NNgHt+HHecdHjWmfz3SIDL8xJf8=</v1:token>

      <v1:type>Person</v1:type>

      <v1:searchString>Admin</v1:searchString>

    </v1:SearchManagedList>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <SearchManagedListResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <SearchManagedListResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <Items>

          <ManagedListItem>

```

```
<FullPath>Administrator, System</FullPath>

<ID>1</ID>

<Name>Administrator, System</Name>

<ParentID>-1</ParentID>

</ManagedListItem>

</Items>

<ListType>Person</ListType>

</SearchManagedListResult>

</SearchManagedListResponse>

</s:Body>

</s:Envelope>
```

GetServerAuthenticationType

Description: Gets the Authentication type being used by Q-Pulse, either Windows or Q-Pulse authentication. This can be used to decide whether to use Authenticate or AuthenticateWindowsID to perform authentication, depending on how the Q-Pulse installation is currently configured (the setting is in the Security -> Settings and Defaults section of the Admin module in Q-Pulse).

Endpoint: `http://<servicepath>/Services/Core.svc/GetServerAuthenticationType`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetServerAuthenticationType>

      <v1:dbName>DB_NAME</v1:dbName>

    </v1:GetServerAuthenticationType>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetServerAuthenticationTypeResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetServerAuthenticationTypeResult>QPulseAuthentication</GetServerAuth
        enticationTypeResult>

    </GetServerAuthenticationTypeResponse>

  </s:Body>

</s:Envelope>
```

HasServerLicenseExpired

Description: Determines whether the Q-Pulse server licence has expired. This can be used as a pre-requisite to authenticating the user, as any authentication attempt will fail if the license has indeed expired.

Endpoint: `http://<servicepath>/Services/Core.svc/HasServerLicenseExpired`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:HasServerLicenseExpired/>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <HasServerLicenseExpiredResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <HasServerLicenseExpiredResult>false</HasServerLicenseExpiredResult>

    </HasServerLicenseExpiredResponse>

  </s:Body>

</s:Envelope>
```

GetLoggedInPersonDetails

Description: Returns information about a person based on their authentication token.

Endpoint: `http://<servicepath>/Services/Core.svc/GetLoggedInPersonDetails`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>
```

```

<s:Body>

  <v1:GetLoggedInPersonDetails>

    <v1:token>AUTHENTICATION TOKEN</v1:token>

  </v1:GetLoggedInPersonDetails>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetLoggedInPersonDetailsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetLoggedInPersonDetailsResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <ID>1</ID>

        <Name>Administrator, System</Name>

      </GetLoggedInPersonDetailsResult>

    </GetLoggedInPersonDetailsResponse>

  </s:Body>

</s:Envelope>

```

SendEmail

Description: Used to send an email to a Q-Pulse user via the Q-Pulse messaging system. A string email body and subject should be specified as well as the Id of the Q-Pulse user to send the email to. The email body can be specified as HTML in which case the isHtml parameter should be set to true.

Endpoint: `http://<servicepath>/Services/Core.svc/SendEmail`

Sample Request:

```
</s:Envelope>

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:SendEmail>

      <v1:token> v1:token>

      <v1:subject>New Email</v1:subject>

      <v1:body>This is a test of emails</v1:body>

      <v1:isHtml>false</v1:isHtml>

      <v1:recipient>

        <v1:Bcc></v1:Bcc>

        <v1:Cc></v1:Cc>

        <v1:To>2</v1:To>

      </v1:recipient>

      <v1:from>test@qpulse.com</v1:from>

    </v1:SendEmail>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <SendEmailResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

    </s:Body>

  </s:Envelope>
```

Occurrence Service

Description: Provides access to Read and Create functionality related to the Q-Pulse Occurrence Module.

Endpoint: `http://<servicepath>/Services/Occurrence.svc`

ListAvailableReportTypes

Description: Gets a list of ReportTypes that are supported by the Occurrence Module. This will return the same results as Core.GetManagedList when specifying ReportTypes as the list to fetch. However, the key difference is that an authentication token has to be passed into Core.GetManagedList. This operation uses credentials stored in the Web Service Web.config to authenticate and get the list. This design decision was taken in order to provide the ability to perform read-ahead caching (e.g. a UI that uses Occurrence ReportTypes in a drop-down box can retrieve them either from a cached version or from the Web Service directly, without having to perform authentication).

Endpoint: `http://<servicepath>/Services/Occurrence.svc/ListAvailableReportTypes`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:ListAvailableReportTypes>

      <v1:databaseName>Bacchus</v1:databaseName>

    </v1:ListAvailableReportTypes>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <ListAvailableReportTypesResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <ListAvailableReportTypesResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <Items>
```

```

    <ManagedListItem>

        <FullPath>Cabin Safety</FullPath>

        <ID>1</ID>

        <Name>Cabin Safety</Name>

    </ManagedListItem>

    <ManagedListItem>

        <FullPath>Flight Safety</FullPath>

        <ID>2</ID>

        <Name>Flight Safety</Name>

    </ManagedListItem>

</Items>

<ListType>OccurrenceReportType</ListType>

</ListAvailableReportTypesResult>

</ListAvailableReportTypesResponse>

</s:Body>

</s:Envelope>

```

GetDataFieldsForReportType

Description: Gets a list of DataFields used by the specified report type. Whilst this uses the same DataField Contract as the Occurrence Contract returned by GetOccurrence and GetOccurrences, a call to this service will return additional information in terms of validation rules that are applied to the fields on the occurrence report.

If you require any form of UI validation, you should call this method and use the supplied validation information to build any UI logic you require for the Report Type requested.

Endpoint: `http://<servicepath>/Services/Occurrence.svc/GetDataFieldsForReportType`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

```

```

<s:Header/>

<s:Body>

  <v1:GetDataFieldsForReportType>

    <v1:reportTypeName>Cabin Safety</v1:reportTypeName>

    <v1:databaseName>Q-PULSE DATABASE NAME</v1:databaseName>

  </v1:GetDataFieldsForReportType>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDataFieldsForReportTypeResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDataFieldsForReportTypeResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <DataField>

          <DataType>String</DataType>

          <Description>Describe in detail the events leading up to
            the Occurrence and what happened</Description>

          <DisplayLabel>Occurrence Desc.</DisplayLabel>

          <Name>Occurrence Description</Name>

          <ReferenceListName/>

          <Validators/>

          <Value/>

        </DataField>

        <DataField>

          <DataType>Numeric</DataType>

```

```

        <Description>Enter the no. of hours that the plane was
        delayed if the delay was a result of this occurrence. Provide delay
        time to the nearest .1 of an hour if possible. E.g. If the delay was 1
        hour and six minutes, enter 1.1

        </Description>

        <DisplayLabel>Flight Delay [Hrs]</DisplayLabel>

        <Name>Flight Delay [Hrs]</Name>

        <ReferenceListName/>

        <Validators/>

        <Value/>

    </DataField>

    <DataField>

        <DataType>ListReference</DataType>

        <Description>From the dropdown list select the flight phase
        the aircraft was in at the time of the Occurrence</Description>

        <DisplayLabel>Flight Phase</DisplayLabel>

        <Name>Flight Phase</Name>

        <ReferenceListName>Flight Phase</ReferenceListName>

        <Validators/>

        <Value>-1</Value>

    </DataField>

</GetDataFieldsForReportTypeResult>

</GetDataFieldsForReportTypeResponse>

</s:Body>

</s:Envelope>

```

GetDataFields

Description: Gets a list of all available DataFields in Q-Pulse including archived fields if this is requested. This also uses the same DataField Contract as the Occurrence Contract returned by GetOccurrence and GetOccurrences, and so, like GetDataFieldsForReportType, will return additional information in terms of validation rules that are applied to the fields on the occurrence report.

Endpoint: <http://<servicepath>/Services/Occurrence.svc/GetDataFields>

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDataFields>

      <v1:databaseName>Q-PULSE DATABASE NAME</v1:databaseName>

      <v1:showArchived>false</v1:showArchived>

    </v1:GetDataFields>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDataFieldsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDataFieldsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <DataField>

          <DataType>ListReference</DataType>

          <Description>Select the appropriate IATA Code for the type
            of aircraft involved in this Occurrence.</Description>

          <DisplayLabel>Aircraft Type</DisplayLabel>

          <ElementIdentifier>Fae7a06ec82ad4577aff4d4b17211a013</ElementIdentifie
            r>

          <Name>A/C Type</Name>

          <ReferenceListName>Aircraft Type</ReferenceListName>

        </DataField>

      </GetDataFieldsResult>

    </GetDataFieldsResponse>

  </s:Body>

</s:Envelope>
```

```

        <Validators/>

        <Value>-1</Value>

    </DataField>

    <DataField>

        <DataType>String</DataType>

        <Description>Describe the actions that took place during
and after the Occurrence to minimise its impact and to resolve the
situation. Give some indication as to whether or not such actions
improved the situation or made it worse.</Description>

        <DisplayLabel>Action Taken</DisplayLabel>

    <ElementIdentifier>F6d1a8921f23e408db0eb1792ee4c9f1b</ElementIdentifie
r>

        <Name>Action Taken</Name>

        <ReferenceListName/>

        <Validators/>

        <Value/>

    </DataField>

    <DataField>

        <DataType>Numeric</DataType>

        <Description>If pertinent to the occurrence, record the
aircraft's actual landing mass, in Kg.</Description>

        <DisplayLabel>Landing Mass [Kg]</DisplayLabel>

    <ElementIdentifier>F65437275385d49da9c52b7d2c14a798b</ElementIdentifie
r>

        <Name>Actual Aircraft Landing Mass [Kg]</Name>

        <ReferenceListName/>

        <Validators/>

        <Value/>

```

```

        </DataField>
        <Validators/>

        <Value/>

    </DataField>

</GetDataFieldsResult>

</GetDataFieldsResponse>

</s:Body>

</s:Envelope>

```

GetReferenceList

Description: Exposes a list of items of a specified type for use in completing occurrence forms (within DataFields).

Endpoint: `http://<servicepath>/Services/Occurrence.svc/GetReferencesList`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetReferenceList>

      <v1:referenceListName>Cabin Activity</v1:referenceListName>

      <v1:databaseName>Q-PULSE DATABASE NAME</v1:databaseName>

    </v1:GetReferenceList>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"

  <s:Body>

```

```

    <GetReferenceListResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetReferenceListResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">

        <IsHierarchy>true</IsHierarchy>

        <Items>

          <ManagedListItem>

            <FullPath>Abnormal Safety Related duties</FullPath>

            <ID>724</ID>

            <Name>Abnormal Safety Related duties</Name>

            <ParentID>-1</ParentID>

          </ManagedListItem>

          <ManagedListItem>

            <FullPath>Beverage service</FullPath>

            <ID>717</ID>

            <Name>Beverage service</Name>

            <ParentID>-1</ParentID>

          </ManagedListItem>

          .....and so on.....

        </Items>

        <ListType>ReferenceList</ListType>

        <ReferenceListName>Cabin Activity</ReferenceListName>

      </GetReferenceListResult>

    </GetReferenceListResponse>

  </s:Body>

</s:Envelope>

```

CreateNewOccurrence

Description: Enables a user to create a new Occurrence record

Endpoint: `http://<servicepath>/Services/Occurrence.svc/CreateNewOccurrence`

Mandatory fields: ReportTypeName, ReportedByPersonName and ReportedDate.

