



Find the Tree. Ignore the Forest.

Integrating IntelliResponse™ With a Portal Using Web Services

The Trouble With Tight Integration

Historically, one of the most challenging aspects of web software integration has been the ability to exchange information effectively between the corresponding applications. Achieving this has typically involved a “tight coupling” of the software. Specifically, Information Technology teams from two or more companies work together to design custom specifications of how their respective software applications will talk to one another, and then integrate based on these specifications. The result of this approach is considerable implementation time and cost. Perhaps more importantly, however, it creates a solution that is proprietary. This often leads to on-going costs and headaches to maintain the “glue” between the web applications. These, and other issues, have motivated a shift toward a leaner, more cost-effective approach to web software integration.

Loose Integration: The Web Service Approach

Technically speaking, a “web service” is a software system designed to support interoperable machine-to-machine interaction over a network. The official W3C web service definition encompasses many different systems, but in common usage refers to communication via XML messages that follow the Simple Object Access Protocol (SOAP) standard. What does this mean, in layman’s terms? Basically, it provides a way for non-related applications on separate machines to “talk to one another” in the same language – similar to human communication.

One benefit of web services is that they facilitate a flexible “plug and play” approach. To use an analogy, say you like to order the grilled salmon entrée at your favourite restaurant. One day, you might order it with rice pilaf and seasonal vegetables. A month

later, you might opt to pair it with roasted potatoes and a red pepper dill sauce. The restaurant makes it easy for you to change aspects of your order to suit whatever you feel like on a given day. In the same way, web services make it easy to customize the functionality and data that is shared between two integrated applications, depending on business needs.

IntelliResponse as a Collection of Web Services

Since its inception in 2001, the IntelliResponse solution has become the de facto standard for web self-service. With its patented “One Right Answer” approach, IntelliResponse consistently delivers a first rate end user experience, and provides invaluable market insight.

With interoperability becoming increasingly more important, the IntelliResponse solution is now offered as a collection of web services. This package provides everything you need to deliver the “One Right Answer”, but also allows you (or your partners) the flexibility to use the accompanying services a la carte, based on your business needs. As a few examples, you can access user question histories, display dynamic Top 10 lists, add the ability to rate responses, and escalate to email. If, when, and how you would like to use these various services is up to you! IntelliResponse as a collection of web services is now a perfect complement to a web Portal solution, a Customer Relationship Management (CRM) tool, or an enterprise Content Management System (CMS).

The balance of this whitepaper provides detail on the particular services that IntelliResponse provides. We are consistently adding new services with each product release, so our web services package will continue to grow richer & more flexible. If you are interested in a particular service and do not see it listed here, please let us know.

IntelliResponse Web Service Details

There are currently eight services that can be called:

- » Get Session ID
- » Ask a Question
- » Ask a Question With a Keyword (optional)
- » Get Top Questions
- » Get Top Questions by Category (optional)
- » Get Top Questions by Keyword (optional)
- » Get Top Questions by Keyword And Category (optional)
- » Get a Response by ID
- » Get User History
- » Escalate to Email
- » Rate response
- » Update Interests

And these services are available from the admin side:

- » Get Folders
- » Get Email By Folder
- » Get Email By Address
- » Get Email By Assignment
- » Mark Message As Read
- » Archive Message

GET SESSION ID

Each new session or series of questions with a user, require a new session ID. This is typically called once for each session with a user and allows IR to track the series of questions that a user asks. The IR analytics module uses this information to cluster questions and monitor user behavior.

Input

in0 - User identifier (e.g. student number, or empty if anonymous) (alphanumeric string)

in1 – IP Address (e.g.. 206.23.87.123) (Optional)

Output

out - session id (numeric string). This value must be saved for the duration of the session with the current user.

Example input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getSessionId xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        </in0>
      <in1 xmlns="http://webservices.ir">
        </in1>
    </getSessionId>
  </soap:Body>
</soap:Envelope>
```

Example output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getSessionIdResponse xmlns="ir">
      <out xmlns="ir">280133</out>
    </getSessionIdResponse>
  </soap:Body>
</soap:Envelope>
```

ASK A QUESTION (ASK)

Send the natural language question the user has entered into the question box to IR for processing and display the resulting response. After the response has been displayed, the user may:

- » enter another natural language question,
- » click on a URL in the response body,
- » click on a URL link,
- » click on a related or suggested drop down
- » escalate to email
- » display the top 10 or
- » go to another area of the web.

