



---

**Q-Pulse Web Service APIs  
Technical Reference Guide**

# Contents

---

<b>Introduction</b>	<b>4</b>
<b>Core Service</b>	<b>5</b>
AvailableConnections	5
Authenticate and AuthenticateWindowsID	6
GetManagedList	8
SearchManagedList	9
GetServerAuthenticationType	10
HasServerLicenseExpired	10
GetLoggedInPersonDetails	11
<b>Occurrence Service</b>	<b>12</b>
ListAvailableReportTypes	12
GetDataFieldsForReportType	13
GetReferenceList	14
CreateNewOccurrence	15
GetOccurrence	17
GetOccurrences	18
GetOccurrenceIDs	20
<b>Incident Service</b>	<b>21</b>
GetIncident	21
GetIncidents	24
GetIncidentIDs	25
<b>Attachment Service</b>	<b>26</b>
PostAttachment	27
GetAttachment	28

<b>Audit Service.....</b>	<b>29</b>
GetAudit .....	29
GetAudits .....	33
GetAuditIDs.....	34
CreateNewAudit .....	35
<b>Document Service .....</b>	<b>37</b>
GetDocument .....	37
GetDocuments .....	39
GetDocumentIDs.....	40
CreateNewDocument .....	41
<b>NonConformance Service.....</b>	<b>43</b>
GetNonConformance .....	43
GetNonConformances.....	46
GetNonConformanceIDs .....	47
CreateNewNonConformance .....	48
<b>Exceptions .....</b>	<b>50</b>
Application Level Exceptions.....	50
Broken Rules.....	51
Unable to authenticate a User.....	52
<b>Debugging Non-Application Exceptions .....</b>	<b>53</b>
Communication Problems .....	53
XML problems .....	54
Other Considerations .....	54
Other Debugging Tools .....	54

## 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](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.

## 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`

### 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/">
  <s:Header>
    <Action s:mustUnderstand="1"
      xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://
      /www.qpulse.com/QPulseWebServices/v1.0/ICore/AvailableConnections</Act
      ion>
    </s:Header>
    <s:Body>
      <AvailableConnections
        xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/" />
      </s:Body>
    </s:Envelope>
```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <xsd:string>Q-PULSE DATABASE NAME</xsd:String>
  </s:Body>
</s:Envelope>
```

## Authenticate and AuthenticateWindowsID

**Description:** Returns an authentication token that can be used to access services requiring authentication. Both 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>).

**Endpoints:** `http://<servicepath>/Services/Core.svc/Authenticate`

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

**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/">
  <s:Header>
    <Action s:mustUnderstand="1"
      xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://
      /www.qpulse.com/QPulseWebServices/v1.0/ICore/Authenticate</Action>
    </s:Header>
  <s:Body>
    <Authenticate xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <userName>Q-Pulse User Name</userName>
      <password>Q-Pulse Password</password>
      <database>Q-Pulse Database</database>
    </Authenticate>
  </s:Body>
</s:Envelope>
```

...and for Windows Authentication...

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header>
    <Action s:mustUnderstand="1"
      xmlns="http://schemas.microsoft.com/ws/2005/05/addressing/none">http://
      /www.qpulse.com/QPulseWebServices/v1.0/ICore/AuthenticateWindowsID</Ac
      tion>
    </s:Header>
```

```

<s:Body>
  <AuthenticateWindowsID
    xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
    <windowsUserName>WINDOWS_DOMAIN\Windows User Name</windowsUserName>
    <database>Q-PULSE DATABASE NAME</database>
  </AuthenticateWindowsID>
</s:Body>
</s:Envelope>

```

Sample Responses:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <AuthenticateResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <AuthenticateResult>AUTHENTICATION TOKEN</AuthenticateResult>
    </AuthenticateResponse>
  </s:Body>
</s:Envelope>

```

...and again for Windows Authentication...

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Header />
  <s:Body>
    <AuthenticateWindowsIDResponse
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
    <AuthenticateWindowsIDResult>AUTHENTICATION
      TOKEN</AuthenticateWindowsIDResult>
    </AuthenticateWindowsIDResponse>
  </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/">
  <s:Body>
    <GetManagedList
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHENTICATION TOKEN</token>
      <type>OccurrenceReportType</type>
    </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/v1.0/">
      <GetManagedListResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <IsHierarchy>false</IsHierarchy>
        <Items>
          <Items>
            <ManagedListItem>
              <FullPath>Flight Safety</FullPath>
              <ID>25</ID>
              <Name>Flight Safety</Name>
              <ParentID>-1</ParentID>
            </ManagedListItem>
            <ManagedListItem>
              <FullPath>Cabin Safety</FullPath>
              <ID>30</ID>
              <IsArchived>true</IsArchived>
              <Name>Cabin Safety</Name>
              <ParentID>-1</ParentID>
            </ManagedListItem>
            and so on.....
          </Items>
          <ListType>Person</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/">
  <s:Body>
<SearchManagedList xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
  <token>AUTHORISATION TOKEN</token>
  <type>Person</type>
  <searchString>Admin</searchString>
</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/v1.0/"
  <SearchManagedListResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
    <Items>
      <ManagedListItem>
        <FullPath>Administrator, System</FullPath>
        <ID>1</ID>
        <Name>Administrator, System</Name>
      </ManagedListItem>
    </Items>
  </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/">
  <s:Body>
    <GetServerAuthenticationType
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <dbName>DB_NAME</dbName>
    </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/v1.0/">

      <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/">
  <s:Body>
    <HasServerLicenseExpired
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
    </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/v1.0/">

      <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/">
  <s:Body>
    <GetLoggedInPersonDetails
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHENTICATION_TOKEN</token>
    </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/v1.0/">
      <GetLoggedInPersonDetailsResult
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <ID>1</ID>
        <Name>Administrator, System</Name>
      </GetLoggedInPersonDetailsResult>
    </GetLoggedInPersonDetailsResponse>
  </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/">
  <s:Body>
    <ListAvailableReportTypes
xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <databaseName>Q-Pulse</databaseName>
    </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/v1.0/">
      <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/">
  <s:Body>
    <GetDataFieldsForReportType
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <reportTypeName>Cabin Safety</reportTypeName>
      <databaseName>Q-PULSE DATABASE NAME</databaseName>
    </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/v1.0/">
      <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>
        </DataField>
      </GetDataFieldsForReportTypeResult>
    </GetDataFieldsForReportTypeResponse>
  </s:Body>
</s:Envelope>