Fields that will be set on the business object when creating an Occurrence (other fields can be specified in the XML but will be ignored): ReportTypeName, Number, IsConfidential, IsMOR, Title, IsRejected, IsMain, RecordedByPersonName, RecordedDate, ReportedDate, DataFields, Notes. Please note that for DataFields of reference list type there are two ways to supply its value. Firstly a DataField can be specified by the desired list item's ID by supplying the ID value in the 'Value' field. Alternatively you can specify the desired list item by supplying the list item's display value in the 'ListItemDisplayValue' field.

Sample Request:

```

<s:Envelope xmlns:s=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Body>

    <v1:CreateNewOccurrence>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:newOccurrence>

        <v1:DataFields>

          <v1:DataField>

            <v1:DataType>String</v1:DataType>

            <v1:Name>Occurrence Description</v1:Name>

            <v1:Value>This is a test occurrence</v1:Value>

          </v1:DataField>

          <v1:DataField>

            <v1:DataType>ListReference</v1:DataType>

            <v1:Name>Flight Phase</v1:Name>

            <v1:ReferenceListName>Flight Phase</v1:ReferenceListName>

            <v1:ListItemDisplayValue>Beverage
service</v1:ListItemDisplayValue>

            <v1:Value>717</v1:Value>

          </v1:DataField>

        </v1:DataFields>

        <v1:ReportTypeName>Cabin Safety</v1:ReportTypeName>

        <v1:ReportedByPersonName/>

        <v1:Title>A Test Title</v1:Title>

      </v1:newOccurrence>

    </v1:CreateNewOccurrence>

```

```

    </s:Body>
</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <CreateNewOccurrenceResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <CreateNewOccurrenceResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <Attachments/>

        <BrokenRules/>

        <DataFields>

          <DataField>

            <DataType>String</DataType>

            <DisplayLabel>Occurrence Desc.</DisplayLabel>

            <Name>Occurrence Description</Name>

            <ReferenceListName/>

            <Value>This is a test occurrence</Value>

          </DataField>

          <DataField>

            <DataType>ListReference</DataType>

            <Description>From the dropdown list select the flight phase
the aircraft was in at the time of the Occurrence</Description>

            <DisplayLabel>Flight Phase</DisplayLabel>

            <Name>Flight Phase</Name>

            <ReferenceListName>Flight Phase</ReferenceListName>

            <Validators/>

```

```

        <ListItemDisplayValue>Beverage
service</ListItemDisplayValue>

        <Value>717</Value>

    </DataField>

</DataFields>

<ID>15</ID>

<IncidentNumber/>

    <IncidentReportSentDate>0001-01-
01T00:00:00</IncidentReportSentDate>

    <IncidentStatusName/>

    <IncidentTitle/>

    <Notes/>

    <Number>CCR14</Number>

    <RecordedByPersonName>Administrator,
System</RecordedByPersonName>

    <RecordedDate>2009-01-27T16:10:00</RecordedDate>

    <ReportTypeName>Cabin Safety</ReportTypeName>

    <ReportedByPersonName>Administrator,
System</ReportedByPersonName>

    <ReportedDate>2009-01-27T16:10:00</ReportedDate>

    <Title>A Test Title</Title>

</CreateNewOccurrenceResult>

</CreateNewOccurrenceResponse>

</s:Body>

</s:Envelope>

```

GetOccurrence

Description: Enables a user to retrieve an existing Occurrence by its ID.

Endpoint: `http://<servicepath>/Services/Occurrence.svc/GetOccurrence`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetOccurrence>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:occurrenceID>1</v1:occurrenceID>

    </v1:GetOccurrence>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetOccurrenceResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetOccurrenceResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <Attachments/>

        <BrokenRules/>

        <DataFields>

          <DataField>

            <DataType>String</DataType>

            <DisplayLabel>Occurrence Desc.</DisplayLabel>

            <Name>Occurrence Description</Name>
```

```

        <ReferenceListName/>

        <Value>This is a test occurrence</Value>

    </DataField>

</DataFields>

<ID>18</ID>

<IncidentNumber/>

    <IncidentReportSentDate>0001-01-
01T00:00:00</IncidentReportSentDate>

    <IncidentStatusName/>

    <IncidentTitle/>

    <Notes/>

    <Number>CCR17</Number>

    <RecordedByPersonName>Administrator,
System</RecordedByPersonName>

    <RecordedDate>2009-01-27T16:20:00</RecordedDate>

    <ReportTypeName>Cabin Safety</ReportTypeName>

    <ReportedByPersonName>Administrator,
System</ReportedByPersonName>

    <ReportedDate>2009-01-27T16:20:00</ReportedDate>

    <Title/>

</GetOccurrenceResult>

</GetOccurrenceResponse>

</s:Body>

</s:Envelope>

```

GetOccurrences

Description: GetOccurrences allows you to search for Occurrences That match the search criteria.

Search criteria: The query object that you pass to this operation allows you to search against the date range the Occurrence was reported, whether it is mandatory, the person who recorded it, the status of the Occurrence (e.g. new, accepted, rejected or any) and any Keywords(this performs a 'contains' search against the Occurrence Number).

GetOccurrences also supports searching against Dynamic Fields. This is simply a case of defining the name of the field, the operator to use, and the value to search for. Whenever a field is of reference list type, the search value supplied should represent the list item display value except when used with the IdEqualTo operator where it should represent the list item's ID. (An array is used in order to support date ranges, in all other cases only the first item in the array will ever be used).

Endpoint: `http://<servicepath>/Services/Occurrence.svc/GetOccurrences`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

  <s:Header/>

  <s:Body>

    <v1:GetOccurrences>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:query>

        <v1:DynamicFieldConditions>

          <v1:DynamicFieldCondition>

            <v1:FieldName>Aircraft Registration</v1:FieldName>

            <v1:Operator>Containing</v1:Operator>

            <v1:Values>

              <arr:string>008</arr:string>

            </v1:Values>

          </v1:DynamicFieldCondition>

        </v1:DynamicFieldConditions>

      </v1:query>

    </v1:GetOccurrences>

  </s:Body>

</s:Envelope>
```

```

        </v1:DynamicFieldCondition>

    </v1:DynamicFieldConditions>

    <v1:IsMain>false</v1:IsMain>

    <v1:IsMandatory>false</v1:IsMandatory>

    <v1:Keywords>ASR</v1:Keywords>

    <v1:Number>OCC12</v1:Number>

    <v1:RecordedByPersonName/>

    <v1:ReportTypeName>Technical Safety</v1:ReportTypeName>

    <v1:ReportedOnOrAfter>2009-09-
01T13:56:00</v1:ReportedOnOrAfter>

    <v1:ReportedOnOrBefore>2009-07-
17T13:57:00</v1:ReportedOnOrBefore>

    <v1:Status>New</v1:Status>

</v1:query>

</v1:GetOccurrences>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <GetOccurrencesResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <GetOccurrencesResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">

                <Occurrence>

...Occurrence Results Go Here... (Set sample message for GetOccurrence
for Details.)

```

```
        </Occurrence>
      </GetOccurrencesResult>
    </GetOccurrencesResponse>
  </s:Body>
</s:Envelope>
```

GetOccurrenceIDs

Description: GetOccurrenceIDs is a lighter weight alternative to GetOccurrences. Where GetOccurrences returns full descriptions of found records, GetOccurrenceIDs returns only the Occurrence IDs. You can then retrieve the full details using GetOccurrence.

Search criteria: The query object that you pass to this operation allows you to search against the date range the Occurrence was reported, whether it is mandatory, the person who recorded it, the status of the Occurrence (e.g. new, accepted, rejected or any) and any Keywords(this performs a 'contains' search against the Occurrence Number).

GetOccurrenceIDs also supports searching against Dynamic Fields. This is simply a case of defining the name of the field, the operator to use, and the value to search for. Whenever a field is of reference list type the search value supplied should represent the list item display value except when used with the IdEqualTo operator where it should represent the list item's ID. (An array is used in order to support date ranges, in all other cases only the first item in the array will ever be used).

Endpoint: `http://<servicepath>/Services/Occurrence.svc/GetOccurrenceIDs`

Sample Request:

```
s<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

  <s:Header/>

  <s:Body>

    <v1:GetOccurrenceIDs>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:query>

        <v1:DynamicFieldConditions>

          <v1:DynamicFieldCondition>

            <v1:FieldName>Aircraft Registration</v1:FieldName>

            <v1:Operator>Containing</v1:Operator>

            <v1:Values>

              <arr:string>008</arr:string>

            </v1:Values>

          </v1:DynamicFieldCondition>

        </v1:DynamicFieldConditions>

      </v1:query>

    </v1:GetOccurrenceIDs>

  </s:Body>

</s:Envelope>
```

```

        </v1:Values>

        </v1:DynamicFieldCondition>

    </v1:DynamicFieldConditions>

    <v1:IsMain>false</v1:IsMain>

    <v1:IsMandatory>false</v1:IsMandatory>

    <v1:Keywords>ASR</v1:Keywords>

    <v1:Number>OCC12</v1:Number>

    <v1:RecordedByPersonName/>

    <v1:ReportTypeName>Technical Safety</v1:ReportTypeName>

    <v1:ReportedOnOrAfter>2009-09-
01T13:56:00</v1:ReportedOnOrAfter>

    <v1:ReportedOnOrBefore>2009-07-
17T13:57:00</v1:ReportedOnOrBefore>

    <v1:Status>New</v1:Status>

</v1:query>

</v1:GetOccurrenceIDs>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetOccurrenceIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetOccurrenceIDsResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <a:int>14</a:int>

        <a:int>15</a:int>

```

```

        </GetOccurrenceIDsResult>

    </GetOccurrenceIDsResponse>

</s:Body>

</s:Envelope>

```

GetOccurrenceGotoUrl

Description: GetOccurrenceGotoUrl returns a Q-Pulse Web Client URL for a particular Occurrence record. NOTE: The web services configuration file must have the appSetting 'QPulseWebLocation' configured to the appropriate Q-Pulse Web Client Site URL

Endpoint: <http://<servicepath>/Services/Occurrence.svc/GetOccurrenceGotoUrl>

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetOccurrenceGotoUrl>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:occurrenceID>1</v1:occurrenceID>

    </v1:GetOccurrenceGotoUrl>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetOccurrenceGotoUrlResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

```

```

    <GetOccurrenceGotoUrlResult>http://localhost/QPulse5Web/goto.aspx?cid=
    5&db=Default</GetOccurrenceGotoUrlResult>

    </GetOccurrenceGotoUrlResponse>

</s:Body>

</s:Envelope>

```

RejectOccurrence

Description: RejectOccurrence is used to set the IsRejected flag of an Occurrence to true, and therefore set the status of the Occurrence to Rejected. An optional string note can be specified which will be added to the Occurrence being rejected. also GetOccurrenceGotoUrl returns a Q-Pulse Web Client URL for a particular Occurrence record. NOTE: The web services configuration file must have the appSetting 'QPulseWebLocation' configured to the appropriate Q-Pulse Web Client Site URL

Endpoint: <http://<servicepath>/Services/Occurrence.svc/GetOccurrenceGotoUrl>

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:RejectOccurrence>

      <v1:token>AUTHORISATION TOKEN</v1:token>

      <v1:occurrenceID>1</v1:occurrenceID>

      <v1:note>Reason for rejection</v1:note>

    </v1:RejectOccurrence>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"

```

```
<s:Body>

  <RejectOccurrenceResponse
    xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

  </s:Body>

</s:Envelope>
```


Incident Service

Description: This service provides access to Create and Read functionality related to the Q-Pulse Incident module.

Endpoint: `http://<servicepath>/Services/Incident.svc`

GetIncident

Description: Enables a user to retrieve an existing Incident.