Input

in0 - interface id (assigned by Comtext)
(numeric string)

in1 - the question that the user has
entered (alphanumeric string)

in2 - session id (from getsessionid call)
(numeric string)

Output

The output is composed of one or more
responses (Response).

Each *Response* is made up of the following
components:

ID – response number (a response ID of
zero indicates IR did not find an answer)

Body – HTML body of the response to
be displayed

Interests – interest codes assigned to
this response (zero, one or more).

Each interest contains two elements:

Code
Name

Links – a list of related URLs strings
(zero, one or more)

Each Link will contain the following
element:

String

Related – a list of related responses
created by an IR administrator (zero,
one or more)

Each Related response contains
the following elements in a

ResponseLink:

ID
Title

Suggested – a list of suggested
responses generated by IR (zero,
one or more)

Each Suggested response contains
the following elements in a

ResponseLink:

ID
Title

Title – Response title (to be optionally
displayed)

A typical question box:

Example Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ask xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        Do you have any promotions?
      </in1>
      <in2 xmlns="http://webservices.ir">
        280132
      </in2>
    </ask>
  </soap:Body>
</soap:Envelope>
```

Example output

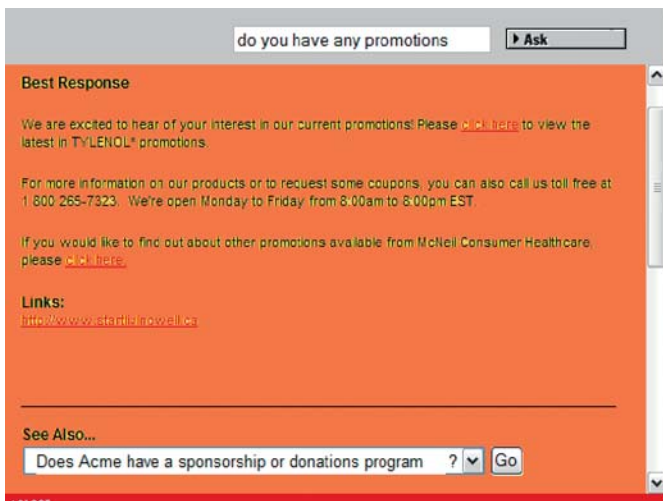
```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <askResponse xmlns="ir">
      <out xmlns="ir">
        <ns1:Response xmlns:ns1="http://webservices.ir">
          <ID xmlns="http://webservices.ir">86</ID>
          <body xmlns="http://webservices.ir"><P>We are excited to hear
of your interest in our current promotions! Please <A class=""
href="http://www.aspirin.ca/english/updates.asp"
target=_new>click here</A> to view the latest in ASPIRIN*
promotions.</P><P>For more information on our products to
request some coupons, you can call us toll free at 1 800 265-7323.
We're open Monday to Friday from 8:00am to 8:00pm EST.
</P><P>If you would like to find out about other promotions
from Acme Consumer Healthcare, please <A class=""
href="http://www.startlivingwell.ca/promotions.asp"
target=_new>click here.</A></P></body>
        </ns1:Response>
      </out>
      <interests xmlns="http://webservices.ir">
```

```

<Interest>
  <code>2</code>
  <name>Arts</name>
</Interest>
<Interest>
  <code>3</code>
  <name>Math</name>
</Interest>
</interests>
<links xmlns="http://webservices.ir">
  <ns2:string xmlns:ns2="ir">
    www.startlivingwell.ca</ns2:string>
</links>
<related xmlns="http://webservices.ir"/>
<suggested xmlns="http://webservices.ir">
  <ResponseLink>
    <id>97</id>
    <title>Does Acme have a sponsorship
      and donation program?</title>
  </ResponseLink>
  <ResponseLink>
    <id>111</id>
    <title>How can I register to your on-line
      programs?</title>
  </ResponseLink>
  <ResponseLink xsi:nil="true"/>
  <ResponseLink xsi:nil="true"/>
</suggested>
<title xmlns="http://webservices.ir">
  Do you have any ASPIRIN* promotions
  available?</title>
</ns1:Response>
</out>
</askResponse>
</soap:Body>
</soap:Envelope>

```

The above XML payload produces a result similar to this:



ASK A QUESTION WITH A KEYWORD (ASKWITHKEYWORD)

Same as *Ask a Question*, but with a specific keyword that can be used as a selection criteria. For example, Top Questions By Keyword, if enabled, would only select questions asked with the given keyword.

Input

in0 – interface id
in1 – question
in2 – session id
in3 – keyword

Sample input

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/
soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <askWithKeyword xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        How do I read your expiry?
      </in0>
      <in1 xmlns="http://webservices.ir">
        1
      </in1>
      <in2 xmlns="http://webservices.ir">
        280333
      </in2>
      <in3 xmlns="http://webservices.ir">
        automated
      </in3>
    </askWithKeyword>
  </soap:Body>
</soap:Envelope>