```

```

        <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>

```

### 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/">
  <s:Body>
    <GetReferenceList
xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION TOKEN</token>
      <referenceListName>Cabin Activity</referenceListName>
      <databaseName>Q-PULSE DATABASE NAME</databaseName>
    </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/v1.0/">
      <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

## Sample Request:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CreateNewOccurrence
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION TOKEN</token>
      <newOccurrence xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <DataFields>
          <DataField>
            <DataType>String</DataType>
            <Name>Occurrence Description</Name>
            <Value>This is a test occurrence</Value>
          </DataField>
        </DataFields>
        <ReportTypeName>Cabin Safety</ReportTypeName>
        <ReportedByPersonName/>
        <Title>A Test Title</Title>
      </newOccurrence>
    </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/v1.0/">
      <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>
        </DataFields>
        <ID>15</ID>
        <IncidentNumber/>
        <IncidentReportSentDate>0001-01-
01T00:00:00</IncidentReportSentDate>
        <IncidentStatusName/>
        <IncidentTitle/>
        <Notes/>
        <Number>CCR14</Number>
      </CreateNewOccurrenceResult>
    </CreateNewOccurrenceResponse>
  </s:Body>
</s:Envelope>

```

```

        <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/">
  <s:Body>
    <GetOccurrence xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION TOKEN</token>
      <occurrenceID>18</occurrenceID>
    </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/v1.0/"
    <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>
    </GetOccurrenceResult>
  </s:Body>
</s:Envelope>

```

```

        <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. (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/">
  <s:Body>
    <GetOccurrences xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION TOKEN</token>
      <query xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <IsMandatory>false</IsMandatory>
        <Keywords>ASR</Keywords>
        <Number>OCC12</Number>
      </query>
    </GetOccurrences>
  </s:Body>
</s:Envelope>

```

```

    <RecordedByPersonName/>
    <ReportTypeName>Technical Safety</ReportTypeName>
    <ReportedOnOrAfter>2009-09-01T13:56:00</ReportedOnOrAfter>
    <ReportedOnOrBefore>2009-07-17T13:57:00</ReportedOnOrBefore>
    <Status>New</Status>
    <DynamicFieldConditions>
      <DynamicFieldCondition>
        <FieldName>Aircraft Registration</FieldName>
        <Operator>Containing</Operator>
        <Values
xmlns:a="http://schemas.microsoft.com/2003/10/Serialization/Arrays"><a
:string>reg008</a:string>
        </Values>
      </DynamicFieldCondition>
    </DynamicFieldConditions>
  </query>
</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/v1.0/">
      <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).