Endpoint: `http://<servicepath>/Services/Incident.svc/GetIncident`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetIncident>

      <v1:token> AUTHORISATION TOKEN</v1:token>

      <v1:incidentID>1</v1:incidentID>

      <v1:includeOccurrences>false</v1:includeOccurrences>

    </v1:GetIncident>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetIncidentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">
```

```

    <GetIncidentResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">

        <Attachments/>

        <Classifications>

            <Classification>

                <ClassificationPath>Liveware (human)</ClassificationPath>

                <DisplayValue>101000000 Physical/sensory
limits</DisplayValue>

                <ListName>Explanatory Factors</ListName>

            </Classification>

            <Classification>

                <ClassificationPath>Hard/software
interface</ClassificationPath>

                <DisplayValue>301000000 Human/hardware
interface</DisplayValue>

                <ListName>Explanatory Factors</ListName>

            </Classification>

        </Classifications>

        <ClosedByPerson/>

        <Costs/>

        <Currency/>

        <DateCreated>2009-01-23T15:09:00</DateCreated>

        <Department/>

        <HasMainOccurrence>true</HasMainOccurrence>

        <ID>1</ID>

        <Notes>

            <Note>

                <CreatedByPersonName>Administrator,
System</CreatedByPersonName>

```

```

        <DateCreated>2009-01-23T15:24:00</DateCreated>

        <ID>1</ID>

        <NoteSubject/>

        <NoteText>reveryrey</NoteText>

    </Note>

</Notes>

<Number>INC1</Number>

    <OccurrenceIDs
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

        <a:int>4</a:int>

        <a:int>5</a:int>

        <a:int>6</a:int>

    </OccurrenceIDs>

    <Occurrences/>

    <Owner>Administrator, System</Owner>

    <Priority/>

    <ReportTypeName>Flight Safety</ReportTypeName>

    <Stages>

        <IncidentStage>

            <Actions/>

            <Attachments/>

            <ClosedByPersonName/>

            <Description>Analyse incident and identify probable root
causes.</Description>

            <Findings>

                <IncidentStageFinding>

                    <Classifications/>

```

```

        <Description>yrturturteurteuee</Description>

        <DisplayValue>1-1 - yrturturteurteuee</DisplayValue>

        <FindingType>Observation</FindingType>

        <Number>1</Number>

        <RiskAssessments>

            <RiskAssessment>

                <Comment>ryeryeryery</Comment>

                <MatrixName>4X4 Risk Tolerance</MatrixName>

                <MatrixXAxisTitle>Severity</MatrixXAxisTitle>

                <MatrixXValue>Moderate</MatrixXValue>

                <MatrixYAxisTitle>Likelihood</MatrixYAxisTitle>

                <MatrixYValue>B (likely)</MatrixYValue>

                <RiskValue>H</RiskValue>

                <SuggestedAction>Manage by routine
procedures</SuggestedAction>

                <Title>1-1 - yrturturteurteuee</Title>

            </RiskAssessment>

        </RiskAssessments>

    </IncidentStageFinding>

</Findings>

<OwnerPersonName/>

<StageTypeName>Investigation</StageTypeName>

</IncidentStage>

<IncidentStage>

    <Actions/>

    <Attachments/>

    <ClosedByPersonName/>

```

```

        <Description>Review implementation and approve for ongoing
usage</Description>

        <Findings/>

        <OwnerPersonName/>

        <StageTypeName>Approval</StageTypeName>

    </IncidentStage>

    <IncidentStage>

        <Actions/>

        <Attachments/>

        <ClosedByPersonName/>

        <Description>Identify solutions, assess risk, plan deployment
and implement</Description>

        <Findings/>

        <OwnerPersonName/>

        <StageTypeName>Implementation</StageTypeName>

    </IncidentStage>

</Stages>

<Status>Open</Status>

<TargetDate>2009-01-30T00:00:00</TargetDate>

<Title/>

</GetIncidentResult>

</GetIncidentResponse>

</s:Body>

</s:Envelope>

```

GetIncidents

Description: Enables a user to search for Incident records by type, status, priority, the person that currently "owns" the incident, event date, whether an Incident must be reported to an external body (i.e. an MOR) and any keyword(s) associated with the Incident.

Search criteria: The query object that you pass to this operation allows you to search against the date range the Incident was reported, whether it is mandatory, the person who owns it, the status of the Incident, the name of the Report Type, Incident Priority and any Keywords(this performs a 'contains' search against the Incident Number and Title).

Additionally, there is a flag 'includeOccurrences' to determine whether any Occurrence associated with an Incident (that matches the search criteria) should be included in the response.

Endpoint: `http://<servicepath>/Services/Incident.svc/GetIncidents`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">
  <s:Header/>
  <s:Body>
    <v1:GetIncidents>
      <v1:token>AUTHORISATION TOKEN</v1:token>
      <!--Optional:-->
      <v1:query>
        <!--Optional:-->
        <v1:EventDateOnOrAfter>2009-06-
05T14:21:00</v1:EventDateOnOrAfter>
        <!--Optional:-->
        <v1:EventDateOnOrBefore>2009-09-
01T14:21:00</v1:EventDateOnOrBefore>
        <!--Optional:-->
        <v1:IsMandatory>true</v1:IsMandatory>
        <v1:ReportTypeName>Flight Safety</v1:ReportTypeName>
```

```

        </v1:query>

        <v1:includeOccurrences>false</v1:includeOccurrences>

    </v1:GetIncidents>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetIncidentsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetIncidentsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <Incident>

          ...Incident Result Goes here... (See Sample Response for message format)

        </Incident>

      </GetIncidentsResult>

    </GetIncidentsResponse>

  </s:Body>

</s:Envelope>

```

GetIncidentIDs

Description: GetIncidentIDs is a lighter weight alternative to GetIncidents. Where GetIncidents returns full descriptions of found records, GetIncidentIDs returns only the Incident Numbers. You can then retrieve the full details using GetIncident

Search criteria: The query object that you pass to this operation allows you to search against the date range the Incident was reported, whether it is mandatory, the person who

owns it, the status of the Incident, the name of the Report Type, Incident Priority and any Keywords(this performs a 'contains' search against the Incident Number and Title).

Endpoint: `http://<servicepath>/Services/Incident.svc/GetIncidentIDs`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetIncidentIDs>

      <v1:token>AUTHORISATION TOKEN=</v1:token>

      <v1:query>

        <v1:EventDateOnOrAfter>2009-06-
05T14:21:00</v1:EventDateOnOrAfter>

        <v1:EventDateOnOrBefore>2009-09-
01T14:21:00</v1:EventDateOnOrBefore>

        <v1:IsMandatory>true</v1:IsMandatory>

        <v1:ReportTypeName>Flight Safety</v1:ReportTypeName>

      </v1:query>

    </v1:GetIncidentIDs>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetIncidentIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetIncidentIDsResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```
<a:int>165</a:int>

<a:int>167</a:int>

<a:int>169</a:int>

<a:int>170</a:int>

</GetIncidentIDsResult>

</GetIncidentIDsResponse>

</s:Body>

</s:Envelope>
```

Attachment Service

Allows you to attach and read files related to Occurrences, NonConformances, and Audits and read files related to Incidents stored within Q-Pulse.

The attachment service differs slightly from the other services provided as it does not support a SOAP interface. This provides maximum flexibility in how you want to deal with attachments without the constraints enforced by a Schema (for example, we want to enable clients to be built that support progressive feedback). Making REST HTTP requests without the aid of a SOAP Toolkit is very simple. All modern platforms support HTTP. Using standard libraries you can use the following objects/methods to make HTTP requests:

- **Microsoft .NET (C#/VB.NET)** - WebClient
- **Java** - UrlConnection
- **PHP** - @fopen
- **JQuery** - JQuery.post()
- **Google Web Toolkit (GWT)** - com.google.gwt.http.client.RequestBuilder
- **OpenJS / jx** - jx.load();
- **Prototype** - Ajax.Request object.
- **Dojo** - dojo.io.bind();
- **Yahoo! UI** - YAHOO.util.Connect.asyncRequest();
- **MochiKit** - doSimpleXMLHttpRequest()

The method and syntax for all these libraries is relatively similar. For example...

Using C# / .Net 3.0

```
using(var client = new WebClient())
{
    client.Headers.Add("Operation", "Upload");

    client.Headers.Add("Token", token);

    client.Headers.Add("ObjectType", "Occurrence");

    client.Headers.Add("ObjectID", "1");

    byte[] response = client.UploadFile(
```

```
"http://<install path>/Services/Attachment.ashx",  
    @"C:\<file path>\Attachment.docx");  
}
```

...or using PHP...

```
$params = array('http' => array(
    'method' => 'POST',
    'content' => $filedata
));

$params['http']['header'] = array(
    'Operation' => 'Upload',
    'ObjectID' => '12',
    'ObjectType' => 'Occurrence',
    'ObjectID' => '1'
)

$cctx = stream_context_create($params);

$fp = @fopen('http://<install_path>/Services/Attachments.ashx', 'rb',
    false, $cctx);

$response = @stream_get_contents($fp);
```

NOTE: When looking at raw HTTP Request messages, be aware that your chosen library must support editing HTTP Request headers and that some headers may be generated by your chosen client library.

PostAttachment

Description: Attaches the uploaded file to the specified record.

Endpoint: `http://<servicepath>/Attachments.ashx`

Sample Request:

```
POST /QPulseWebServices/Services/Attachment.ashx HTTP/1.1
```

```
Operation: Upload

Token: ng0veA7xqebw1Ui0NNgHt+HHecdHjWmfz3SIDL8xJf8=

ObjectType: Occurrence

ObjectID: 1

Content-Type: multipart/form-data; boundary=-----
      8cb48fbb149a67f

Host: localhost

Content-Length: 116926

Expect: 100-continue


-----8cb48fbb149a67f

Content-Disposition: form-data; name="file"; filename="testUpload.docx"

Content-Type: application/octet-stream
```

Sample Response:

```
HTTP/1.1 200 OK

Cache-Control: private

Content-Type: text/plain; charset=utf-8

Server: Microsoft-IIS/7.0

X-AspNet-Version: 2.0.50727

X-Powered-By: ASP.NET

Date: Tue, 20 Jan 2009 09:50:43 GMT

Content-Length: 2


200 OK
```

GetAttachment

Description: Retrieves an embedded attachment from the Q-Pulse database.

Endpoint: `http://<servicepath>/Attachments.ashx`

Sample Request:

```
GET /QPulseWebServices/Services/Attachment.ashx HTTP/1.1

Operation: Download

Token: ng0veA7xqebw1UissNgHt+HHssdhjWmfz3SIDL8xJf8=

ObjectType: Occurrence

ObjectID: 1

AttachmentID: 1

Host: localhost
```

Sample Response:

```
HTTP/1.1 200 OK

Cache-Control: private

Transfer-Encoding: chunked

Content-Type: application/msword

Server: Microsoft-IIS/7.0

Content-Disposition: inline; filename="testUpload.docx"