```

intelliresponse™

Find the Tree. Ignore the Forest.

TOP QUESTIONS (GETTOPQUESTIONS)

Top Questions will return a list of *ResponseLinks*. The *title* for each *ResponseLink* should be displayed for the user. When the user clicks on one of the titles, the corresponding *ID* should be used to request a *Response* by *ID*.

Input

in0 = interface id

in1 = number of top questions to return (typically 10)

Output

ResponseLink – a list of up to N (as specified with in1) responses ranked by popularity

Each ResponseLink contains the following elements:

iD

title

Sample input:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getTopQuestions xmlns="http://webservic.es.ir">
      <in0 xmlns="http://webservic.es.ir">
        1
      </in0>
      <in1 xmlns="http://webservic.es.ir">
        10
      </in1>
    </getTopQuestions>
  </soap:Body>
</soap:Envelope>
```

Sample output:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getTopQuestionsResponse xmlns="ir">
      <out xmlns="ir">
        <ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
          <id xmlns="http://webservic.es.ir">145</id>
          <title xmlns="http://webservic.es.ir">Help. How do I use ASK ASPIRIN*?</title>
        </ns1:ResponseLink>
        <ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
```

```
<id xmlns="http://webservic.es.ir">175</id>
<title xmlns="http://webservic.es.ir">Can I take ASPIRIN* products with other medications?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">86</id>
  <title xmlns="http://webservic.es.ir">Do you have any ASPIRIN* promotions available?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">398</id>
  <title xmlns="http://webservic.es.ir">How can I contact the makers of ASPIRIN*?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">176</id>
  <title xmlns="http://webservic.es.ir">Do ASPIRIN* products have side-effects?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">77</id>
  <title xmlns="http://webservic.es.ir">What are the ingredients in ASPIRIN* products?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">66</id>
  <title xmlns="http://webservic.es.ir">What are the recommended dosages for ASPIRIN* products?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">389</id>
  <title xmlns="http://webservic.es.ir">Where can I get information on ASPIRIN* NO. 3?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">105</id>
  <title xmlns="http://webservic.es.ir">Can I use ASPIRIN* products if I'm pregnant or nursing?</title>
</ns1:ResponseLink>
<ns1:ResponseLink xmlns:ns1="http://webservic.es.ir">
  <id xmlns="http://webservic.es.ir">388</id>
  <title xmlns="http://webservic.es.ir">Is there codeine in ASPIRIN*?</title>
</ns1:ResponseLink>
</out>
</getTopQuestionsResponse>
</soap:Body>
</soap:Envelope>
```

TOP QUESTIONS BY CATEGORY (GETTOPQUESTIONSBYCATEGORY)

This API call is just like *Top Questions*, but only selects questions from the given category id.

Input

in1 – interface id

in2 – category id

in3 – Maximum number of questions to return

Sample input:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getTopQuestionsByCategory xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        1
      </in1>
      <in2 xmlns="http://webservices.ir">
        10
      </in2>
    </getTopQuestionsByCategory>
  </soap:Body>
</soap:Envelope>
```

TOP QUESTIONS BY KEYWORD (GETTOPQUESTIONSBYKEYWORD)

This API call is just like *Top Questions*, but only selects questions asked with the given keyword (if enabled in the admin application)

Input

in1 – interface id

in2 – keyword

in3 – Maximum number of questions to return

Sample input:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getTopQuestionsByKeyword xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        automated
      </in1>
    </getTopQuestionsByKeyword>
  </soap:Body>
</soap:Envelope>
```

```
</in1>
<in2 xmlns="http://webservices.ir">
  10
</in2>
</getTopQuestionsByKeyword>
</soap:Body>
</soap:Envelope>
```

