



XMPP

XEP-0055: Jabber Search

Peter Saint-Andre

<mailto:stpeter@stpeter.im>

<xmpp:stpeter@jabber.org>

<https://stpeter.im/>

2009-09-15

Version 1.3

Status	Type	Short Name
Active	Historical	iq-search

This specification provides canonical documentation of the jabber:iq:search namespace currently in use within the Jabber community.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Use Cases	1
2.1	Searching	1
3	Extensibility	3
4	Internationalization Considerations	5
5	Security Considerations	6
6	IANA Considerations	6
7	XMPP Registrar Considerations	6
7.1	Protocol Namespaces	6
7.2	Field Standardization	6
8	XML Schema	7

1 Introduction

This specification documents a protocol currently used to search information repositories on the Jabber network. To date, the jabber:iq:search protocol has been used mainly to search for people who have registered with user directories (e.g., the "Jabber User Directory" hosted at users.jabber.org). However, the jabber:iq:search protocol is not limited to user directories, and could be used to search other Jabber information repositories (such as chatroom directories) or even to provide a Jabber interface to conventional search engines.

The basic functionality is to query an information repository regarding the possible search fields, to send a search query, and to receive search results. Note well that there is currently no mechanism for paging through results or limiting the number of "hits", and that the allowable search fields are limited to those defined in the XML schema; however, extensibility MAY be provided via the [Data Forms \(XEP-0004\)](#)¹ protocol, as described below.

2 Use Cases

2.1 Searching

In order to search an information repository, a user first needs to discover what search fields are supported by the service:

Listing 1: Requesting Search Fields

```
<iq type='get'
  from='romeo@montague.net/home'
  to='characters.shakespeare.lit'
  id='search1'
  xml:lang='en'>
  <query xmlns='jabber:iq:search' />
</iq>
```

The service MUST then return the possible search fields to the user, and MAY include instructions:

Listing 2: Receiving Search Fields

```
<iq type='result'
  from='characters.shakespeare.lit'
  to='romeo@montague.net/home'
  id='search1'
  xml:lang='en'>
  <query xmlns='jabber:iq:search'>
    <instructions>
      Fill in one or more fields to search
```

¹XEP-0004: Data Forms <<https://xmpp.org/extensions/xep-0004.html>>.

```
    for any matching Jabber users.  
</instructions>  
<first/>  
<last/>  
<nick/>  
<email/>  
</query>  
</iq>
```

The user MAY then submit a search request, specifying values for any desired fields:

Listing 3: Submitting a Search Request

```
<iq type='set'  
  from='romeo@montague.net/home'  
  to='characters.shakespeare.lit'  
  id='search2'  
  xml:lang='en'>  
  <query xmlns='jabber:iq:search'>  
    <last>Capulet</last>  
  </query>  
</iq>
```

The service SHOULD then return a list of search results that match the values provided:

Listing 4: Receiving Search Results

```
<iq type='result'  
  from='characters.shakespeare.lit'  
  to='romeo@montague.net/home'  
  id='search2'  
  xml:lang='en'>  
  <query xmlns='jabber:iq:search'>  
    <item jid='juliet@capulet.com'>  
      <first>Juliet</first>  
      <last>Capulet</last>  
      <nick>JuliC</nick>  
      <email>juliet@shakespeare.lit</email>  
    </item>  
    <item jid='tybalt@shakespeare.lit'>  
      <first>Tybalt</first>  
      <last>Capulet</last>  
      <nick>ty</nick>  
      <email>tybalt@shakespeare.lit</email>  
    </item>  
  </query>  
</iq>
```

If there are no matching directory entries, the service MUST return an empty <query/> element:

Listing 5: Service Informs User that No Records Match

```
<iq type='result'
  from='characters.shakespeare.lit'
  to='romeo@montague.net/home'
  id='search2'
  xml:lang='en'>
  <query xmlns='jabber:iq:search' />
</iq>
```

3 Extensibility

The fields defined in the 'jabber:iq:search' namespace are strictly limited to those specified in the schema. If a host needs to gather additional information, **Data Forms** SHOULD be used; a host MUST NOT add new fields to the 'jabber:iq:search' namespace. Support for extensibility via **Data Forms** is RECOMMENDED, but is not required for compliance with this document. The extensibility mechanism for searching makes use of a hidden FORM_TYPE field for the purpose of standardizing field names within the form, as defined in [Field Standardization for Data Forms \(XEP-0068\)](#)². Implementations supporting this extensibility mechanism SHOULD support field standardization as well.

Listing 6: Entity Requests Search Fields from Service

```
<iq type='get'
  from='juliet@capulet.com/balcony'
  to='characters.shakespeare.lit'
  id='search3'
  xml:lang='en'>
  <query xmlns='jabber:iq:search' />
</iq>
```

Listing 7: Host Returns Search Form to Entity

```
<iq type='result'
  from='characters.shakespeare.lit'
  to='juliet@capulet.com/balcony'
  id='search3'
  xml:lang='en'>
  <query xmlns='jabber:iq:search'>
    <instructions>
      Use the enclosed form to search. If your Jabber client does not
```