X-AspNet-Version: 2.0.50727

X-Powered-By: ASP.NET

Date: Tue, 20 Jan 2009 09:54:36 GMT

400

Content
```


Audit Service

Description: Provides access to Create and Read functionality related to the Q-Pulse Audit Module.

Endpoint: `http://<servicepath>/Services/Audit.svc`

GetAudit

Description: Enables a user to retrieve an existing Audit.

Endpoint: `http://<servicepath>/Services/Audit.svc/GetAudit`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">
  <s:Header/>
  <s:Body>
    <v1:GetAudit>
      <v1:token>AUTHORISATION TOKEN</v1:token>
      <v1:auditID>1</v1:auditID>
    </v1:GetAudit>
  </s:Body>
</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetAuditResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">
      <GetAuditResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">
        <ActualDisplayDuration>0 Day/s</ActualDisplayDuration>
```

```

    <ActualDuration>0</ActualDuration>

    <ActualDurationUnitID>Days</ActualDurationUnitID>

    <ActualDurationUnitName>Day/s</ActualDurationUnitName>

    <AgreedByPersonID>-1</AgreedByPersonID>

    <AgreedByPersonName/>

    <AgreedOnDateTime>0001-01-01T00:00:00</AgreedOnDateTime>

    <AllDayEvent>false</AllDayEvent>

    <Attachments/>

    <AuditCategoryDescription>Health and Safety
Audit</AuditCategoryDescription>

    <AuditCategoryID>6</AuditCategoryID>

    <AuditChecklistsCollection/>

    <AuditFindingsCollection/>

    <AuditNotesCollection/>

    <AuditScopeItemsCollection>

        <AuditScopeItem>

            <AuditID>1</AuditID>

            <Description>ISO14001</Description>

            <FriendlyName>31</FriendlyName>

            <ID>31</ID>

            <IsMain>true</IsMain>

            <IsPlanned>true</IsPlanned>

            <Name>ISO14001</Name>

            <Path>ISO14001</Path>

            <Type>AreaOfStandard</Type>

        </AuditScopeItem>

    </AuditScopeItemsCollection>

```

```

<AuditeesCollection/>

<AuditeesCount>0</AuditeesCount>

<AuditorsCollection>

  <Auditor>

    <AuditID>1</AuditID>

    <DepartmentOrganisationName/>

    <ID>2</ID>

    <IsLeadAuditor>true</IsLeadAuditor>

    <Note/>

    <PersonName>Martin, Graham</PersonName>

  </Auditor>

</AuditorsCollection>

<AuditorsCount>1</AuditorsCount>

<BaselineDuration>-1</BaselineDuration>

<BaselineDurationUnitName/>

<BrokenRulesCollection/>

<CanAccept>false</CanAccept>

<CanClose>false</CanClose>

<ClosedByPersonID>-1</ClosedByPersonID>

<ClosedByPersonName/>

<CreatedByPersonID>1</CreatedByPersonID>

<FriendlyName>AUD1 - ISO14001</FriendlyName>

<HasBeenAgreed>false</HasBeenAgreed>

<ID>1</ID>

<LeadAuditor>Martin, Graham</LeadAuditor>

<LeadAuditorPersonID>2</LeadAuditorPersonID>

```

```

    <MainScopeID>31</MainScopeID>

    <MainScopeItem>ISO14001</MainScopeItem>

    <MainScopeType>AreaOfStandard</MainScopeType>

    <Number>AUD1</Number>

    <NumberOfIncidents>0</NumberOfIncidents>

    <NumberOfInitiatives>0</NumberOfInitiatives>

    <NumberOfObservations>0</NumberOfObservations>

    <NumberOfOutstandingIncidents>0</NumberOfOutstandingIncidents>

    <Overview/>

    <ScheduleConfirmed>false</ScheduleConfirmed>

    <ScheduledDisplayDuration>0 Day/s</ScheduledDisplayDuration>

    <ScheduledDuration>0</ScheduledDuration>

    <ScheduledDurationUnitID>Days</ScheduledDurationUnitID>

    <ScheduledDurationUnitName>Day/s</ScheduledDurationUnitName>

    <ScheduledEnd>2012-05-01T00:00:00</ScheduledEnd>

    <ScheduledEndCalendar>2012-05-
02T00:00:00</ScheduledEndCalendar>

    <ScheduledEndDateTime>2012-05-
02T00:00:00</ScheduledEndDateTime>

    <ScheduledStart>2012-05-01T00:00:00</ScheduledStart>

    <ScheduledStartDate>01/05/2012</ScheduledStartDate>

    <ScopeDepartmentID/>

    <ScopeItemCount>1</ScopeItemCount>

    <SeriesInfo/>

    <StatusID>1</StatusID>

    <StatusIndicator>1</StatusIndicator>

    <StatusName>Scheduled</StatusName>

```

```
<Title>ISO14001</Title>

<TotalFindingsCount>0</TotalFindingsCount>

<UserDefinedField1Description/>

<UserDefinedField1ID>-1</UserDefinedField1ID>

<UserDefinedField2Description/>

<UserDefinedField2ID>-1</UserDefinedField2ID>

<auditNonConformanceFindingsCollection/>

</GetAuditResult>

</GetAuditResponse>

</s:Body>

</s:Envelope>
```

GetAudits

Description: Enables a user to search for Audit records by Auditor name, Calendar, Scheduled Date, Audit status and any keyword(s) associated with the Audit.

Search criteria: The query object that you pass to this operation allows you to search against the Auditor name, Calendar, Scheduled date, Status and Keywords (which performs a 'contains' search against the Number, Title and Overview of Audits).

Endpoint: `http://<servicepath>/Services/Audit.svc/GetAudits`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:GetAudits>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:query>

        <v1:AuditorName>Administrator, System</v1:AuditorName>

        <v1:CalendarCategoryID>2</v1:CalendarCategoryID>

        <v1:Keywords i:nil="true"/>

        <v1:ScheduledDate>

          <v1:EndDate i:nil="true"/>

          <v1:Operator>After</v1:Operator>

          <v1:StartDate>2009-06-15T11:48:00</v1:StartDate>

        </v1:ScheduledDate>

        <v1:StatusID>2</v1:StatusID>

      </v1:query>

    </v1:GetAudits>

  </s:Body>

</s:Envelope>
```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetAuditsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetAuditsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance"/>
        <Audit>

          ...Audit Result Goes here... (See Sample Response to
          GetAudit for message format)

        </Audit>

      </GetAuditsResult>

    </GetAuditsResponse>

  </s:Body>

</s:Envelope>

```

GetAuditIDs

Description: GetAuditIDs is a lighter weight alternative to GetAudits. Where GetAudits returns full descriptions of found records, GetAuditIDs returns only the Audit IDs. You can then retrieve the full details using GetAudit.

Search criteria: The query object that you pass to this operation allows you to search against the Auditor name, Calendar, Scheduled date, Status and Keywords (which performs a 'contains' search against the Number, Title and Overview of Audits).

Endpoint: http://<servicepath>/Services/Audit.svc/GetAuditIDs

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

```

```

<s:Body>

  <v1:GetAuditIDs>

    <v1:token>AUTHENTICATION TOKEN</v1:token>

    <v1:query>

      <v1:AuditorName>Administrator, System</v1:AuditorName>

      <v1:CalendarCategoryID>2</v1:CalendarCategoryID>

      <v1:Keywords i:nil="true"/>

      <!--Optional:-->

      <v1:ScheduledDate>

        <v1:EndDate i:nil="true"/>

        <v1:Operator>After</v1:Operator>

        <v1:StartDate>2009-06-15T11:48:00</v1:StartDate>

      </v1:ScheduledDate>

      <v1:StatusID>2</v1:StatusID>

    </v1:query>

  </v1:GetAuditIDs>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetAuditIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetAuditIDsResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
        <a:int>1</a:int>
      </GetAuditIDsResult>
    </GetAuditIDsResponse>
  </s:Body>
</s:Envelope>

```

```

        </GetAuditIDsResult>

    </GetAuditIDsResponse>

</s:Body>
</s:Envelope>

```

CreateNewAudit

Description: Enables a user to create a new Audit

Endpoint: `http://<servicepath>/Services/Audit.svc/CreateNewAudit`

Mandatory fields: AuditCategoryID, Title, ScheduledStartDate

Fields that will be set on the business object when creating an Audit (other fields can be specified in the XML but will be ignored):

AuditCategoryID, AuditCategoryDescription, SeriesInfo, Number, Title, ScheduledDuration, ScheduledDurationUnitID, ScheduleConfirmed, ActualDuration, ActualDurationUnitID, AgreedByPersonID, UserDefinedField1ID, UserDefinedField1Description, UserDefinedField2ID, UserDefinedField2Description, Overview, ScheduledStartDate

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:CreateNewAudit>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:newAudit>

        <v1:AuditCategoryID>1</v1:AuditCategoryID>

        <v1:ScheduledStartDate>2012-05-
01T12:47:00</v1:ScheduledStartDate>

        <v1:Title>Test audit</v1:Title>

      </v1:newAudit>
    </v1:CreateNewAudit>
  </s:Body>
</s:Envelope>

```

```
</v1:CreateNewAudit>

</s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <CreateNewAuditResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <CreateNewAuditResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">
        ...Audit Result Goes here... (See Sample Response to
        GetAudit for message format)

      </CreateNewAuditResult>

    </CreateNewAuditResponse>

  </s:Body>

</s:Envelope>
```

Document Service

Description: Provides access to Create and Read functionality related to the Q-Pulse Document Module.

Endpoint: `http://<servicepath>/Services/Document.svc`

GetDocument

Description: Enables a user to retrieve an existing Active Document by supplying its document ID.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocument`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocument>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentID>1</v1:documentID>

    </v1:GetDocument>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentResult xmlns:i="http://www.w3.org/2001/XMLSchema-
        instance">

        <ActiveDateTime>2012-04-27T00:00:00</ActiveDateTime>

      </GetDocumentResult>

    </GetDocumentResponse>

  </s:Body>

</s:Envelope>
```

```

        <DocumentTypeDescription>Doc Type</DocumentTypeDescription>

        <DocumentTypeID>1</DocumentTypeID>

        <ID>1</ID>

        <Number>DOC1</Number>

        <RevisionID>1</RevisionID>

        <RevisionIDs
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

            <a:int>1</a:int>

        </RevisionIDs>

        <RevisionReference>

            <IsSet>true</IsSet>

            <Value>1</Value>

        </RevisionReference>

        <Status>Active</Status>

        <StatusName>Active</StatusName>

        <Title>Title of document</Title>

        and so on .....

    </GetDocumentResult>

</GetDocumentResponse>

</s:Body>

</s:Envelope>

```

GetDocumentByNumber

Description: Enables a user to retrieve an existing Active Document by supplying its document Number.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentByNumber`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentByNumber>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentNumber>DOC1</v1:documentNumber>

    </v1:GetDocumentByNumber>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentByNumberResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentByNumberResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)
      </GetDocumentByNumberResult>

    </GetDocumentByNumberResponse>

  </s:Body>

</s:Envelope>

```

GetDocumentByRevisionReference

Description: Enables a user to retrieve an existing Document by supplying its document Number and Revision Reference.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentByRevisionReference`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentByRevisionReference>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentNumber>DOC1</v1:documentNumber>

      <v1:documentRevisionReference>1</v1:documentRevisionReference>

    </v1:GetDocumentByRevisionReference>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentByRevisionReferenceResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentByRevisionReferenceResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)
      </GetDocumentByRevisionReferenceResult>

    </GetDocumentByRevisionReferenceResponse>

  </s:Body>

</s:Envelope>
```

GetDocumentByRevisionID

Description: Enables a user to retrieve an existing Document by supplying its Revision ID.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentByRevisionID`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentByRevisionID>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentRevisionID>1</v1:documentRevisionID>

    </v1:GetDocumentByRevisionID>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentByRevisionIDResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentByRevisionIDResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)

      </GetDocumentByRevisionIDResult>

    </GetDocumentByRevisionIDResponse>

  </s:Body>
```

```
</s:Envelope>
```

GetDocuments

Description: Enables a user to search for Document records by Approver, Copyholder, Document owner, Document status, Document type and any keyword(s) associated with the Document.

Search criteria: The query object that you pass to this operation allows you to search against the Document Approver, Copy Holder, Document Owner, Document Number, Document Status, Document Type and Keywords (which is used to perform a 'contains' search against Document Number, Title and Keywords).