TOP QUESTIONS BY KEYWORD AND CATEGORY (GETTOPQUESTIONSBYKEYWORDANDCATEGORY)

This API call is just like *Top Questions*, but only selects questions asked with the given keyword (if enabled in the admin application)

Input

in1 – interface id

in2 – keyword

in3 – category

in4 – Maximum number of questions to return

Sample input:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getTopQuestionsByKeywordAndCategory
xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        automated
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
      <in3 xmlns="http://webservices.ir">
        10
      </in3>
    </getTopQuestionsByKeywordAndCategory>
  </soap:Body>
</soap:Envelope>
```

GET A RESPONSE BY ID (GETRESPONSE)

This API call is used to retrieve a specific response by its id number.

Input

in0 - interface id

in1 - response id

in2 - session id

in3 - source (Manual, Related, Suggested, TopQuestions, ResponseTree, EmbeddedLink, ExternalLink). The source code is used to track the question source and monitor user behaviour. Use TopQuestions source code, for example, when the user clicks on a LinkedResponse from the Top-10 display.

Output

The output is composed of zero or one response made up of the following components:

iD – response number

body – HTML body of the response to be displayed

interests – interest codes assigned to this response (zero, one or more).

Each interest contains two elements:

Code

Name

links – a list of related URLs strings (zero, one or more)

Each Link will contain the following element:

String

related – a list of related responses created by an IR administrator (zero, one or more)

Each Related response contains the following elements in a ResponseLink:

ID

Title

suggested – a list of suggested responses generated by IR (zero, one or more)

Each Suggested response contains the following elements in a ResponseLink:

ID

Title

title – Response title (to be optionally displayed)

Sample Input:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getResponse xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
```

```
</in1 xmlns="http://webservices.ir">
    90
  </in1>
  <in2 xmlns="http://webservices.ir">
    280132
  </in2>
  <in3 xmlns="http://webservices.ir">
    Suggested
  </in3>
</getResponse>
</soap:Body>
</soap:Envelope>
```

Sample Output:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getResponseResponse xmlns="ir">
      <out xmlns="ir">
        <ID xmlns="http://webservices.ir">90</ID>
        <body xmlns="http://webservices.ir">&lt;P&gt;All
          ASPIRIN* products available in Canada are
          phenylpropranolamine (PPA) free.To ensure
          the products you are using do not
          contain&nbsp;PPA, Health Canada
          advises that you check&nbsp;for
          &lt;U&gt;&lt;STRONG&gt;phenylpropranolamine
          &lt;/STRONG&gt;&lt;/U&gt;
          on&nbsp;product labels or&nbsp;that
          you consult your&nbsp;pharmacist
          or&nbsp;physician.&nbsp;&nbsp;&nbsp;
          &lt;/P&gt;</body>
        <interests xmlns="http://webservices.ir" xsi:nil="true"/>
        <links xmlns="http://webservices.ir">
          <ns1:string xmlns:ns1="ir">
            www.startlivingwell.ca
          </ns1:string>
        </links>
        <related xmlns="http://webservices.ir"/>
        <suggested xmlns="http://webservices.ir">
          <ResponseLink>
            <id>86</id>
            <title>Do you have any ASPIRIN* promotions
              available?</title>
          </ResponseLink>
```

```

<ResponseLink>
  <id>111</id>
  <title>How can I register to your on-line
  programs?</title>
</ResponseLink>
<ResponseLink xsi:nil="true"/>
<ResponseLink xsi:nil="true"/>
</suggested>
<title xmlns="http://webservices.ir">Does ASPIRIN*
contain PPA?</title>
</out>
</getResponseResponse>
</soap:Body>
</soap:Envelope>

```

GET USER HISTORY

This API call is used to get a list of all the questions a user has asked during a session.

Input

in0 – session id number

in1 – number of questions to return. If *in1* is 0, all questions are returned, if *in1* > 0, most recent *in1* questions are returned.

Output

out – an array of strings that contain the questions asked, chronologically sorted most recent to oldest.

Sample input:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getUserHistory xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        280263
      </in0>
      <in1 xmlns="http://webservices.ir">
        3
      </in0>
    </getUserHistory>
  </soap:Body>
</soap:Envelope>