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. (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:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetOccurrenceIDs
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION_TOKEN</token>
      <query xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <ReportedOnOrAfter>2009-01-
27T15:55:32.9744016+00:00</ReportedOnOrAfter>
        <Status>New</Status>
      </query>
    </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/v1.0/">
      <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>
```

## 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/">
  <s:Body>
    <GetIncident
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION TOKEN</token>
      <incidentID>1</incidentID>
      <includeOccurrences>false</includeOccurrences>
    </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/v1.0/">
      <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>ryeryeryrevery</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>l-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/">
  <s:Body>
    <GetIncidents
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/"
      <token>AUTHORISATION TOKEN</token>
      <query xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <EventDateOnOrAfter>2009-06-05T14:21:00</EventDateOnOrAfter>
        <EventDateOnOrBefore>2009-09-01T14:21:00</EventDateOnOrBefore>
        <IsMandatory>true</IsMandatory>
        <ReportTypeName>Flight Safety</ReportTypeName>
      </query>
      <includeOccurrences>false</includeOccurrences>
    </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/v1.0/"
      <GetIncidentsResult xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
        <Incident>

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

        </Incident>
      </GetIncidentsResult>
    </s:Body>
  </s:Envelope>
```

```

    </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/">
  <s:Body>
    <GetIncidentIDs
      xmlns="http://www.qpulse.com/QPulseWebServices/v1.0/">
      <token>AUTHORISATION_TOKEN</token>
      <query xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <StatusName>Open</StatusName>
      </query>
    </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/v1.0/">
      <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'
)

$ctx = stream_context_create($params);
$fp = @fopen('http://<install path>/Services/Attachments.ashx', 'rb',
    false, $ctx);
$response = @stream_get_contents($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: ng0veA7xqebw1Ui0NNgHt+HHecdHjWmfz3SIDL8xJf8=
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/">
  <s:Body>
    <GetAudit xmlns="http://tempuri.org/">
      <token> AUTHORISATION TOKEN </token>
      <auditID>1</auditID>
    </GetAudit>
  </s:Body>
</s:Envelope>
```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetAuditResponse xmlns="http://tempuri.org/">
      <GetAuditResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ActualDisplayDuration>0 Day/s</a:ActualDisplayDuration>
        <a:ActualDuration>0</a:ActualDuration>
        <a:ActualDurationUnitID>Days</a:ActualDurationUnitID>
        <a:ActualDurationUnitName>Day/s</a:ActualDurationUnitName>
        <a:ActualEnd>0001-01-01T00:00:00</a:ActualEnd>
        <a:ActualEndDateTime>0001-01-01T00:00:00</a:ActualEndDateTime>
        <a:ActualStart>2009-06-15T00:00:00</a:ActualStart>
        <a:ActualStartDateTime>2009-06-
15T00:00:00</a:ActualStartDateTime>
        <a:AgreedByPersonID>-1</a:AgreedByPersonID>
        <a:AgreedByPersonName/>
        <a:AgreedOn>0001-01-01T00:00:00</a:AgreedOn>
        <a:AgreedOnDateTime>0001-01-01T00:00:00</a:AgreedOnDateTime>
        <a:AllDayEvent>>false</a:AllDayEvent>
        <a:AuditNonConformanceFindingsCollection/>
      </GetAuditResult>
    </GetAuditResponse>
  </s:Body>
</s:Envelope>
```

```

    <a:AuditCategoryDescription>External
Audit</a:AuditCategoryDescription>
    <a:AuditCategoryID>2</a:AuditCategoryID>
    <a:AuditChecklistsCollection>
      <a:AuditChecklist>
        <a:AuditChecklistQuestionsCollection>
          <a:AuditChecklistQuestion>
            <a:AreaOfStandardID>-1</a:AreaOfStandardID>
            <a:AreaOfStandardName/>
            <a:FriendlyName>1</a:FriendlyName>
            <a:Guidance/>
            <a:ID>1</a:ID>
            <a:Level>1</a:Level>
            <a:Number>1</a:Number>
            <a:ObjectParentID>1</a:ObjectParentID>
            <a:QuestionText>Sample question</a:QuestionText>
          </a:AuditChecklistQuestion>
        </a:AuditChecklistQuestionsCollection>
        <a:AuditID>1</a:AuditID>
        <a:CompletedByPersonID>-1</a:CompletedByPersonID>
        <a:CompletedByPersonName/>
        <a:CompletedDateTime>9999-12-
31T23:59:59.9999999</a:CompletedDateTime>
        <a:Description/>
        <a:FriendlyName>Checklist 1</a:FriendlyName>
        <a:ID>1</a:ID>
        <a>Title>Checklist 1</a>Title>
      </a:AuditChecklist>
    </a:AuditChecklistsCollection>
    <a:AuditFindingsCollection>
      <a:AuditFinding>
        <a:CanClose>true</a:CanClose>
        <a:Details>example observation</a:Details>
        <a:ID>1</a:ID>
        <a:Number>1</a:Number>
        <a>Status>na</a>Status>
        <a>Type>Observation</a>Type>
      </a:AuditFinding>
    </a:AuditFindingsCollection>
    <a:AuditNotesCollection/>
    <a:AuditScopeItemsCollection>
      <a:AuditScopeItem>
        <a:AuditID>1</a:AuditID>
        <a>Description>Management Commitment (ISO
9001:2008\Management Responsibility\Management
Commitment)</a>Description>
        <a:FriendlyName>10</a:FriendlyName>
        <a:ID>10</a:ID>
        <a:IsMain>true</a:IsMain>
        <a:IsPlanned>true</a:IsPlanned>
        <a>Name>Management Commitment</a>Name>

```