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocuments`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:GetDocuments>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:query>

        <v1:ApproverID i:nil="true"/>

        <v1:CopyHolderID i:nil="true"/>

        <v1:DocumentOwnerID>2</v1:DocumentOwnerID>

        <v1:DocumentStatus>Active</v1:DocumentStatus>

        <v1:DocumentTypeID>1</v1:DocumentTypeID>

        <v1:Keywords i:nil="true"/>

      </v1:query>

    </v1:GetDocuments>

  </s:Body>
```

```
</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">
      <GetDocumentsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <Document>
          ...Document Result Goes here... (See Sample Response to
GetDocument for message format)
        </Document>
      </GetDocumentsResult>
    </GetDocumentsResponse>
  </s:Body>
</s:Envelope>
```

GetDocumentIDs

Description: GetDocumentIDs is a lighter-weight alternative to GetDocuments. Where GetDocuments returns full descriptions of found records, GetDocumentIDs returns only the Document Numbers. You can then retrieve the full details using GetDocument.

Search criteria: The query object that you pass to this operation allows you to search against the Document Approver, Copy Holder, Document Owner, Document Number, Document Status, Document Type and Keywords (which is used to perform a 'contains' search against Document Number, Title and Keywords).

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentIDs`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```

<s:Header/>

<s:Body>

  <v1:GetDocumentIDs>

    <v1:token>AUTHENTICATION TOKEN</v1:token>

    <v1:query>

      <v1:ApproverID i:nil="true"/>

      <v1:CopyHolderID i:nil="true"/>

      <v1:DocumentOwnerID>2</v1:DocumentOwnerID>

      <v1:DocumentStatus>Active</v1:DocumentStatus>

      <v1:DocumentTypeID>1</v1:DocumentTypeID>

      <v1:Keywords i:nil="true"/>

    </v1:query>

  </v1:GetDocumentIDs>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentIDsResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <a:int>1</a:int>

        <a:int>36</a:int>

      </GetDocumentIDsResult>

    </GetDocumentIDsResponse>

  </s:Body>

</s:Envelope>

```

```

    </s:Body>
</s:Envelope>

```

GetDocumentsByIDs

Description: Enables a user to retrieve a list of full document descriptions by supplying a list of document IDs.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentsByIDs`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentsByIDs>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentIDs>

        <arr:int>1</arr:int>

        <arr:int>22</arr:int>

      </v1:documentIDs>

    </v1:GetDocumentsByIDs>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentsByIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

```

```

    <GetDocumentsByIdsResult
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

      <Document>
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)

      </Document>

      <Document>

        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)

      </Document>

    </GetDocumentsByIdsResult>

  </GetDocumentsByIdsResponse>

</s:Body>
</s:Envelope>

```

GetDocumentSummaries

Description: Enables a user to search for the summaries of Document records by Approver, Copyholder, Document owner, Document status, Document type and any keyword(s) associated with the Document. A DocumentSummary is a subset of the Document record.

Search criteria: The query object that you pass to this operation allows you to search against the Document Approver, Copy Holder, Document Owner, Document Number, Document Status, Document Type and Keywords (which is used to perform a 'contains' search against Document Number, Title and Keywords).

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentSummaries`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentSummaries>

```

```

    <v1:token>AUTHENTICATION TOKEN</v1:token>

    <v1:query>

        <v1:ApproverID i:nil="true"/>

        <v1:CopyHolderID i:nil="true"/>

        <v1:DocumentOwnerID>2</v1:DocumentOwnerID>

        <v1:DocumentStatus>Active</v1:DocumentStatus>

        <v1:DocumentTypeID>1</v1:DocumentTypeID>

        <v1:Keywords i:nil="true"/>

    </v1:query>

    </v1:GetDocumentSummaries>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <GetDocumentSummariesResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <GetDocumentSummariesResult
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

                <DocumentSummary>

                    <ActiveDateTime>2012-04-27T00:00:00</ActiveDateTime>

                    <CanView>true</CanView>

                <DefaultAttachmentStorageType>Embedded</DefaultAttachmentStorageType>

                <DefaultDocumentAttachmentItem>

                    <AbsolutePath/>

                    <AttachmentType>File</AttachmentType>

                </DefaultDocumentAttachmentItem>

            </GetDocumentSummariesResult>

        </GetDocumentSummariesResponse>

    </s:Body>

</s:Envelope>

```

```

        <DisplayFileName>Create
Occurrence.txt</DisplayFileName>

        <FileSize>2920</FileSize>

        <ModifiedDate>2011-06-27T09:49:23.81</ModifiedDate>

        <RelativePath/>

        <RevisionID>1</RevisionID>

    </DefaultDocumentAttachmentItem>

    <DocumentDeleted>false</DocumentDeleted>

    <DocumentTypeDescription>GM Type</DocumentTypeDescription>

    <FriendlyName>DOC1 (1)</FriendlyName>

    <HasAttachedFiles>true</HasAttachedFiles>

    <HasDraftRevision>false</HasDraftRevision>

    <ID>1</ID>

    <Number>DOC1</Number>

    <ReviewDateTime>2013-04-27T00:00:00</ReviewDateTime>

    <RevisionID>1</RevisionID>

    <RevisionReference>

        <IsSet>true</IsSet>

        <Value>1</Value>

    </RevisionReference>

    <Status>Active</Status>

    <StatusName>Active</StatusName>

    <Title>GM Doc</Title>

    </DocumentSummary>

</GetDocumentSummariesResult>

</GetDocumentSummariesResponse>

</s:Body>

```

```
</s:Envelope>
```

GetDocumentSummariesByIDs

Description: Enables a user to retrieve a list of document summaries by supplying a list of document IDs.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentSummariesByIDs`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentSummariesByIDs>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentIDs>

        <arr:int>1</arr:int>

        <arr:int>14</arr:int>

      </v1:documentIDs>

    </v1:GetDocumentSummariesByIDs>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentSummariesByIDsResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentSummariesByIDsResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <DocumentSummary>
          ...DocumentSummary Result Goes here... (See Sample Response
          to GetDocumentSummaries for message format)
        </DocumentSummary>
        ...
      </GetDocumentSummariesByIDsResult>
    </GetDocumentSummariesByIDsResponse>

  </s:Body>

</s:Envelope>
```

```

        </DocumentSummary>

        <DocumentSummary>

            ...DocumentSummary Result Goes here... (See Sample Response to
            GetDocumentSummaries for message format)

        </DocumentSummary>

    </GetDocumentSummariesByIdsResult>

</GetDocumentSummariesByIdsResponse>

</s:Body>

</s:Envelope>

```

GetDocumentAttachment

Description: Enables a user to retrieve an existing Active Document's attachment by passing a document ID.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentAttachment`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentAttachment>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentID>1</v1:documentID>

    </v1:GetDocumentAttachment>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentAttachmentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentAttachmentResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <ID>1</ID>

        <Items>

          <DocumentAttachmentItem>

            <AbsolutePath/>

            <AttachmentType>File</AttachmentType>

            <DisplayFileName>Test.txt</DisplayFileName>

            <FileSize>2920</FileSize>

            <ModifiedDate>2011-06-27T09:49:23.81</ModifiedDate>

            <RelativePath/>

            <RevisionID>1</RevisionID>

          </DocumentAttachmentItem>

        </Items>

        <Name>Create Occurrence</Name>

        <StorageType>Embedded</StorageType>

      </GetDocumentAttachmentResult>

    </GetDocumentAttachmentResponse>

  </s:Body>

</s:Envelope>

```

GetDocumentAndFolders

Description: Enables a user to retrieve information about a document and any folders that the document is associated with.

Endpoint: <http://<servicepath>/Services/Document.svc/GetDocumentAndFolders>

Availability: Q-Pulse Web Services API version 02.000.580.010 and above

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentFolders xmlns="
http://www.qpulse.com/QPulseWebServices/v1.2/"
      <token>AUTHENTICATION_TOKEN</token>
      <documentID>3</type>
    </GetDocumentFolders>
  </s:Body>
</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentAndFoldersResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">
      <GetDocumentAndFoldersResult
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <ActiveDateTime>2009-06-04T00:00:00</ActiveDateTime>
        <AssociatedFolders>
          <DocumentFolder>
            <Comment/>
            <FolderCategoryID>2</FolderCategoryID>
```

```

    <FolderCategoryName>Default</FolderCategoryName>

    <FolderStatus>Private</FolderStatus>

    <FolderType>Virtual</FolderType>

    <ID>3</ID>

    <LastModifiedDate></LastModifiedDate>

    <Name>Test Folder</Name>

    <RegisterViewType></RegisterViewType>

  </DocumentFolder>

</AssociatedFolders>

  <AuthorPersonName/>Bloggs, Joe</AuthorPersonName>

  <ChangeDetails/>

  <DocumentID>1</DocumentID>

  <DocumentNotesCollection/>

  <DocumentTypeDescription>docType3</DocumentTypeDescription>

  <DocumentTypeID>3</DocumentTypeID>

  <ID>1</ID>

  <Keywords/>

  <Number>DOC1</Number>

  <OwnerPersonName/>

  <RelatedDocumentIDs/>

  <ReviewDateTime>2010-06-04T00:00:00</ReviewDateTime>

  <ReviewsCount>0</ReviewsCount>

  <RevisionID/>

  <RevisionReference>

    <IsSet>true</IsSet>

    <Value>1</Value>

```

```

        </RevisionReference>

        <Status>Active</Status>

        <Title>Test Document</Title>

    </GetDocumentAndFolderResult>

</GetDocumentAndFolderResponse>

</s:Body>

</s:Envelope>

```

GetDocumentFolders

Description: Enables a user to retrieve document folders by folder type.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentFolders`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentFolders>

      <v1:token>ng0veA7xqebw1Ui0NNgHt+HHecdhjWmfz3SIDL8xJf8=</v1:token>

      <v1:type>Virtual</v1:type>

    </v1:GetDocumentFolders>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

```

```

    <GetDocumentFoldersResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentFoldersResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

          <DocumentFolder>

            <Comment/>

            <CreatedByPersonName>Administrator,
System</CreatedByPersonName>

            <FolderCategoryID>2</FolderCategoryID>

            <FolderCategoryName>Default</FolderCategoryName>

            <FolderStatus>Private</FolderStatus>

            <FolderType>Virtual</FolderType>

            <ID>11</ID>

            <LastModifiedDate>27/04/2012 08:54</LastModifiedDate>

            <Name>Test Folder</Name>

            <RegisterViewType>2</RegisterViewType>

          </DocumentFolder>

        </GetDocumentFoldersResult>

      </GetDocumentFoldersResponse>

    </s:Body>

  </s:Envelope>

```

GetDocumentFoldersByCategory

Description: Enables a user to retrieve document folders by folder type and folder category.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentFoldersByCategory`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

```

```

<s:Body>

  <v1:GetDocumentFoldersByCategory>

    <v1:token>AUTHENTICATION TOKEN</v1:token>

    <v1:type>Virtual</v1:type>

    <v1:categoryID>1</v1:categoryID>

  </v1:GetDocumentFoldersByCategory>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentFoldersByCategoryResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentFoldersByCategoryResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
        <DocumentFolder>

          ...DocumentFolder Result Goes here... (See Sample Response to
          GetDocumentFolders for message format)