```

Sample output:

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getUserHistoryResponse xmlns="ir">
      <out xmlns="ir">
        <string>Does ASPIRIN* contain PPA?</string>
        <string>How do I read your expiry?</string>
        <string>What's in Aspirin?</string>
      </out>
    </getUserHistoryResponse>
  </soap:Body>
</soap:Envelope>

```

ESCALATE TO EMAIL

This API call is used to send a question and user information to the admin application to be answered by a person.

Input

in0 – interface number

in1 – session id number

in2 – user's first name

in3 – user's last name

in4 – user's email address

in5 – user's city

in6 – user's state/province

in7 – user's country

in8 – subject line for the email

in9 – the body of the email. Often just the question, but the user can flesh it out. Our jsp's truncate this field at 16,000 characters but it is practically unlimited.

in10 – a set of strings that represent the user defined fields in this interface of the admin application. Can be left blank if user defined fields are not configured.

Output

Out – a short thank you message defined in the interface of the admin application

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <escalate xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        280335
      </in1>
      <in2 xmlns="http://webservices.ir">
        Joe
      </in2>
      <in3 xmlns="http://webservices.ir">
        Blow
      </in3>
      <in4 xmlns="http://webservices.ir">
        joe280335@domain.com
      </in4>
      <in5 xmlns="http://webservices.ir">
        Kitchener
      </in5>
      <in6 xmlns="http://webservices.ir">
        Ontario
      </in6>
      <in7 xmlns="http://webservices.ir">
        Canada
      </in7>
      <in8 xmlns="http://webservices.ir">
        Subject Line
      </in8>
      <in9 xmlns="http://webservices.ir">
        Do you sell gum?
      </in9>
      <in10 xmlns="http://webservices.ir">
      </in10>
    </escalate>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<soap:Body>
  <escalateResponse xmlns="ir">
    <out xmlns="ir">Thanks for sending us your question. Be assured that we'll respond as quickly as possible. <p>For immediate assistance, please contact our Information Centre at 1 800 265-7323. <p>Our specialists are available Monday to Friday from 8:00am to 8:00pm EST.</out>
  </escalateResponse>
</soap:Body>
</soap:Envelope>
```

RATE RESPONSE

This API is used to give a rating to a specific response as it relates to the question asked.

Input

- in0 – the interface number
- in1 – the response id number
- in2 – the rating from 0 to 100 (higher is better)
- in3 – the session id number

Output

- Out – a short thank you defined for this interface in the admin application

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <rateResponse xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        90
      </in1>
      <in2 xmlns="http://webservices.ir">
        80
      </in2>
      <in3 xmlns="http://webservices.ir">
        280335
      </in3>
    </rateResponse>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <rateResponseResponse xmlns="ir">
      <out xmlns="ir">Thank you for your input</out>
    </rateResponseResponse>
  </soap:Body>
</soap:Envelope>
```

UPDATE INTERESTS

This API is used to update the Interests table on the IR Server.

Input

in0 – interface number

in1 – a list of interest codes and descriptions that represent all of the entries in the interest table that IR needs to be aware of. Each interest consists of

- Code – a numeric string used to reference the interest
- Name – a string that describes the interest

Output

Out – the number of interests that were processed and written to the interest table in IR.

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <updateInterests xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        1
      </in0>
      <in1 xmlns="http://webservices.ir">
        <Interest><code>1</code><name>Engineering
        </name></Interest>
        <Interest><code>2</code><name>Arts
        </name></Interest>
        <Interest><code>3</code><name>Math
        </name></Interest>
      </in1>
    </updateInterests>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
<soap:Body>
  <updateInterestsResponse xmlns="ir">
    <out xmlns="ir">3</out>
  </updateInterestsResponse>
</soap:Body>
</soap:Envelope>
```

Email Service

Since email is a privileged service, all SOAP requests must include the login id and password of the administrator attempting to access the information. By including these credentials within the body of the SOAP request, SSL encryption is leveraged to keep the login information protected.

GET FOLDERS

This API is used to get the list of email folders available to the specified administrator in the given interface.