```

        <a:Path>ISO 9001:2008\Management Responsibility\Management
Commitment</a:Path>
        <a:Type>AreaOfStandard</a:Type>
    </a:AuditScopeItem>
</a:AuditScopeItemsCollection>
<a:AuditeesCollection>
    <a:Auditee>
        <a:AuditID>1</a:AuditID>
        <a:DepartmentOrganisationName/>
        <a:ID>1</a:ID>
        <a:Note/>
        <a:PersonName>Administrator, System</a:PersonName>
    </a:Auditee>
</a:AuditeesCollection>
<a:AuditeesCount>1</a:AuditeesCount>
<a:AuditorsCollection>
    <a:Auditor>
        <a:AuditID>1</a:AuditID>
        <a:DepartmentOrganisationName/>
        <a:ID>1</a:ID>
        <a:Note/>
        <a:PersonName>Administrator, System</a:PersonName>
    </a:Auditor>
</a:AuditorsCollection>
<a:AuditorsCount>1</a:AuditorsCount>
<a:BaseLindEndDateTime>2009-06-
26T00:00:00</a:BaseLindEndDateTime>
    <a:BaseLineStartDateTime>2009-06-
25T00:00:00</a:BaseLineStartDateTime>
    <a:BaseLineStateDateTime>2009-06-
25T00:00:00</a:BaseLineStateDateTime>
    <a:BaselineDuration>0</a:BaselineDuration>
    <a:BaselineDurationUnitID>2</a:BaselineDurationUnitID>
    <a:BaselineDurationUnitName>Day/s</a:BaselineDurationUnitName>
    <a:BrokenRulesCollection/>
    <a:CanAccept>false</a:CanAccept>
    <a:CanClose>false</a:CanClose>
    <a:Closed>0001-01-01T00:00:00</a:Closed>
    <a:ClosedByPersonID>-1</a:ClosedByPersonID>
    <a:ClosedByPersonName/>
    <a:ClosedDateTime>0001-01-01T00:00:00</a:ClosedDateTime>
    <a:CreatedByPersonID>1</a:CreatedByPersonID>
    <a:FindingsCompleteDateTime>0001-01-
01T00:00:00</a:FindingsCompleteDateTime>
    <a:FriendlyName>AUD1 - Management Commitment (ISO
9001:2008\Management Responsibility\)</a:FriendlyName>
    <a:HasBeenAgreed>false</a:HasBeenAgreed>
    <a:ID>1</a:ID>
    <a:LeadAuditor/>
    <a:LeadAuditorPersonID>-1</a:LeadAuditorPersonID>
    <a:MainScopeID>10</a:MainScopeID>

```

```

    <a:MainScopeItem>ISO 9001:2008\Management
    Responsibility\Management Commitment</a:MainScopeItem>
    <a:MainScopeType>AreaOfStandard</a:MainScopeType>
    <a:Number>AUD1</a:Number>
    <a:NumberOfIncidents>0</a:NumberOfIncidents>
    <a:NumberOfInitiatives>0</a:NumberOfInitiatives>
    <a:NumberOfObservations>1</a:NumberOfObservations>

    <a:NumberOfOutstandingIncidents>0</a:NumberOfOutstandingIncidents>
    <a:Overview/>
    <a:ScheduleConfirmed>true</a:ScheduleConfirmed>
    <a:ScheduledDisplayDuration>0 Day/s</a:ScheduledDisplayDuration>
    <a:ScheduledDuration>0</a:ScheduledDuration>
    <a:ScheduledDurationUnitID>2</a:ScheduledDurationUnitID>
    <a:ScheduledDurationUnitName>Day/s</a:ScheduledDurationUnitName>
    <a:ScheduledEnd>2009-06-25T00:00:00</a:ScheduledEnd>
    <a:ScheduledEndCalendar>2009-06-
    26T00:00:00</a:ScheduledEndCalendar>
    <a:ScheduledEndDateTime>2009-06-
    26T00:00:00</a:ScheduledEndDateTime>
    <a:ScheduledStart>2009-06-25T00:00:00</a:ScheduledStart>
    <a:ScheduledStartDate>25/06/2009</a:ScheduledStartDate>
    <a:ScopeDepartmentID/>
    <a:ScopeItemCount>1</a:ScopeItemCount>
    <a:SeriesInfo/>
    <a:StatusID>3</a:StatusID>
    <a:StatusIndicator>4</a:StatusIndicator>
    <a:StatusName>Performed</a:StatusName>
    <a:TargetDateTime>0001-01-01T00:00:00</a:TargetDateTime>
    <a>Title>Management Commitment(ISO 9001:2008\Management
    Responsibility\)</a>Title>
    <a:TotalFindingsCount>1</a:TotalFindingsCount>
    <a:UserDefinedField1Description/>
    <a:UserDefinedField1ID>-1</a:UserDefinedField1ID>
    <a:UserDefinedField2Description/>
    <a:UserDefinedField2ID>-1</a:UserDefinedField2ID>
  </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/">
  <s:Body>
    <GetAudits xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <query
        xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:AuditorName>Administrator, System</a:AuditorName>
        <a:CalendarCategoryID>2</a:CalendarCategoryID>
        <a:Keywords i:nil="true"/>
        <a:ScheduledDate>
          <a:EndDate i:nil="true"/>
          <a:Operator>After</a:Operator>
          <a:StartDate>2009-06-15T11:48:00</a:StartDate>
        </a:ScheduledDate>
        <a:StatusID>2</a:StatusID>
      </query>
    </GetAudits>
  </s:Body>
</s:Envelope>
```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetAuditsResponse xmlns="http://tempuri.org/">
      <GetAuditsResult
        xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Audit>
          ...Audit Result Goes here... (See Sample Response to
          GetAudit for message format)

        </a: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/">
  <s:Body>
    <GetAuditIDs xmlns="http://tempuri.org/">
      <token> AUTHENTICATION TOKEN </token>
      <query
xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:AuditorName>Administrator, System</a:AuditorName>
        <a:CalendarCategoryID>2</a:CalendarCategoryID>
        <a:Keywords i:nil="true"/>
        <a:ScheduledDate>
          <a:EndDate i:nil="true"/>
          <a:Operator>After</a:Operator>
          <a:StartDate>2009-06-15T12:06:00</a:StartDate>
        </a:ScheduledDate>
        <a>StatusID>2</a>StatusID>
      </query>
    </GetAuditIDs>
  </s:Body>
</s:Envelope>
```