        </DocumentFolder>

      </GetDocumentFoldersByCategoryResult>

    </GetDocumentFoldersByCategoryResponse>

  </s:Body>

</s:Envelope>

```

GetDocumentFolderByID

Description: Enables a user to retrieve a document folder by its ID.

Endpoint: <http://<servicepath>/Services/Document.svc/GetDocumentFolderByID>

Availability: Q-Pulse Web Services API version 02.000.580.008 and above

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentFolderByID>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:folderID>11</v1:folderID>

    </v1:GetDocumentFolderByID>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentFoldersByCategoryResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentFoldersByCategoryResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
        <DocumentFolder>

          ...DocumentFolder Result Goes here... (See Sample Response to
          GetDocumentFolders for message format)

        </DocumentFolder>

      </GetDocumentFolderByIDResult>

    </GetDocumentFolderByIDResponse>

  </s:Body>

</s:Envelope>

```

GetDocumentsByFolder

Description: Enables a user to retrieve a list of full document descriptions contained in a particular document folder by supplying a folder ID.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentsByFolder`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentsByFolder>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:folderID>11</v1:folderID>

    </v1:GetDocumentsByFolder>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentsByFolderResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentsByFolderResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Document>

          ...Document Result Goes here... (See Sample Response to
          GetDocument for message format)

        </Document>

        <Document>

          ...Document Result Goes here... (See Sample Response to
          GetDocument for message format)
```

```

        </Document>

    </GetDocumentsByFolderResult>

</GetDocumentsByFolderResponse>

</s:Body>

</s:Envelope>

```

GetDocumentIDsByFolder

Description: GetDocumentIDsByFolder is a lighter-weight alternative to GetDocumentsByFolder. Where GetDocumentsByFolder returns full descriptions of found records, GetDocumentIDsByFolder returns only the Document Numbers. You can then retrieve the full details using GetDocument.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentIDsByFolder`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentIDsByFolder>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:folderID>11</v1:folderID>

    </v1:GetDocumentIDsByFolder>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentIDsByFolderResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

```

```

    <GetDocumentIDsByFolderResult
      xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
      xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

      <a:int>34</a:int>

      <a:int>40</a:int>

      <a:int>32</a:int>

    </GetDocumentIDsByFolderResult>

  </GetDocumentIDsByFolderResponse>

</s:Body>

</s:Envelope>

```

GetDocumentSummariesByFolder

Description: Enables a user to retrieve a list of document summaries contained in a particular document folder by supplying a folder ID.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentSummariesByFolder`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentSummariesByFolder>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:folderID>11</v1:folderID>

    </v1:GetDocumentSummariesByFolder>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentSummariesByFolderResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentSummariesByFolderResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <DocumentSummary>

          ...DocumentSummary Result Goes here... (See Sample Response to
          GetDocumentSummaries for message format)

        </DocumentSummary>

        <DocumentSummary>

          ...DocumentSummary Result Goes here... (See Sample Response to
          GetDocumentSummaries for message format)

        </DocumentSummary>

      </GetDocumentSummariesByFolderResult>

    </GetDocumentSummariesByFolderResponse>

  </s:Body>
</s:Envelope>

```

GetWorkloads

Description: Enables a user to retrieve a list of Document based workloads. NOTE: Only the workload type OutstandingDistributionAcknowledgments is currently supported.

Endpoint: http://<servicepath>/Services/Document.svc/GetWorkloads

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

```

```

    <v1:GetWorkloads>

        <v1:token>AUTHENTICATION TOKEN</v1:token>

    </v1:GetWorkloads>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <GetWorkloadsResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <GetWorkloadsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">

                <DocumentWorkload>

                    <Count>0</Count>

                    <Description>Documents to be acknowledged</Description>

                    <FriendlyName>To Acknowledge</FriendlyName>

                <WorkloadType>OutstandingDistributionAcknowledgments</WorkloadType>

                </DocumentWorkload>

            </GetWorkloadsResult>

        </GetWorkloadsResponse>

    </s:Body>

</s:Envelope>

```

GetDocumentsByWorkload

Description: Enables a user to retrieve a list of full document descriptions that are part of a workload by supplying a workload type.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentsByWorkload`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentsByWorkload>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:workload>OutstandingDistributionAcknowledgments
    </v1:workload>

    </v1:GetDocumentsByWorkload>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentsByWorkloadResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentsByWorkloadResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>

        <Document>

          ...Document Result Goes here... (See Sample Response to
          GetDocument for message format)

        </Document>

        <Document>

          ...Document Result Goes here... (See Sample Response to
          GetDocument for message format)

        </Document>

      </GetDocumentsByWorkloadResult>

    </GetDocumentsByWorkloadResponse>

  </s:Body>

```

```
</s:Envelope>
```

GetDocumentIDsByWorkload

Description: GetDocumentIDsByWorkload is a lighter-weight alternative to GetDocumentsByWorkload. Where GetDocumentsByWorkload returns full descriptions of found records, GetDocumentIDsByWorkload returns only the Document Numbers. You can then retrieve the full details using GetDocument or acknowledge the document using AcknowledgeDocument.

Endpoint: `http://<servicepath>/Services/Document.svc/GetDocumentIDsByWorkload`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentIDsByWorkload>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:workload>OutstandingDistributionAcknowledgments</v1:workload>

    </v1:GetDocumentIDsByWorkload>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetDocumentIDsByWorkloadResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentIDsByWorkloadResult
        xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>

        <a:int>34</a:int>

        <a:int>40</a:int>

      </GetDocumentIDsByWorkloadResult>

    </GetDocumentIDsByWorkloadResponse>

  </s:Body>

</s:Envelope>
```

```

        <a:int>32</a:int>

      </GetDocumentIDsByWorkloadResult>

    </GetDocumentIDsByWorkloadResponse>

  </s:Body>
</s:Envelope>

```

GetDocumentSummariesByWorkload

Description: Enables a user to retrieve a list of document summaries that are part of a workload by supplying a workload type.

Endpoint:
<http://<servicepath>/Services/Document.svc/GetDocumentSummariesByWorkload>

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetDocumentSummariesByWorkload>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:workload>OutstandingDistributionAcknowledgments</v1:workload>

    </v1:GetDocumentSummariesByWorkload>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

```

```

    <GetDocumentSummariesByWorkloadResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetDocumentSummariesByWorkloadResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
        <DocumentSummary>

          ...DocumentSummary Result Goes here... (See Sample Response to
          GetDocumentSummaries for message format)

        </DocumentSummary>

        <DocumentSummary>

          ...DocumentSummary Result Goes here... (See Sample Response to
          GetDocumentSummaries for message format)

        </DocumentSummary>

      </GetDocumentSummariesByWorkloadResult>

    </GetDocumentSummariesByWorkloadResponse>

  </s:Body>
</s:Envelope>

```

CreateNewDocument

Description: Enables a user to create a new Document record

Endpoint: `http://<servicepath>/Services/Document.svc/CreateNewDocument`

Mandatory fields: Number, DocumentTypeID

Fields that will be set on the business object when creating a Document (other fields can be specified in the XML but will be ignored): ActiveDateTime, AuthorPersonID, ChangeDetails, CreatedByPersonID, DoCheckIn, DocumentTypeID, IsOnlyVisibleToAuthor, Keywords, Number, OwnerPersonID, PersonTrainingEventRenewBy, RenewPeronTrainingEvents, RevisionReference, Status, Title, UserDefinedField1Description, UserDefinedField1ID, UserDefinedField2Description, UserDefinedField2ID, UserDefinedField3Description, UserDefinedField3ID, UserDefinedField4Description, UserDefinedField4ID, UserDefinedField5PersonID, UserDefinedField5PersonName

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:arr="http://schemas.microsoft.com/2003/10/Serialization/Arrays">

  <s:Header/>

  <s:Body>

    <v1:CreateNewDocument>

      <v1:token> AUTHENTICATION TOKEN</v1:token>

      <v1:newDocument>

        <v1:ApprovalNotes/>

        <v1:Approvers/>

        <v1:ChangeRequestsCollection/>

        <v1:CopyHoldersCollection/>

        <v1:DocumentNotesCollection/>

        <v1:DocumentRelatedDepartmentsCollection/>

        <v1:DocumentRelatedDocumentsIDs/>

        <v1:DocumentRelatedProcessesCollection/>

        <v1:DocumentRelatedStandardsCollection/>

        <v1:DocumentReviewsCollection/>

        <v1:DocumentTypeID>1</v1:DocumentTypeID>

        <v1:OwnerPersonID>3</v1:OwnerPersonID>

        <v1:RevisionIDs/>

        <v1:Status>Active</v1:Status>

        <v1:Title>New Document</v1:Title>

      </v1:newDocument>

    </v1:CreateNewDocument>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <CreateNewDocumentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <CreateNewDocumentResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        ...Document Result Goes here... (See Sample Response to GetDocument
        for message format)

      </CreateNewDocumentResult>

    </CreateNewDocumentResponse>

  </s:Body>

</s:Envelope>

```

AddNote

Description: Enables a user to add a note to an existing Document by supplying its document ID and the note string to add.

Endpoint: `http://<servicepath>/Services/Document.svc/AddNote`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <AddNote xmlns="http://tempuri.org/">

      <token>AUTHENTICATION TOKEN</token>

    </AddNote>

  </s:Body>

</s:Envelope>

```

```

        <v1:token>AUTHENTICATION TOKEN</v1:token>

        <v1:documentID>2</v1:documentID>

        <v1:note>This is a note</v1:note>

    </v1:AddNote>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <AddNoteResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <AddNoteResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
                ...Document Result Goes here... (See Sample Response to
GetDocument for message format)
            </AddNoteResult>

        </AddNoteResponse>

    </s:Body>

</s:Envelope>

```

AcknowledgeDocument

Description: Enables a user to acknowledge a document that is part of a users OutstandingDistributionAcknowledgments workload. If Digital Signatures are enabled within Q-Pulse then the users userName and password parameters must be passed

Endpoint: http://<servicepath>/Services/Document.svc/AcknowledgeDocument

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:AcknowledgeDocument>

      <v1:token> AUTHENTICATION TOKEN</v1:token>

      <v1:documentID>1</v1:documentID>

      <v1:userName>Q-PULSE USER</v1:userName>

      <v1:password>Q-PULSE PASSWORD</v1:password>

    </v1:AcknowledgeDocument>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <AcknowledgeDocumentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <AcknowledgeDocumentResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)
      </AcknowledgeDocumentResult>

    </AcknowledgeDocumentResponse>

  </s:Body>

</s:Envelope>

```

ReviseDocumentToActive

Description: Enables a user to revise a document to a new revision

Endpoint: <http://<servicepath>/Services/Document.svc/ReviseDocumentToActive>

Availability: Q-Pulse Web Services API version 02.000.580.006 and above

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:ReviseDocumentToActive>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:documentID>1</v1:documentID>

      <v1:newRevisionReference>2</v1:newRevisionReference>

      <v1:activeDate>2011-12-19T00:00:00</v1:activeDate>

      <v1:changeDetails>New updates to revision</v1:changeDetails>

      <v1:deactivatedRevisionStatus>Obsolete</v1:deactivatedRevisionStatus>

      <v1:keepCopyHoldersFromCurrentRevision>>false</v1:keepCopyHoldersFromCu
        rrentRevision>

    </v1:ReviseDocumentToActive>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <ReviseDocumentToActiveResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <ReviseDocumentToActiveResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```

```

        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)

    </ReviseDocumentToActiveResult>

</ReviseDocumentToActiveResponse>

</s:Body>

</s:Envelope>

```

AddUrlAttachmentItem

Description: Enables a user to add a URLAttachment item to an attachment of a particular document's revision. If no existing attachment exists for the revision an attachment is created with a name matching the document number.

