



XMPP

XEP-0401: Easy User Onboarding

Marc Schink

2020-01-08
Version 0.4.0

Status	Type	Short Name
Experimental	Standards Track	N/A

This document defines a protocol and URI scheme for user invitation in order to allow a third party to register on a server. The goal of this is to make onboarding for XMPP IM newcomers as easy as possible.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2020 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
1.1	User Invitation	1
1.2	Account Creation	1
2	Requirements	1
3	Discovery	2
4	Glossary	2
5	Use Cases	3
5.1	Creating a User Invitation	3
5.2	Landing Page	4
5.3	Redeeming a User Invitation	4
5.3.1	Pre-Configured Account	5
5.3.2	No Configured Account	5
5.4	Initiating Account Creation	5
5.5	Pre-Authenticated In-Band Registration	7
6	Business Rules	9
6.1	Fallback to Client-Side PARS	9
6.2	Account Creation	10
7	Implementation Notes	10
7.1	XMPP Server Suggestion for Invitees	10
8	Accessibility Considerations	10
9	Internationalization Considerations	10
10	Security Considerations	10
11	IANA Considerations	10
12	XMPP Registrar Considerations	11
13	XML Schema	11

1 Introduction

Romeo is an active XMPP IM (Instant Messaging) user or the operator of an XMPP server. He convinces Juliet (who may not have an XMPP account yet) to install a client but she may still need to choose an XMPP server, create an account, and add Romeo as a contact. This specification defines two ways for Romeo to simplify this process for Juliet:

1.1 User Invitation

If Romeo is an XMPP user, he can create an out-of-band link (URI) which allows Juliet to:

1. Download an XMPP client (if needed).
2. Optionally register an account with Romeo's server (if permitted by the server rules), or with a public server.
3. Establish a mutual presence subscription between Romeo and Juliet.

The process is designed to automatically skip steps that Juliet already completed, to make the overall experience as smooth as possible.

1.2 Account Creation

If Romeo is an administrator of an XMPP server, he can create an out-of-band link (URI) which allows Juliet to:

1. Download an XMPP client (if needed).
2. Register an account on Romeo's server with a user name defined by Romeo and a password not known to Romeo.
3. Establish a mutual presence subscription between Romeo and Juliet.

2 Requirements

This specification makes use of XMPP URIs. The basic URI scheme for XMPP is defined in [RFC 5122](#)¹ and extended in [XMPP URI Query Components \(XEP-0147\)](#)² and [Pre-Authenticated Roster Subscription \(XEP-0379\)](#)³. Furthermore, this heavily builds upon the blocks provided

¹RFC 5122: Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP) <<http://tools.ietf.org/html/rfc5122>>.

²XEP-0147: XMPP URI Query Components <<https://xmpp.org/extensions/xep-0147.html>>.

³XEP-0379: Pre-Authenticated Roster Subscription <<https://xmpp.org/extensions/xep-0379.html>>.

in XEP-0379 for landing page and roster subscription.

To create out-of-band invitation links, Romeo's server needs to implement the [Ad-Hoc Commands \(XEP-0050\)](#)⁴ commands specified in the following section, and his client must be able to execute them.

Furthermore, Romeo's server SHOULD provide a HTTPS service hosting the landing page.

3 Discovery

Romeo can query his server for the availability of "User Invitation" and "Account Creation" commands:

Listing 1: Discover available ad-hoc commands

```
<iq type='get' from='romeo@example.com' to='example.com' id='disco'>
  <query xmlns='http://jabber.org/protocol/disco#items'
        node='http://jabber.org/protocol/commands' />
</iq>
```

TODO: use appropriate node namespace.

Listing 2: Discovery result for available ad-hoc commands

```
<iq type='result' to='romeo@example.com' from='example.com' id='disco'>
  <query xmlns='http://jabber.org/protocol/disco#items'
        node='http://jabber.org/protocol/commands'>
    <item jid='example.com'
          node='urn:xmpp:invite#invite'
          name='Invite_user' />
    <item jid='example.com'
          node='urn:xmpp:invite#create-account'
          name='Create_account' />
  </query>
</iq>
```

When performing the account creation, Juliet's client needs to ensure that the server supports the extended IBR protocol with a <preauth> token: TODO

4 Glossary

OPTIONAL.

⁴XEP-0050: Ad-Hoc Commands <<https://xmpp.org/extensions/xep-0050.html>>.

5 Use Cases

5.1 Creating a User Invitation

A user can execute the 'invite' command to obtain a new invitation link with a unique invitation token.