**Sample Response:**

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

## 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/">
  <s:Body>
    <CreateNewAudit xmlns="http://tempuri.org/">
      <token> AUTHENTICATION TOKEN </token>
      <newAudit xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:AuditNonConformanceFindingsCollection/>
        <a:AuditCategoryID>1</a:AuditCategoryID>
        <a:AuditChecklistsCollection/>
        <a:AuditFindingsCollection/>
        <a:AuditNotesCollection/>
        <a:AuditScopeItemsCollection/>
        <a:AuditeesCollection/>
        <a:AuditorsCollection/>
        <a:BrokenRulesCollection/>
        <a:ScheduledStart>2009-06-15T12:47:00</a:ScheduledStart>
        <a:Title>Title of new audit</a:Title>
      </newAudit>
    </CreateNewAudit>
  </s:Body>
</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CreateNewAuditResponse xmlns="http://tempuri.org/">
      <CreateNewAuditResult
        xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
        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>
```

```
        </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 view an existing Document.

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

**Sample Request:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocument xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <documentID>1</documentID>
    </GetDocument>
  </s:Body>
</s:Envelope>
```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentResponse xmlns="http://tempuri.org/">
      <GetDocumentResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ActiveDateTime>2009-06-04T00:00:00</a:ActiveDateTime>
        <a:ApproverNotesCount>0</a:ApproverNotesCount>
        <a:ApproversCount>0</a:ApproversCount>
        <a:AttachmentName/>
        <a:AuthorPersonID>-1</a:AuthorPersonID>
        <a:AuthorPersonName/>
        <a:BrokenRulesCollection/>
        <a:CanAcknowledgeReceipt>>false</a:CanAcknowledgeReceipt>
        <a:CanDistribute>>false</a:CanDistribute>
        <a:CanRespond>>false</a:CanRespond>
        <a:ChangeAttachmentName/>
        <a:ChangeDetails/>
        <a:ChangeRequestsCollection/>
        <a:ChangeRequestsCount>0</a:ChangeRequestsCount>
        <a:CopyHoldersCollection/>
        <a:CopyHoldersCount>0</a:CopyHoldersCount>
      </GetDocumentResult>
    </GetDocumentResponse>
  </s:Body>
</s:Envelope>
```

```

    <a:CreatedByPersonID>
      <a:IsSet>true</a:IsSet>
      <a:Value>1</a:Value>
    </a:CreatedByPersonID>
    <a:DoCheckIn>>false</a:DoCheckIn>
    <a:DocumentDeleted>>false</a:DocumentDeleted>
    <a:DocumentID>1</a:DocumentID>
    <a:DocumentNotesCollection/>
    <a:DocumentRelatedDepartmentID/>
    <a:DocumentRelatedDepartmentsCollection/>
    <a:DocumentRelatedDocumentsIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
    <a:DocumentRelatedProcessesCollection/>
    <a:DocumentRelatedStandardsCollection/>
    <a:DocumentReviewsCollection/>
    <a:DocumentTypeDescription>docType3</a:DocumentTypeDescription>
    <a:DocumentTypeID>3</a:DocumentTypeID>
    <a:FriendlyName>DOC1 (1)</a:FriendlyName>
    <a:HasAttachedFiles>>false</a:HasAttachedFiles>
    <a:HasDraftRevision>>false</a:HasDraftRevision>
    <a:HasDraftText>No</a:HasDraftText>
    <a:ID>1</a:ID>
    <a:ImplementByDateTime>0001-01-
01T00:00:00</a:ImplementByDateTime>
    <a:IsCheckedOut>>false</a:IsCheckedOut>
    <a:IsOnlyVisibleToAuthor>>false</a:IsOnlyVisibleToAuthor>
    <a:Keywords/>
    <a>LastCycleStatusID>1</a>LastCycleStatusID>
    <a:Number>DOC1</a:Number>
    <a:OwnerPersonID>-1</a:OwnerPersonID>
    <a:OwnerPersonName/>
    <a:PersonTrainingEventRenewBy>2009-06-
15T00:00:00+01:00</a:PersonTrainingEventRenewBy>
    <a:RenewPeronTrainingEvents>>false</a:RenewPeronTrainingEvents>
    <a:ReviewDateTime>2010-06-04T00:00:00</a:ReviewDateTime>
    <a:ReviewsCount>0</a:ReviewsCount>
    <a:RevisionIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
      <b:int>1</b:int>
    </a:RevisionIDs>
    <a:RevisionReference>
      <a:IsSet>true</a:IsSet>
      <a:Value>1</a:Value>
    </a:RevisionReference>
    <a:Status>Active</a:Status>
    <a:StatusName>Active</a:StatusName>
    <a:SubmissionDateTime>0001-01-01T00:00:00</a:SubmissionDateTime>
    <a>Title/>
    <a>UserDefinedField1Description/>
    <a>UserDefinedField1ID>-1</a>UserDefinedField1ID>
    <a>UserDefinedField2Description/>
    <a>UserDefinedField2ID>-1</a>UserDefinedField2ID>