Endpoint: `http://<servicepath>/Services/Document.svc/AddUrlAttachmentItem`

Availability: Q-Pulse Web Services API version 02.000.580.006 and above

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:AddUrlAttachmentItem>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:revisionID>4</v1:revisionID>

      <v1:newURL>www.google.com</v1:newURL>

    </v1:AddUrlAttachmentItem>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <AddUrlAttachmentItemResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/"
      <AddUrlAttachmentItemResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)
      </AddUrlAttachmentItemResult>
    </AddUrlAttachmentItemResponse>
  </s:Body>
</s:Envelope>

```

DistributeActiveDocument

Description: Enables a user to distribute an active document to its holders.

Endpoint: `http://<servicepath>/Services/Document.svc/DistributeActiveDocument`

Availability: Q-Pulse Web Services API version 02.000.580.006 and above

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  <s:Header/>
  <s:Body>
    <v1:DistributeActiveDocument>
      <v1:token>AUTHENTICATION TOKEN</v1:token>
      <v1:documentID>1</v1:documentID>
    </v1:DistributeActiveDocument>
  </s:Body>
</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  <s:Body>
    <DistributeActiveDocumentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/"
      <DistributeActiveDocumentResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)
      </DistributeActiveDocumentResult>
    </DistributeActiveDocumentResponse>
  </s:Body>
</s:Envelope>

```

DeactivateDocument

Description: Enables a user to deactivate an active document to Obsolete or Inactive revision status.

Endpoint: `http://<servicepath>/Services/Document.svc/DeactivateDocument`

Availability: Q-Pulse Web Services API version 02.000.580.006 and above

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  <s:Header/>
  <s:Body>
    <v1:DeactivateDocument>
      <v1:token>AUTHENTICATION TOKEN</v1:token>
      <v1:documentID>3</v1:documentID>
      <v1:status>Inactive</v1:status>
    </v1:DeactivateDocument>
  </s:Body>
</s:Envelope>

```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <DeactivateDocumentResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <DeactivateDocumentResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        ...Document Result Goes here... (See Sample Response to
        GetDocument for message format)

      </DeactivateDocumentResult>

    </DeactivateDocumentResponse>

  </s:Body>

</s:Envelope>
```

NonConformance Service

Description: Provides access to Create and Read functionality related to the Q-Pulse NonConformance Module.

Endpoint: `http://<servicepath>/Services/NonConformance.svc`

GetNonConformance

Description: Enables a user to view an existing NonConformance record.

Endpoint: `http://<servicepath>/Services/NonConformance.svc/GetNonConformance`

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:GetNonConformance>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:nonConformanceID>1</v1:nonConformanceID>

    </v1:GetNonConformance>

  </s:Body>

</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

  <s:Body>

    <GetNonConformanceResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetNonConformanceResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

        <AssociatedDocumentDescription/>

      </GetNonConformanceResult>

    </GetNonConformanceResponse>

  </s:Body>

</s:Envelope>
```

```
<AssociatedDocumentID>-1</AssociatedDocumentID>

<Attachments/>

<AuditID>-1</AuditID>

<AuditStatus>-1</AuditStatus>

<AuditTitle/>

<BatchesCollection/>

<BrokenRulesCollection/>

<CanClose>>false</CanClose>

<Closed>>false</Closed>

<ClosedByPersonID>-1</ClosedByPersonID>

<ClosedByPersonName/>

<ClosedDate/>

<Cost1>0.0000</Cost1>

<Cost1String>0.00</Cost1String>

<Cost2>0.0000</Cost2>

<Cost2String>0.00</Cost2String>

<Cost3>0.0000</Cost3>

<Cost3String>0.00</Cost3String>

<Cost4>0.0000</Cost4>

<Cost4String>0.00</Cost4String>

<Cost5>0.0000</Cost5>

<Cost5String>0.00</Cost5String>

<CreatedByPersonID>1</CreatedByPersonID>

<CurrencyID>-1</CurrencyID>

<CurrencyName/>

<CurrencySymbol/>
```

```

<Details>sdafdsaf dsfd sfa</Details>

<DurationUnit>Hours</DurationUnit>

<DurationUnitName>Hour/s</DurationUnitName>

<EquipmentItems/>

<FaultCategoryID>-1</FaultCategoryID>

<FaultCategoryPath/>

<FriendlyName>NC1</FriendlyName>

<ID>1</ID>

<IncidentSourceTypeID>4</IncidentSourceTypeID>

<IncidentSourceTypePath>Employee
Suggestion</IncidentSourceTypePath>

<IncidentTemplateID>-2</IncidentTemplateID>

<IsOverdue>>false</IsOverdue>

<Keywords/>

<NonConformanceIncidentStageID>-
1</NonConformanceIncidentStageID>

<Notes/>

<Number>NC1</Number>

<OccurrenceIncidentID>-1</OccurrenceIncidentID>

<OccurrenceIncidentStageDescription/>

<OccurrenceIncidentStageID>-1</OccurrenceIncidentStageID>

<OwnerPersonID>3</OwnerPersonID>

<OwnerPersonName>Guy, Test</OwnerPersonName>

<ParentID>-1</ParentID>

<RaisedAgainst>-1</RaisedAgainst>

<RaisedAgainstAreaOfStandardID>-
1</RaisedAgainstAreaOfStandardID>

<RaisedAgainstAreaOfStandardPath/>

```

```

    <RaisedAgainstDepartmentID>-1</RaisedAgainstDepartmentID>

    <RaisedAgainstDepartmentPath/>

    <RaisedAgainstName/>

    <RaisedAgainstPersonID>-1</RaisedAgainstPersonID>

    <RaisedAgainstPersonName/>

    <RaisedAgainstProcessDescription/>

    <RaisedAgainstProcessID>-1</RaisedAgainstProcessID>

    <RaisedAgainstProductServiceDescription/>

    <RaisedAgainstProductServiceID>-
1</RaisedAgainstProductServiceID>

    <RaisedAgainstRevisionID>-1</RaisedAgainstRevisionID>

    <RaisedAgainstRevisionTitle/>

    <RaisedAgainstSupplierID>-1</RaisedAgainstSupplierID>

    <RaisedAgainstSupplierPath/>

    <RaisedBy>-1</RaisedBy>

    <RaisedByExternal>-1;-1</RaisedByExternal>

    <RaisedDate>03/05/2012</RaisedDate>

    <RaisedDateTime>2012-05-03T00:00:00</RaisedDateTime>

    <RaisingContactID>-1</RaisingContactID>

    <RaisingContactName/>

    <RaisingCustomerID>-1</RaisingCustomerID>

    <RaisingCustomerPath/>

    <RaisingName/>

    <RaisingPersonID>-1</RaisingPersonID>

    <RaisingPersonName/>

    <RaisingPersonOrCustomer/>

    <ReactionTime>0</ReactionTime>

```

```

    <RecordedByPersonID>1</RecordedByPersonID>

    <RecordedByPersonName>Administrator,
System</RecordedByPersonName>

    <ResolutionID>-1</ResolutionID>

    <ResolutionPath/>

    <RevisionDescription/>

    <RootCauseID>-1</RootCauseID>

    <RootCausePath/>

    <SeverityID>-1</SeverityID>

    <SeverityName/>

    <StatusID>

        <IsSet>true</IsSet>

        <Value>1</Value>

    </StatusID>

    <StatusName>Open</StatusName>

    <TargetDate>07/05/2012</TargetDate>

    <TargetDateTime>2012-05-07T00:00:00</TargetDateTime>

    <Time1>0</Time1>

    <Time2>0</Time2>

    <Time3>0</Time3>

    <Time4>0</Time4>

    <Time5>0</Time5>

    <TotalCost>0.0000</TotalCost>

    <TotalCostString>0.00</TotalCostString>

    <TotalTime>0</TotalTime>

    <UserDefinedField1ID>-1</UserDefinedField1ID>

    <UserDefinedField1Path/>

```

```

    <UserDefinedField2ID>-1</UserDefinedField2ID>

    <UserDefinedField2Path/>

    <UserDefinedField3/>

    <UserDefinedField3DateTime>0001-01-
01T00:00:00</UserDefinedField3DateTime>

    <UserDefinedField4/>

    <UserDefinedField4DateTime>0001-01-
01T00:00:00</UserDefinedField4DateTime>

    <UserDefinedField5PersonID>-1</UserDefinedField5PersonID>

    <UserDefinedField5PersonName/>

    <UserDefinedField6PersonID>-1</UserDefinedField6PersonID>

    <UserDefinedField6PersonName/>

    <UserDefinedField7/>

    <UserDefinedField8/>

    <nonConformanceStages>

      <NonConformanceStage>

        <CanClose>true</CanClose>

        <CanDelete>true</CanDelete>

        <ClosedByPersonID>-1</ClosedByPersonID>

        <ClosedByPersonName/>

        <CreatedByPersonID>1</CreatedByPersonID>

        <Currency/>

        <DisplayName>Corrective Action</DisplayName>

        <FriendlyName>Corrective Action</FriendlyName>

        <ID>1</ID>

        <NonConformanceID>1</NonConformanceID>

      <NonConformanceStageTypeID>3</NonConformanceStageTypeID>

```

```

        <NonConformanceStageTypeName>Corrective
Action</NonConformanceStageTypeName>

        <OwnerPersonID>-1</OwnerPersonID>

        <OwnerPersonName/>

        <ParentID>1</ParentID>

        <ReactionTime>0</ReactionTime>

        <SortOrder>1</SortOrder>

        <Summary/>

        <TargetDateTime>2012-05-06T00:00:00</TargetDateTime>

        <TimeForStage>0</TimeForStage>

        <WorkflowOrder>1</WorkflowOrder>

        <WorkflowStatus>Started</WorkflowStatus>

        <nonConformanceStageActions/>

</NonConformanceStage>

<NonConformanceStage>

        <CanClose>true</CanClose>

        <CanDelete>true</CanDelete>

        <ClosedByPersonID>-1</ClosedByPersonID>

        <ClosedByPersonName/>

        <CreatedByPersonID>1</CreatedByPersonID>

        <Currency/>

        <DisplayName>Follow Up</DisplayName>

        <FriendlyName>Follow Up</FriendlyName>

        <ID>2</ID>

        <NonConformanceID>1</NonConformanceID>

        <NonConformanceStageTypeID>5</NonConformanceStageTypeID>

```

```

        <NonConformanceStageTypeName>Follow
Up</NonConformanceStageTypeName>

        <OwnerPersonID>-1</OwnerPersonID>

        <OwnerPersonName/>

        <ParentID>1</ParentID>

        <ReactionTime>0</ReactionTime>

        <SortOrder>2</SortOrder>

        <Summary>Follow Up required</Summary>

        <TargetDateTime>2012-05-07T00:00:00</TargetDateTime>

        <TimeForStage>0</TimeForStage>

        <WorkflowOrder>1</WorkflowOrder>

        <WorkflowStatus>Started</WorkflowStatus>

        <nonConformanceStageActions/>

    </NonConformanceStage>

</nonConformanceStages>

</GetNonConformanceResult>

</GetNonConformanceResponse>

</s:Body>

</s:Envelope>

```

GetNonConformances

- Description:** Enables a user to search for NonConformancerecords by Severity, Source type, Status, Raised Date, Target Date and any keyword(s) associated with the NonConformance record.
- Search criteria:** The query object that you pass to this operation allows you to search against the date a NonConformance was raised, the Severity, the Source Type, the Status, the Target date and Keywords (which performs a 'contains' search against the Number, Details and Keywords of NonConformances).
- Endpoint:** `http://<servicepath>/Services/NonConformance.svc/GetNonConformances`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:GetNonConformances>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:query>

        <v1:Keywords i:nil="true"/>

        <v1:RaisedDateCondition>

          <v1:EndDate i:nil="true"/>

          <v1:Operator>Today</v1:Operator>

          <v1:StartDate i:nil="true"/>

        </v1:RaisedDateCondition>

        <v1:SeverityID>4</v1:SeverityID>

        <v1:SourceTypeID>7</v1:SourceTypeID>

        <v1:StatusID>1</v1:StatusID>

        <v1:TargetDateCondition i:nil="true"/>

      </v1:query>

    </v1:GetNonConformances>

  </s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"

  <s:Body>

```

```

    <GetNonConformancesResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

      <GetNonConformancesResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>
        <NonConformance>

          ...NonConformance Result Goes here... (See Sample Response to
          GetNonConformance for message format)

        </NonConformance>

      </GetNonConformancesResult>

    </GetNonConformancesResponse>

  </s:Body>
</s:Envelope>

```

GetNonConformanceIDs

Description: GetNonConformanceIDs is a lighter weight alternative to GetNonConformances. Where GetNonConformances returns full descriptions of found records, GetNonConformanceIDs returns only the NonConformance record numbers. You can then retrieve the full details using GetNonConformance.