Listing 3: Execute user invitation command

```
<iq type='set' from='romeo@example.com' to='example.com' id='exec1'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='urn:xmpp:invite#invite'
    action='execute' />
</iq>
```

Listing 4: User invitation finished

```
<iq type='result' to='romeo@example.com' from='example.com' id='exec2'
  >
  <command xmlns='http://jabber.org/protocol/commands'
    node='urn:xmpp:invite#invite'
    status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <item>
        <field var='uri'>
          <value>xmpp:inviter@example.com?roster;preauth=TOKEN;ibr=y</
            value>
          </field>
          <field var='landing-url'>
            <value>https://example.com/invite/#TOKEN</value>
          </field>
          <field var='expire'>
            <value>2017-11-06T02:56:15Z</value>
          </field>
        </item>
      </x>
    </command>
  </iq>
```

The token should be unique, sufficiently long and generated by a strong random number generator.

A server MUST provide the **uri** field which contains an XMPP URI of the following format:

```
xmpp:inviter@example.com?roster;preauth=TOKEN;ibr=y
```

The **ibr** query component in the XMPP URI indicates that the invitee is allowed to create an account on Romeo's server, using the 'preauth' token. If the server does not support or allow in-band registration for invited users, the server MUST omit the **ibr** query component.

Additionally, the server SHOULD provide the **landing-url** field which contains an HTTPS URL of a web-based landing page as described in [Pre-Authenticated Roster Subscription \(XEP-0379\)](#)⁵ § 3.3. The URL format may differ from the example shown here depending on where the landing page is hosted.

If the server omits the **landing-page** field, Romeo's client SHOULD generate an appropriate landing page URL hosted by the client developer or a trusted third party.

A server MAY provide a field which provides the expiration date of the generated token. The expiration date MUST conform to the DateTime profile specified in [XMPP Date and Time Profiles \(XEP-0082\)](#)⁶. If the field is not provided, the token does not expire.

Romeo's client should provide adequate means to export the **landing-page** URL, possibly accompanied with a short description and the **expire** information, so that Romeo can share it with Juliet by other means than XMPP, like e-mail or a QR code.

5.2 Landing Page

The landing page that the generated URL points to should correspond to the format described in XEP-0379 §3.3, and it needs to convey the following information:

- A short text that this is an XMPP invitation from Romeo.
- A client recommendation (based on the detected web browser/OS) with download links.
- A prominent button that activates the encoded **xmpp:** link.

If the landing page is hosted by Romeo's server, the server MAY display additional information based on the supplied TOKEN value, like the name of the inviter or validity information.

5.3 Redeeming a User Invitation

If Juliet does not have an XMPP client installed, she will not be able to open the **xmpp:** link from the invitation page. For this case, the landing page needs to indicate that a client must be installed first, and that the link will not work as intended without. The automatic installation of an appropriate IM client when a user clicks on an **xmpp:** is outside of the scope of this document.

With an XMPP client installed, Juliet can open the **xmpp:** link and have the client process it appropriately, as follows:

⁵XEP-0379: Pre-Authenticated Roster Subscription <<https://xmpp.org/extensions/xep-0379.html>>.

⁶XEP-0082: XMPP Date and Time Profiles <<https://xmpp.org/extensions/xep-0082.html>>.

5.3.1 Pre-Configured Account

If Juliet's client is already configured with an account, the default action for the presented `xmpp:inviter@example.com?roster;...` URI is to add the inviter to Juliet's roster. This should be performed as described in §3.4 of XEP-0379, by sending a presence subscription request containing the 'preauth' token.

If Juliet already has Romeo in her roster, her client should open the appropriate chat interface instead.

5.3.2 No Configured Account

If Juliet's client does not have an XMPP account configured, she needs to login or register an account first. Therefore, the client should provide an interface with the following options:

- Login with an existing XMPP account.
- Register an account with Romeo's server (if the URI contains a `ibr=y` parameter).
- Register an account with a public or client-endorsed server.

If the `xmpp:` URI provided by Romeo contains the `ibr=y` parameter, it indicates that the server supports the [Pre-Authenticated In-Band Registration](#) defined in this document. If Juliet chooses this approach, the server will ensure that after the registration, Romeo is added to her roster with a full presence subscription.

If Juliet chooses to login or register with a different server, her client must complete the respective process and issue a subscription request as described in §3.4 of XEP-0379.

5.4 Initiating Account Creation

If Romeo is the administrator of an XMPP server, he might want to ensure that Juliet obtains an account on this server, with a username defined either by Romeo or by Juliet, and in a way that does not require the out-of-band communication of user passwords.

TODO: description of overall process steps, design motivation.