```

```

    <a:UserDefinedField3Description/>
    <a:UserDefinedField3ID>-1</a:UserDefinedField3ID>
    <a:UserDefinedField4Description/>
    <a:UserDefinedField4ID>-1</a:UserDefinedField4ID>
    <a:UserDefinedField5PersonID>-1</a:UserDefinedField5PersonID>
    <a:UserDefinedField5PersonName/>
  </GetDocumentResult>
</GetDocumentResponse>
</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/">
  <s:Body>
    <GetDocuments xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <query
xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ApproverID i:nil="true"/>
        <a:CopyHolderID i:nil="true"/>
        <a:DocumentOwnerID>1</a:DocumentOwnerID>
        <a:DocumentStatus>Draft</a:DocumentStatus>
        <a:DocumentTypeID>1</a:DocumentTypeID>
        <a:Keywords i:nil="true"/>
      </query>
    </GetDocuments>
  </s:Body>
</s:Envelope>

```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentsResponse xmlns="http://tempuri.org/">
      <GetDocumentsResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Document>
          ...Document Result Goes here... (See Sample Response to
GetDocument for message format)

        </a: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/">
  <s:Body>
    <GetDocumentIDs xmlns="http://tempuri.org/">
      <token>ng0veA7xqebw1Ui0NNgHt7b+WRnZzVBZzSSYfXlNV6M=</token>
      <query
xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:ApproverID i:nil="true"/>
        <a:CopyHolderID i:nil="true"/>
        <a:DocumentOwnerID>1</a:DocumentOwnerID>
        <a:DocumentStatus>Draft</a:DocumentStatus>
        <a:DocumentTypeID>1</a:DocumentTypeID>
        <a:Keywords i:nil="true"/>
      </query>
    </GetDocumentIDs>
  </s:Body>
</s:Envelope>
```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetDocumentIDsResponse xmlns="http://tempuri.org/">
      <GetDocumentIDsResult
        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>
      </GetDocumentIDsResult>
    </GetDocumentIDsResponse>
  </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/">
  <s:Body>
    <CreateNewDocument xmlns="http://tempuri.org/">
      <token>ng0veA7xqebw1Ui0NNgHt7b+WRnZzVBZzSSYfXlNV6M=</token>
      <newDocument
        xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:BrokenRulesCollection/>
        <a:ChangeRequestsCollection/>
        <a:CopyHoldersCollection/>
        <a:DocumentNotesCollection/>
        <a:DocumentRelatedDepartmentsCollection/>
      </newDocument>
    </CreateNewDocument>
  </s:Body>
</s:Envelope>
```

```

        <a:DocumentRelatedDocumentsIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
        <a:DocumentRelatedProcessesCollection/>
        <a:DocumentRelatedStandardsCollection/>
        <a:DocumentReviewsCollection/>
        <a:DocumentTypeID>1</a:DocumentTypeID>
        <a:OwnerPersonID>1</a:OwnerPersonID>
        <a:RevisionIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
        <a:Title>Title of new document</a:Title>
    </newDocument>
</CreateNewDocument>
</s:Body>
</s:Envelope>

```

#### Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CreateNewDocumentResponse xmlns="http://tempuri.org/">
      <CreateNewDocumentResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
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>

```

## 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/">
  <s:Body>
    <GetNonConformance xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <NonConformanceID>1</NonConformanceID>
    </GetNonConformance>
  </s:Body>
</s:Envelope>
```

**Sample Response:**

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetNonConformanceResponse xmlns="http://tempuri.org/">
      <GetNonConformanceResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:AssociatedDocumentDescription/>
        <a:AssociatedDocumentID>-1</a:AssociatedDocumentID>
        <a:AuditID>-1</a:AuditID>
        <a:AuditStatus>-1</a:AuditStatus>
        <a:AuditTitle/>
        <a:BatchesCollection/>
        <a:BrokenRulesCollection/>
        <a:CanClose>>false</a:CanClose>
        <a:NonConformanceSourceTypeID>7</a:NonConformanceSourceTypeID>
        <a:NonConformanceSourcePath>Customer
Audit</a:NonConformanceSourcePath>
        <a:NonConformanceStageID>-1</a:NonConformanceStageID>
        <a:NonConformanceStageIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays">
          <b:int>1</b:int>
          <b:int>2</b:int>
        </a:NonConformanceStageIDs>
      </GetNonConformanceResult>
    </GetNonConformanceResponse>
  </s:Body>
</s:Envelope>
```