²XEP-0068: Field Data Standardization for Data Forms <<https://xmpp.org/extensions/xep-0068.html>>.

```

    support Data Forms, visit http://shakespeare.lit/
</instructions>
<x xmlns='jabber:x:data' type='form'>
  <title>User Directory Search</title>
  <instructions>
    Please provide the following information
    to search for Shakespearean characters.
  </instructions>
  <field type='hidden'
    var='FORM_TYPE'>
    <value>jabber:iq:search</value>
  </field>
  <field type='text-single'
    label='Given_Name'
    var='first' />
  <field type='text-single'
    label='Family_Name'
    var='last' />
  <field type='list-single'
    label='Gender'
    var='x-gender'>
    <option label='Male'><value>male</value></option>
    <option label='Female'><value>female</value></option>
  </field>
</x>
</query>
</iq>

```

Listing 8: Entity Submits Search Form

```

<iq type='set'
  from='juliet@capulet.com/balcony'
  to='characters.shakespeare.lit'
  id='search4'
  xml:lang='en'>
  <query xmlns='jabber:iq:search'>
    <x xmlns='jabber:x:data' type='submit'>
      <field type='hidden' var='FORM_TYPE'>
        <value>jabber:iq:search</value>
      </field>
      <field var='x-gender'>
        <value>male</value>
      </field>
    </x>
  </query>
</iq>

```

Listing 9: Service Returns Search Results

```

<iq type='result'

```

```

from='characters.shakespeare.lit'
to='juliet@capulet.com/balcony'
id='search4'
xml:lang='en'>
<query xmlns='jabber:iq:search'>
  <x xmlns='jabber:x:data' type='result'>
    <field type='hidden' var='FORM_TYPE'>
      <value>jabber:iq:search</value>
    </field>
    <reported>
      <field var='first' label='Given_Name' type='text-single'>
      <field var='last' label='Family_Name' type='text-single'>
      <field var='jid' label='Jabber_ID' type='jid-single'>
      <field var='x-gender' label='Gender' type='list-single'>
    </reported>
    <item>
      <field var='first'><value>Benvolio</value></field>
      <field var='last'><value>Montague</value></field>
      <field var='jid'><value>benvolio@montague.net</value></field>
      <field var='x-gender'><value>male</value></field>
    </item>
    <item>
      <field var='first'><value>Romeo</value></field>
      <field var='last'><value>Montague</value></field>
      <field var='jid'><value>romeo@montague.net</value></field>
      <field var='x-gender'><value>male</value></field>
    </item>
  </x>
</query>
</iq>

```

4 Internationalization Considerations

The intent of the <first/> and <last/> elements (and associated data form fields) is that they map to given name and family name, respectively. Therefore, cultures that place the family name first and the given name second (e.g., as is done in China) would use <first/> for the given name and <last/> for the family name, NOT the other way around.

For example, the name of the Chinese philosopher Confucius would be rendered as shown below because his given name was Qui and his family name was Kong.

Listing 10: Internationalization of names

```

<iq type='submit'
  from='confucius@scholars.lit/home'
  to='registrar.scholars.lit'
  id='kj3b157n'
  xml:lang='en'>

```



```
<query xmlns='jabber:iq:register'>
  <username>confucius</username>
  <first>Qui</first>
  <last>Kong</last>
</query>
</iq>
```

5 Security Considerations

There are no security features or concerns related to this proposal.

6 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)³.

7 XMPP Registrar Considerations

The [XMPP Registrar](#)⁴ shall include the following information in its registries.

7.1 Protocol Namespaces

The XMPP Registrar includes the 'jabber:iq:search' namespace in its registry of protocol namespaces.

7.2 Field Standardization

The following fields are reserved for use within jabber:x:data forms scoped by a FORM_TYPE of 'jabber:iq:search'; additional fields MAY be added via the XMPP Registrar's normal registration process but are outside the scope of this document.

³The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

⁴The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

```
<form_type>
  <name>jabber:iq:search</name>
  <doc>XEP-0055</doc>
  <desc>Forms enabling directory searches.</desc>
  <field
    var='first'
    type='text-single'
    label='Given_Name' />
  <field
    var='last'
    type='text-single'
    label='Family_Name' />
  <field
    var='nick'
    type='text-single'
    label='Nickname' />
  <field
    var='email'
    type='text-single'
    label='Email_Address' />
</form_type>
```

8 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='jabber:iq:search'
  xmlns='jabber:iq:search'
  elementFormDefault='qualified'>

  <xs:import namespace='jabber:x:data'
    schemaLocation='http://www.xmpp.org/schemas/x-data.xsd' />

  <xs:annotation>
    <xs:documentation>
      The protocol documented by this schema is defined in
      XEP-0055: http://www.xmpp.org/extensions/xep-0055.html
    </xs:documentation>
  </xs:annotation>

  <xs:element name='query'>
    <xs:complexType>
      <xs:choice>
        <xs:all xmlns:xdata='jabber:x:data'>
          <xs:element name='instructions' type='xs:string' />
        </xs:all>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        <xs:element name='first' type='xs:string' />
        <xs:element name='last' type='xs:string' />
        <xs:element name='nick' type='xs:string' />
        <xs:element name='email' type='xs:string' />
        <xs:element ref='xdata:x' minOccurs='0' />
    </xs:all>
    <xs:element ref='item' minOccurs='0' maxOccurs='unbounded' />
</xs:choice>
</xs:complexType>
</xs:element>

<xs:element name='item'>
    <xs:complexType>
        <xs:all>
            <xs:element name='first' type='xs:string' />
            <xs:element name='last' type='xs:string' />
            <xs:element name='nick' type='xs:string' />
            <xs:element name='email' type='xs:string' />
        </xs:all>
        <xs:attribute name='jid' type='xs:string' use='required' />
    </xs:complexType>
</xs:element>

</xs:schema>
```