Listing 5: Execute account creation command

```
<iq type='set' from='romeo@example.com' to='example.com' id='exec1'>
  <command xmlns='http://jabber.org/protocol/commands'
    node='urn:xmpp:invite#create-account'
    action='execute' />
</iq>
```


Listing 6: Service returns form for account creation

```

<iq type='result' to='romeo@example.com' from='example.com' id='exec1'
>
  <command xmlns='http://jabber.org/protocol/commands'
    sessionid='config:20020923T213616Z-700'
    node='urn:xmpp:invite#create-account'
    status='executing'>
    <actions execute='complete'>
      <complete/>
    </actions>
    <x xmlns='jabber:x:data' type='form'>
      <field var='username' label='Username' type='text-single'/>
      <field var='roster-subscription' label='Roster_subscription'
        type='boolean'/>
    </x>
  </command>
</iq>

```

A server MAY require a username to be specified for account creation. In this case, the server MUST add the <required/> element to the username field. The username MUST be a valid localpart as defined in RFC 6122⁷ §2.3.

Listing 7: Account creation with specified username

```

<iq type='set' from='romeo@example.com' to='example.com' id='exec2'>
  <command xmlns='http://jabber.org/protocol/commands'
    sessionid='config:20020923T213616Z-700'
    node='urn:xmpp:invite#create-account'>
    <x xmlns='jabber:x:data' type='submit'>
      <field var='username'>
        <value>juliet</value>
      </field>
    </x>
  </command>
</iq>

```

Listing 8: Account creation finished

```

<iq type='result' to='romeo@example.com' from='example.com' id='exec2'
>
  <command xmlns='http://jabber.org/protocol/commands'
    sessionid='config:20020923T213616Z-700'
    node='urn:xmpp:invite#create-account'
    status='completed'>
    <x xmlns='jabber:x:data' type='result'>
      <item>

```

⁷RFC 6122: Extensible Messaging and Presence Protocol (XMPP): Address Format <<http://tools.ietf.org/html/rfc6122>>.

```

    <field var='uri'>
      <value>xmpp:juliet@example.com?register;preauth=TOKEN</value>
    </field>
    <field var='landing-url'>
      <value>https://example.com/invite/#TOKEN</value>
    </field>
    <field var='expire'>
      <value>2017-11-06T02:56:15Z</value>
    </field>
  </item>
</x>
</command>
</iq>

```

The server's response for account creation is the same as for user invitation except for the format of the **uri** field which contains an XMPP URI of the following format:

```
xmpp:juliet@example.com?register;preauth=TOKEN
```

If no username was specified during the account creation process, the local part of the JID in the XMPP URI is omitted by the server which results in the following format:

```
xmpp:example.com?register;preauth=TOKEN
```

TODO: note about sensitivity of TOKEN

5.5 Pre-Authenticated In-Band Registration

In order to allow invited users to register on a server, the registration process as defined in [In-Band Registration \(XEP-0077\)](#)⁸ needs to be extended. The invited user's client MUST add a `<preauth>` element in the 'TODO' namespace to the 'jabber:iq:register' query in order to inform the server that it wants to perform Pre-Authenticated IBR:

Listing 9: Retrieving registration fields

```

<iq type='get' id='reg1' to='example.com'>
  <query xmlns='jabber:iq:register'>
    <preauth xmlns='urn:xmpp:invite:1' />
  </query>
</iq>

```

If the server supports and is ready to perform Pre-Authenticated IBR, it MUST add a `<token>` element to the response (TODO: 'token' or 'preauth?'):

⁸XEP-0077: In-Band Registration <<https://xmpp.org/extensions/xep-0077.html>>.

Listing 10: Receiving registration form

```

<iq type='result' to='romeo@example.com' from='example.com' id='reg1'>
  <query xmlns='jabber:iq:register'>
    <x xmlns='jabber:x:data' type='form'>
      <field type='hidden' var='FORM_TYPE'>
        <value>urn:xmpp:invite:1</value>
      </field>
      <field type='text-single' label='Username' var='username'>
        <required/>
      </field>
      <field type='text-private' label='Password' var='password'>
        <required/>
      </field>
      <field type='text-single' label='Invite_token' var='token'>
        <required/>
      </field>
    </x>
  </query>
</iq>

```

Listing 11: Receiving registration form with error (invalid token)

```

<iq type='error' from='example.com' id='reg1'>
  <query xmlns='jabber:iq:register'>
    <x xmlns='jabber:x:data' type='form'>
      <field type='hidden' var='FORM_TYPE'>
        <value>urn:xmpp:invite:1</value>
      </field>
      <field type='text-single' var='username'>
        <value>juliet</value>
      </field>
      <field type='text-private' var='password'>
        <value>m1cro$oft</value>
      </field>
      <field type='text-single' var='token'>
        <value>BADTOKEN</value>
      </field>
    </x>
  </query>
  <error type='modify'>
    <bad-request xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
    <invalid-token xmlns='urn:xmpp:invite:1' />
  </error>
</iq>