Input

- in0=login id
- in1=login password
- in2=interface id

Output

out=array of Strings listing the names of the available email folders

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getFolders xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
    </getFolders>
  </soap:Body>
</soap:Envelope>
```



Find the Tree. Ignore the Forest.

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getFoldersResponse xmlns="ir">
      <out xmlns="ir">
        <string>Inbox</string>
        <string>Archived</string>
        <string>folder1</string>
        <string>oldmail</string>
      </out>
    </getFoldersResponse>
  </soap:Body>
</soap:Envelope>
```

GET EMAIL BY FOLDER

This API is used to retrieve the complete list of emails available to the specified administrator in the given folder of the given interface.

Input

in0=login id
in1=login password
in2=interface id
in3=folder name

Output

out=an array of Email messages

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByFolder xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
      <in3 xmlns="http://webservices.ir">
        inbox
      </in3>
    </getEmailByFolder>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByFolderResponse xmlns="ir">
      <out xmlns="ir">
        <ns1:Email xmlns:ns1="http://webservices.ir">
          <assignedAdmin xmlns="http://webservices.ir">
            someguy</assignedAdmin>
          <body xmlns="http://webservices.ir">side affects
            simply dozy</body>
          <fromAddress xmlns="http://webservices.ir">
            rreddy32@hotmail.com
          </fromAddress>
          <fromName xmlns="http://webservices.ir">R<
            /fromName>
          <inquirySource xmlns="http://webservices.ir">w<
            /inquirySource>
          <messageId xmlns="http://webservices.ir">16297<
            /messageId>
          <read xmlns="http://webservices.ir">true</read>
          <sentDate xmlns="http://webservices.ir">
            2006-02-19T17:19:31.473-05:00
          </sentDate>
          <subjectLine xmlns="http://webservices.ir"/>
        </ns1:Email>
        <ns1:Email xmlns:ns1="http://webservices.ir">
          <assignedAdmin xmlns="http://webservices.ir">0<
            /assignedAdmin>
          <body xmlns="http://webservices.ir">Where can I
            buy?</body>
          <fromAddress xmlns="http://webservices.ir">
            joe280180@domain.com
          </fromAddress>
          <fromName xmlns="http://webservices.ir">Joe<
            /fromName>
          <inquirySource xmlns="http://webservices.ir">w<
            /inquirySource>
          <messageId xmlns="http://webservices.ir">16721<
            /messageId>
          <read xmlns="http://webservices.ir">true</read>
          <sentDate xmlns="http://webservices.ir">
            2007-02-07T15:13:09.377-05:00
          </sentDate>
          <subjectLine xmlns="http://webservices.ir">Subject
            Line2</subjectLine>
        </ns1:Email>
      </out>
    </getEmailByFolderResponse>
  </soap:Body>
</soap:Envelope>
```



Find the Tree. Ignore the Forest.

GET EMAIL BY ADDRESS

This API is used to retrieve the complete list of emails from a given address.

Input

in0=login id
in1=login password
in2=interface id
in3=email address

Output

out=an array of Email messages

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByAddress xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
      <in3 xmlns="http://webservices.ir">
        josh@gosh.com
      </in3>
    </getEmailByAddress>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByAddressResponse xmlns="ir">
      <out xmlns="ir">
        <ns1:Email xmlns:ns1="http://webservices.ir">
          <assignedAdmin xmlns="http://webservices.ir">
            someguy</assignedAdmin>
          <body xmlns="http://webservices.ir">
            Where can I get information on Aspirin?
          </body>
          <fromAddress xmlns="http://webservices.ir">
            josh@gosh.com</fromAddress>
```

```
<fromName xmlns="http://webservices.ir">
  Josh Gosh</fromName>
<inquirySource xmlns="http://webservices.ir">w<
  /inquirySource>
<messageId xmlns="http://webservices.ir">16543<
  /messageId>
<read xmlns="http://webservices.ir">true</read>
<sentDate xmlns="http://webservices.ir">
  2006-09-27T10:04:47-04:00
</sentDate>
<subjectLine xmlns="http://webservices.ir"/>
</ns1:Email>
<ns1:Email xmlns:ns1="http://webservices.ir">
  <assignedAdmin xmlns="http://webservices.ir">-1<
  /assignedAdmin>
  <body xmlns="http://webservices.ir">How can I
  contact the makers?</body>
  <fromAddress xmlns="http://webservices.ir">
    josh@gosh.com</fromAddress>
  <fromName xmlns="http://webservices.ir">
    Josh Gosh</fromName>
  <inquirySource xmlns="http://webservices.ir">w<
  /inquirySource>
  <messageId xmlns="http://webservices.ir">16550<
  /messageId>
  <read xmlns="http://webservices.ir">true</read>
  <sentDate xmlns="http://webservices.ir">
    2006-09-27T12:49:28.900-04:00
  </sentDate>
  <subjectLine xmlns="http://webservices.ir"/>
  </ns1:Email>
  </out>
</getEmailByAddressResponse>
</soap:Body>
</soap:Envelope>
```

GET EMAIL BY ASSIGNMENT

This API is used to retrieve the complete list of emails currently assigned to the given administrator within the specified folder.