```

</a:NonConformanceStageIDs>
<a:NonConformanceTemplateID>-2</a:NonConformanceTemplateID>
<a:Closed>>false</a:Closed>
<a:ClosedByPersonID>-1</a:ClosedByPersonID>
<a:ClosedByPersonName/>
<a:ClosedDate/>
<a:ClosedDateTime>0001-01-01T00:00:00</a:ClosedDateTime>
<a:Cost1>0.0000</a:Cost1>
<a:Cost1String>0.00</a:Cost1String>
<a:Cost2>0.0000</a:Cost2>
<a:Cost2String>0.00</a:Cost2String>
<a:Cost3>0.0000</a:Cost3>
<a:Cost3String>0.00</a:Cost3String>
<a:Cost4>0.0000</a:Cost4>
<a:Cost4String>0.00</a:Cost4String>
<a:Cost5>0.0000</a:Cost5>
<a:Cost5String>0.00</a:Cost5String>
<a:CreatedByPersonID>1</a:CreatedByPersonID>
<a:CurrencyID>-1</a:CurrencyID>
<a:CurrencyName/>
<a:CurrencySymbol/>
<a:Details>Something happened</a:Details>
<a:DurationUnit>Hours</a:DurationUnit>
<a:DurationUnitName>Hour/s</a:DurationUnitName>
<a:EquipmentItems/>
<a:FaultCategoryID>2</a:FaultCategoryID>
<a:FaultCategoryPath>Service Failing</a:FaultCategoryPath>
<a:FriendlyName>NC1</a:FriendlyName>
<a:ID>1</a:ID>
<a:IsOverdue>>false</a:IsOverdue>
<a:Keywords/>
<a:Notes/>
<a:Number>NC1</a:Number>
<a:OccurrenceIncidentID>-1</a:OccurrenceIncidentID>
<a:OccurrenceIncidentStageDescription/>
<a:OccurrenceIncidentStageID>-1</a:OccurrenceIncidentStageID>
<a:OwnerPersonID>1</a:OwnerPersonID>
<a:OwnerPersonName>Administrator, System</a:OwnerPersonName>
<a:ParentID>-1</a:ParentID>
<a:RaisedAgainst>-1</a:RaisedAgainst>
<a:RaisedAgainstAreaOfStandardID>-
1</a:RaisedAgainstAreaOfStandardID>
<a:RaisedAgainstAreaOfStandardPath/>
<a:RaisedAgainstDepartmentID>-1</a:RaisedAgainstDepartmentID>
<a:RaisedAgainstDepartmentPath/>
<a:RaisedAgainstName/>
<a:RaisedAgainstPersonID>-1</a:RaisedAgainstPersonID>
<a:RaisedAgainstPersonName/>
<a:RaisedAgainstProcessDescription/>
<a:RaisedAgainstProcessID>-1</a:RaisedAgainstProcessID>
<a:RaisedAgainstProductServiceDescription/>

```

```

    <a:RaisedAgainstProductServiceID>-
1</a:RaisedAgainstProductServiceID>
    <a:RaisedAgainstRevisionID>-1</a:RaisedAgainstRevisionID>
    <a:RaisedAgainstRevisionTitle/>
    <a:RaisedAgainstSupplierID>-1</a:RaisedAgainstSupplierID>
    <a:RaisedAgainstSupplierPath/>
    <a:RaisedBy>-1</a:RaisedBy>
    <a:RaisedByExternal>-1;-1</a:RaisedByExternal>
    <a:RaisedDate>15/06/2009</a:RaisedDate>
    <a:RaisedDateTime>2009-06-15T00:00:00</a:RaisedDateTime>
    <a:RaisingContactID>-1</a:RaisingContactID>
    <a:RaisingContactName/>
    <a:RaisingCustomerID>-1</a:RaisingCustomerID>
    <a:RaisingCustomerPath/>
    <a:RaisingName/>
    <a:RaisingPersonID>-1</a:RaisingPersonID>
    <a:RaisingPersonName/>
    <a:RaisingPersonOrCustomer/>
    <a:ReactionTime>0</a:ReactionTime>
    <a:RecordedByPersonID>1</a:RecordedByPersonID>
    <a:RecordedByPersonName>Administrator,
System</a:RecordedByPersonName>
    <a:ResolutionID>-1</a:ResolutionID>
    <a:ResolutionPath/>
    <a:RevisionDescription/>
    <a:RootCauseID>13</a:RootCauseID>
    <a:RootCausePath>System\Equipment Failure</a:RootCausePath>
    <a:SeverityID>4</a:SeverityID>
    <a:SeverityName>Major</a:SeverityName>
    <a:StatusID>
    <a:IsSet>true</a:IsSet>
    <a:Value>1</a:Value>
</a:StatusID>
    <a:StatusName>Open</a:StatusName>
    <a:TargetDate>19/06/2009</a:TargetDate>
    <a:TargetDateTime>2009-06-19T00:00:00</a:TargetDateTime>
    <a:Time1>0</a:Time1>
    <a:Time2>0</a:Time2>
    <a:Time3>0</a:Time3>
    <a:Time4>0</a:Time4>
    <a:Time5>0</a:Time5>
    <a:TotalCost>0.0000</a:TotalCost>
    <a:TotalCostString>0.00</a:TotalCostString>
    <a:TotalTime>0</a:TotalTime>
    <a:UserDefinedField1ID>-1</a:UserDefinedField1ID>
    <a:UserDefinedField1Path/>
    <a:UserDefinedField2ID>-1</a:UserDefinedField2ID>
    <a:UserDefinedField2Path/>
    <a:UserDefinedField3/>
    <a:UserDefinedField3DateTime>0001-01-
01T00:00:00</a:UserDefinedField3DateTime>
    <a:UserDefinedField4/>

```