```

Listing 12: Receiving registration form with error (token expired)

```

<iq type='error' from='example.com' id='reg1'>
  <query xmlns='jabber:iq:register'>

```

```

<x xmlns='jabber:x:data' type='form'>
  <field type='hidden' var='FORM_TYPE'>
    <value>urn:xmpp:invite:1</value>
  </field>
  <field type='text-single' var='username'>
    <value>juliet</value>
  </field>
  <field type='text-private' var='password'>
    <value>microsoft</value>
  </field>
  <field type='text-single' var='token'>
    <value>OLDTOKEN</value>
  </field>
</x>
</query>
<error type='cancel'>
  <not-allowed xmlns='urn:ietf:params:xml:ns:xmpp-stanzas' />
  <token-expired xmlns='urn:xmpp:invite:1' />
</error>
</iq>

```

After the invitee has successfully registered on the inviter's server and roster subscription is enabled for account creation, the server MUST use roster pushes as defined in RFC 6121⁹ §2.1.6 in order to inform the inviter about the invitee's new account without the need to reconnect.

Listing 13: Push roster item of invitee to inviter

```

<iq type='set' from='romeo@example.com' id='push'>
  <query xmlns='jabber:iq:roster'>
    <item subscription='both' jid='juliet@example.com' />
  </query>
</iq>

```

6 Business Rules

6.1 Fallback to Client-Side PARS

If the inviter's server does not support user invitation, the client application SHOULD silently fall back to [Pre-Authenticated Roster Subscription \(XEP-0379\)](#)¹⁰ for a good user experience.

⁹RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

¹⁰XEP-0379: Pre-Authenticated Roster Subscription <<https://xmpp.org/extensions/xep-0379.html>>.

6.2 Account Creation

If a username was specified during the account creation process, the server SHOULD NOT create an account on the server until the invitee actually registers it with the corresponding token. The server MUST reserve the username at least until the corresponding token expires.

7 Implementation Notes

7.1 XMPP Server Suggestion for Invitees

If the invitee opens an invitation URI with **ibr=y** and chooses to create a new account, the client SHOULD pre-fill the inviter JID's domain part as the new account's domain. The client MAY provide a mechanism to enter or choose a different server, though.

8 Accessibility Considerations

OPTIONAL.

9 Internationalization Considerations

OPTIONAL.

10 Security Considerations

See security considerations in [Pre-Authenticated Roster Subscription \(XEP-0379\)](#) ¹¹.

11 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#) ¹².

¹¹XEP-0379: Pre-Authenticated Roster Subscription <<https://xmpp.org/extensions/xep-0379.html>>.

¹²The 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/>>.

12 XMPP Registrar Considerations

As authorized by [XMPP URI Query Components \(XEP-0147\)](https://xmpp.org/extensions/xep-0147.html)¹³, the XMPP Registrar maintains a registry of queries and key-value pairs for use in XMPP URIs (see [<https://xmpp.org/registrar/querytypes.html>](https://xmpp.org/registrar/querytypes.html)).

The key-value parameter **preauth** is added to the **register** query action as defined in [In-Band Registration \(XEP-0077\)](https://xmpp.org/extensions/xep-0077.html)¹⁴

```
<querytype>
  <name>register</name>
  ...
  <key>
    <name>preauth</name>
    <desc>the token used to allow one-time in-band registration on
      the inviter's_s_server</desc>
  </key>
</querytype>
```

In addition to the **preauth** key-value parameter define in [Pre-Authenticated Roster Subscription \(XEP-0379\)](https://xmpp.org/extensions/xep-0379.html)¹⁵, the **ibr** parameter is added to the **roster** query action.

```
<querytype>
  <name>roster</name>
  ...
  <key>
    <name>ibr</name>
    <value>y</value>
    <desc>the parameter to indicate that the token allows the
      invitee to create an account on the inviter's_s_server_via_in-
      band_registration</desc>
  </key>
</querytype>
```

13 XML Schema

REQUIRED for protocol specifications.

¹³XEP-0147: XMPP URI Query Components [<https://xmpp.org/extensions/xep-0147.html>](https://xmpp.org/extensions/xep-0147.html).

¹⁴XEP-0077: In-Band Registration [<https://xmpp.org/extensions/xep-0077.html>](https://xmpp.org/extensions/xep-0077.html).

¹⁵XEP-0379: Pre-Authenticated Roster Subscription [<https://xmpp.org/extensions/xep-0379.html>](https://xmpp.org/extensions/xep-0379.html).