Input

in0=login id
in1=login password
in2=interface id
in3=folder name
in4=administrator login id



Find the Tree. Ignore the Forest.

Output

out=an array of Email messages

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByAssignment xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
      <in3 xmlns="http://webservices.ir">
        inbox
      </in3>
      <in4 xmlns="http://webservices.ir">
        someguy
      </in4>
    </getEmailByAssignment>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <getEmailByAddressResponse xmlns="ir">
      <out xmlns="ir">
        <ns1:Email xmlns:ns1="http://webservices.ir">
          <assignedAdmin xmlns="http://webservices.ir">
            someguy</assignedAdmin>
          <body xmlns="http://webservices.ir">side affects
            simply dozy</body>
          <fromAddress xmlns="http://webservices.ir">
            reddy32@hotmail.com
          </fromAddress>
          <fromName xmlns="http://webservices.ir">R<
            /fromName>
          <inquirySource xmlns="http://webservices.ir">w<
            /inquirySource>
```

```
<messageId xmlns="http://webservices.ir">16297<
  /messageId>
  <read xmlns="http://webservices.ir">true</read>
  <sentDate xmlns="http://webservices.ir">
    2006-02-19T17:19:31.473-05:00
  </sentDate>
  <subjectLine xmlns="http://webservices.ir"/>
</ns1:Email>
<ns1:Email xmlns:ns1="http://webservices.ir">
  <assignedAdmin xmlns="http://webservices.ir">
    someguy</assignedAdmin>
  <body xmlns="http://webservices.ir">
    Can I take aspirin with other medications
  </body>
  <fromAddress xmlns="http://webservices.ir">
    josh@gosh.com</fromAddress>
  <fromName xmlns="http://webservices.ir">Josh<
    /fromName>
  <inquirySource xmlns="http://webservices.ir">w<
    /inquirySource>
  <messageId xmlns="http://webservices.ir">16548<
    /messageId>
  <read xmlns="http://webservices.ir">true</read>
  <sentDate xmlns="http://webservices.ir">
    2006-09-27T10:39:46.170-04:00
  </sentDate>
  <subjectLine xmlns="http://webservices.ir"/>
</ns1:Email>
</out>
</getEmailByAssignmentResponse>
</soap:Body>
</soap:Envelope>
```

MARK MESSAGE AS READ

This API is used to mark a given Email as having been read. This is useful for synchronizing mail states between applications.

Input

in0=login id
in1=login password
in2=message id

Output

out=success message

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <markMessageAsRead xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        16550
      </in2>
    </markMessageAsRead>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <markMessageAsReadResponse xmlns="ir">
      <out xmlns="ir">Success</out>
    </markMessageAsReadResponse>
  </soap:Body>
</soap:Envelope>
```

ARCHIVE MESSAGE

This API is used to move a given Email to the specified archive folder.
This is useful for organizing mail between applications.

Input

in0=login id
in1=login password
in2=interface id
in3=message id
in4=folder name

Output

out=success message

Sample Input

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <archiveMessage xmlns="http://webservices.ir">
      <in0 xmlns="http://webservices.ir">
        admin
      </in0>
      <in1 xmlns="http://webservices.ir">
        adminpass
      </in1>
      <in2 xmlns="http://webservices.ir">
        1
      </in2>
      <in3 xmlns="http://webservices.ir">
        16550
      </in3>
      <in4 xmlns="http://webservices.ir">
        oldmail
      </in4>
    </archiveMessage>
  </soap:Body>
</soap:Envelope>
```

Sample Output

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <archiveMessageResponse xmlns="ir">
      <out xmlns="ir">Success</out>
    </archiveMessageResponse>
  </soap:Body>
</soap:Envelope>
```