Search criteria: The query object that you pass to this operation allows you to search against the date a NonConformance was raised, the Severity, the Source Type, the Status, the Target date and Keywords (which performs a 'contains' search against the Number, Details and Keywords of NonConformances).

Endpoint: `http://<servicepath>/Services/NonConformance.svc/GetNonConformanceIDs`

Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

  <s:Header/>

  <s:Body>

    <v1:GetNonConformanceIDs>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:query>

        <v1:Keywords i:nil="true"/>

      </v1:query>

    </v1:GetNonConformanceIDs>

  </s:Body>

</s:Envelope>

```

```

        <v1:RaisedDateCondition>

            <v1:EndDate i:nil="true"/>

            <v1:Operator>Today</v1:Operator>

            <v1:StartDate i:nil="true"/>

        </v1:RaisedDateCondition>

        <v1:SeverityID>4</v1:SeverityID>

        <v1:SourceTypeID>7</v1:SourceTypeID>

        <v1:StatusID>1</v1:StatusID>

        <v1:TargetDateCondition i:nil="true"/>

    </v1:query>

</v1:GetNonConformanceIDs>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <GetNonConformanceIDsResponse
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <GetNonConformanceIDsResult
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance"/>

                <a:int>1</a:int>

            </GetNonConformanceIDsResult>

        </GetNonConformanceIDsResponse>

    </s:Body>

</s:Envelope>

```

CreateNewNonConformance

Description: Enables a user to create a new Non-conformance record

Endpoint: `http://<servicepath>/Services/NonConformance.svc/CreateNewNonConformance`

Mandatory fields: IncidentSourceTypeID

Fields that will be set on the business object when creating a NonConformance (other fields can be specified in the XML but will be ignored): StatusName, RaisedDate, TargetDate, OwnerPersonName, Details, RaisingPersonName, Keywords, IncidentSourceTypeID, RaisedAgainstDepartmentPath and SeverityName.

Sample Request:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:v1="http://www.qpulse.com/QPulseWebServices/v1.2/">

  <s:Header/>

  <s:Body>

    <v1:CreateNewNonConformance>

      <v1:token>AUTHENTICATION TOKEN</v1:token>

      <v1:newNonConformance>

        <v1:Attachments/>

        <v1:BatchesCollection/>

        <v1:Details>Something happened</v1:Details>

        <v1:EquipmentItems/>

        <v1:IncidentSourceTypeID>2</v1:IncidentSourceTypeID>

        <v1:Notes/>

        <v1:Number>NC1</v1:Number>

        <v1:OwnerPersonID>2</v1:OwnerPersonID>

        <v1:RaisingPersonID>2</v1:RaisingPersonID>

        <v1:SeverityID>4</v1:SeverityID>

        <v1:StatusID>
```

```

        <v1:IsSet>true</v1:IsSet>

        <v1:Value>1</v1:Value>

    </v1:StatusID>

    <v1:nonConformanceStages>

        <v1:NonConformanceStage>

            <v1:NonConformanceStageTypeID>3</v1:NonConformanceStageTypeID>

            <v1:OwnerPersonID>1</v1:OwnerPersonID>

            <v1:Summary>Do this</v1:Summary>

            <v1:nonConformanceStageActions>

                <v1:NonConformanceStageAction>

                    <v1:Details>Action</v1:Details>

                </v1:NonConformanceStageAction>

            </v1:nonConformanceStageActions>

        </v1:NonConformanceStage>

    </v1:nonConformanceStages>

</v1:newNonConformance>

</v1:CreateNewNonConformance>

</s:Body>

</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <CreateNewNonConformanceResponse
            xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">

            <CreateNewNonConformanceResult
                xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

```

```
    ...NonConformance Result Goes here... (See Sample Response to  
    GetNonConformance for message format  
  
    </CreateNewNonConformanceResult>  
  
    </CreateNewNonConformanceResponse>  
  
</s:Body>  
</s:Envelope>
```

Exceptions

Application Level Exceptions

The Q-Pulse Web Service has been designed to return well-formed `ServiceFaultExceptions` when something undesirable happens in the Web Service call lifetime. When an exception is raised, it is assigned a message and a Fault Code. The current possible FaultCodes are:

- `SessionInvalid` - the Q-Pulse session for the current user is invalid
- `AuthenticationFailed` – an attempt to authenticate against Q-Pulse has failed
- `ConnectionsNotAvailable` – the Q-Pulse service that provides the list of available Databases is not available
- `ReferenceListTypeNotFound` – Q-Pulse was not able to resolve the requested Occurrence ReferenceList
- `ManagedListTypeNotFound` – Q-Pulse was not able to resolve the requested Managed List
- `OperationNotSupported` – the requested method is not supported without further configuration
- `InvalidDatabase` - the requested database does not exist or is currently unavailable
- `ItemNotFoundInDatabase` - the item requested could not be found in the database
- `CannotConnectToService` – the service is not available
- `InvalidOperation` - the operation requested could not be completed due to invalid state on the server
- `UserLockedOut` - the user has been locked out due to too many failed login attempts
- `NullParameterReceived` - the parameter to the Web Service endpoint has been received as null
- `UnhandledException` – an unhandled exception has occurred in the Web Service
- `XMLDeserializationException` – an exception has occurred while trying to deserialize the request XML

The unhandled exception fault code is a special case where the exception encountered was not anticipated. All the other fault codes specifically identify the reason for the exception.

Here is an example of a typical exception message:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <s:Fault>
      <faultcode>s:Client</faultcode>
      <faultstring xml:lang="en-GB">Invalid Database</faultstring>
      <detail>
```

```

        <ServiceFault
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

            <FaultCode>InvalidDatabase</FaultCode>

            <Message>Invalid Database</Message>

        </ServiceFault>

    </detail>

</s:Fault>

</s:Body>

</s:Envelope>

```

This is what an unhandled exception response message looks like:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">

    <s:Body>

        <s:Fault>

            <faultcode>s:Client</faultcode>

            <faultstring xml:lang="en-GB">Unhandled exception. Please check
server logs.</faultstring>

            <detail>

                <ServiceFault
xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">

                    <FaultCode>UnhandledException</FaultCode>

                    <Message>Unhandled exception. Please check server
logs.</Message>

                </ServiceFault>

            </detail>

        </s:Fault>

    </s:Body>

```

```
</s:Envelope>
```

Broken Rules

While attempting to create one of the Business objects via the Web Service (e.g. Occurrence, Document, Audit or Non-conformance), if there is problem saving the business object, a business object will be returned with a negative ID and a non-empty Broken Rules collection, like so:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <RaiseNewOccurrenceResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.2/">
      <RaiseNewOccurrenceResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <Attachments/>
        <BrokenRules>
          <BrokenRule>
            <Description>Type must be specified</Description>
            <Severity>Error</Severity>
          </BrokenRule>
        </BrokenRules>
        <DataFields/>
        <ID>-3</ID>
        <IncidentNumber/>
        <IncidentReportSentDate>0001-01-01T00:00:00</IncidentReportSentDate>
        <IncidentStatusName/><IncidentTitle/>
        <Notes/>
        <RecordedByPersonName/>
        <RecordedDate>2009-08-26T13:24:00</RecordedDate>
        <ReportTypeName/>
      </RaiseNewOccurrenceResult>
    </RaiseNewOccurrenceResponse>
  </s:Body>
</s:Envelope>
```

```

    <ReportedByPersonName/>

    <ReportedDate>2009-08-26T13:24:00</ReportedDate>

  </RaiseNewOccurrenceResult>

</RaiseNewOccurrenceResponse>

</s:Body>

</s:Envelope>

```

The broken rules returned, can be used to determine why the saving of the Business object failed.

Unable to authenticate a User

If you are unable to authenticate a User, make sure you are using the correct username, password and database (these are all case-sensitive). If the username or password is invalid, the message returned will be 'User could not be authenticated'. If the database was wrong the message will be 'Invalid Database'. The possible database names can be obtained via the Core.AvailableConnections operation.

If you are sure the credentials and database name are correct, try investigating the following:

- The server license is still valid (you can determine this using Core.HasServerLicenseExpired)
- The user has adequate login permissions ('Enable login' check-box in Security -> People section of the Admin module)
- The user can log into Q-Pulse via the Windows client

Debugging Non-Application Exceptions

Communication Problems

The first step to debugging any problem with the Q-Pulse Web Services is to make sure you can connect to them. The easiest way to do this is by making sure you can request the WSDL of one of the service endpoints. This can be achieved by entering the URL into a Web browser e.g. <http://servername/QPulse5WebServices/Services/Core.svc?wsdl>.

If there is no meaningful response (you should receive back XML), make sure you used the correct URL. If you are sure you used the correct URL, try investigating the following:

- Ping the server to rule out network problems (only works if server is configured to respond to pings)
- Check that any firewalls are correctly configured and traffic is passing through successfully
- Make sure that the Q-Pulse Web Services can be accessed locally from the server hosting them

XML problems

If you can communicate with the Web Service (i.e. retrieve WSDL) but don't get the expected response, it's possible that the XML request message you are sending is not being understood by the Web Service. Here is a sample of the fault codes returned for various common XML problems.

Description	Example	FaultCode
Parameter missing	<userName> not declared in request message	NullParameterReceived
Wrong case	<username> used instead of <userName>	NullParameterReceived
Misspelt element	<user> used instead of <userName>	NullParameterReceived
Namespace not applied to parameter element	While defining the <userName> element there is no namespace applied e.g. <ns:userName>	XmlDeserializationException
Mismatched tags	<usrName> </userName>	XmlDeserializationException
Namespace not applied to operation element	While defining the <Authenticate> element there is no namespace applied e.g. <ns:Authenticate>	XmlDeserializationException

Other Considerations

If seemingly strange Application Exceptions are being received. It may be that the Q-Pulse Server isn't running properly (if at all), in which case it is worth restarting it using the Windows Service administration tool.

Other Debugging Tools

Windows Event Viewer

Windows Event Viewer is available in Windows by default. It displays any errors that have been logged to the Even Log. There are several places it's worth viewing the log to check if there is any meaningful output: the server hosting the Q-Pulse Web Services, the server hosting the Q-Pulser Server and the server hosting the database.

DebugView

DebugView is produced by Microsoft and is available for download from their Technet website. It captures any debug output from applications currently running on the local machine so can be useful for finding out about any problems that are not surface anywhere else. You won't be able to use DebugView to get any debug output from the Q-Pulse Web Services as any debug information will have been removed during the build process.



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.