```

        <a:UserDefinedField4DateTime>0001-01-
01T00:00:00</a:UserDefinedField4DateTime>
        <a:UserDefinedField5PersonID>-1</a:UserDefinedField5PersonID>
        <a:UserDefinedField5PersonName/>
        <a:UserDefinedField6PersonID>-1</a:UserDefinedField6PersonID>
        <a:UserDefinedField6PersonName/>
        <a:UserDefinedField7/>
        <a:UserDefinedField8/>
    </GetNonConformanceResult>
</GetNonConformanceResponse>
</s:Body>
</s:Envelope>

```

## GetNonConformances

**Description:** Enables a user to search for NonConformance records 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/">
  <s:Body>
    <GetNonConformances xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <query
xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Keywords i:nil="true"/>
        <a:RaisedDateCondition>
          <a:EndDate i:nil="true"/>
          <a:Operator>Today</a:Operator>
          <a:StartDate i:nil="true"/>
        </a:RaisedDateCondition>
        <a:SeverityID>4</a:SeverityID>
        <a:SourceTypeID>7</a:SourceTypeID>
        <a:StatusID>1</a:StatusID>
        <a:TargetDateCondition i:nil="true"/>
      </query>
    </GetNonConformances>
  </s:Body>
</s:Envelope>

```

Sample Response:

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetNonConformancesResponse xmlns="http://tempuri.org/">
      <GetNonConformancesResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:NonConformance>
          ...NonConformance Result Goes here... (See Sample Response to
GetNonConformance for message format)
        </a: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/">
  <s:Body>
    <GetNonConformanceIDs xmlns="http://tempuri.org/">
      <token>AUTHENTICATION TOKEN</token>
      <query
xmlns:a="http://schemas.datacontract.org/2004/07/QPulseWebServices.Contracts.Data"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:Keywords i:nil="true"/>
        <a:RaisedDateCondition>
          <a:EndDate i:nil="true"/>
          <a:Operator>Today</a:Operator>
          <a:StartDate i:nil="true"/>
        </a:RaisedDateCondition>
        <a:SeverityID>4</a:SeverityID>
        <a:SourceTypeID>7</a:SourceTypeID>
      </query>
    </GetNonConformanceIDs>
  </s:Body>
</s:Envelope>
```

```

        <a:StatusID>1</a:StatusID>
        <a:TargetDateCondition i:nil="true"/>
    </query>
</GetNonConformanceIDs>
</s:Body>
</s:Envelope>

```

Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <GetNonConformanceIDsResponse xmlns="http://tempuri.org/">
      <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/">
  <s:Body>
    <CreateNewNonConformance xmlns="http://tempuri.org/">
      <token>ng0veA7xqebw1Ui0NNgHt7b+WRnZzVBZzSSYfXlNV6M=</token>
      <newNonConformance
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:BatchesCollection/>
        <a:BrokenRulesCollection/>

```

```

        <a:NonConformanceSourceTypeID>1</a:NonConformanceSourceTypeID>
        <a:NonConformanceStageIDs
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"/>
        <a:Details>Something happened</a:Details>
        <a:EquipmentItems/>
        <a:Notes/>
        <a:RaisingPersonID>1</a:RaisingPersonID>
        <a:StatusID>
            <a:IsSet>true</a:IsSet>
            <a:Value>1</a:Value>
        </a:StatusID>
    </newNonConformance>
</CreateNewNonConformance>
</s:Body>
</s:Envelope>

```

#### Sample Response:

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <CreateNewNonConformanceResponse xmlns="http://tempuri.org/">
      <CreateNewNonConformanceResult
xmlns:a="http://www.qpulse.com/QPulseWebServices/v1.0/"
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/v1.0/"
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/v1.0/"
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/v1.0/">
      <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/>
      </RaiseNewOccurrenceResult>
    </s:Body>
  </s:Envelope>

```

```

    <Notes/>
    <RecordedByPersonName/>
    <RecordedDate>2009-08-26T13:24:00</RecordedDate>
    <ReportTypeName/>
    <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](mailto:info@gaelquality.com)

**w:** [www.gaelquality.com](http://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.