Lab 11: Configure User Self-Service

Goal

This lab focuses on configuring the self-service features of OpenIDM as presented in chapter 4.3, "Configuring User Self-Service" of the *OpenIDM Integrator's Guide*.

Objectives

Upon completion of this lab, you should be able to:

- Configure outgoing email service for OpenIDM
- Enable email-based self-registration and password reset
- Enable Google reCAPTCHA
- Implement authentication-based password reset

Requirements

The following requirements are U before starting this exercise:

- You ned an installed version of OpenIDM.

References

The following references were used in the development of this lab and might be helpful for future reference:

- Chapter 4.3, "Configuring User Self-Service" of the *OpenIDM Integrator's Guide*.
- Chapter 20, "Sending Email" of the *OpenIDM Integrator's Guide*.

Exercise 1: Configure an outgoing email service for OpenIDM

Goal

Each user self-service function (User Registration, Forgotten Username and Password Reset) has the option to use email as part of their function. Email is also used extensively in workflow.

In this exercise, you configure an outgoing email service for OpenIDM so that all self-service functions can use email as part of their self-service function.

Two options are provided: Gmail and the FakeSMTP utility that mimics a locally installed SMTP service. Gmail is the better option because you can interact directly with emails that are generated during Self Service functions and workflows. You need an active Gmail account to use that option; if you don't have one, you can use FakeSMTP.

Objectives

Upon completion of this exercise, you should be able to:

- Configure OpenIDM to use Gmail or FakeSMTP
- Modify the external email configuration file
- View the email service configuration in OpenIDM
- Verify that the email service is active in the Felix console
- Verify that the email service is working with a REST command to send an email

Requirements

The following requirements are necessary before starting this exercise:

- Access to the hosted learning environment
- An active Gmail account if you choose to use Gmail
- The FakeSMTP utility if you choose to use FakeSMTP

References

The following references were used in the development of this exercise and might be helpful for future reference:

- Chapter 20, "Sending Email" of the *OpenIDM Integrator's Guide*.

Detailed Tasks

Task 1: Configure Gmail

If you want to use Gmail and you have a valid account, the instructions below explain how to configure it for OpenIDM. If you'd rather use FakeSMTP, skip ahead to the next task.

1. Temporarily enable Less Secure Apps in Google:

   a. Make sure you're logged into Gmail and go your **My Account** settings.

   b. Select **Sign-in & Security**.

   c. Select **Connected Apps and Sites**.

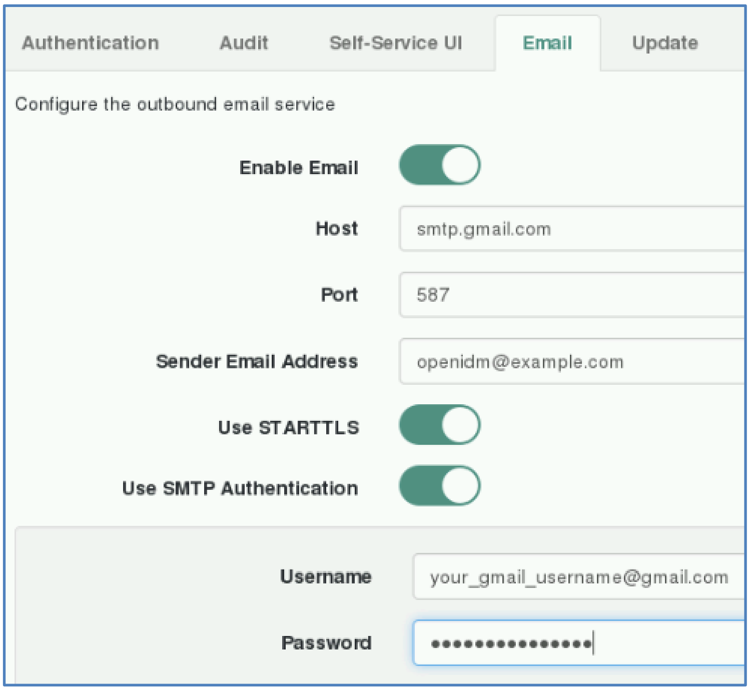
   d. Scroll down to **Allow Less Secure Apps** at the bottom of the page and set to **On**.

   e. When done testing, set **Less Secure Apps** back to Off.

2. Log into OpenIDM as the openidm-admin user and go to **Configure/System Preferences/Email**.

3. Enter the configuration informaton for Gmail and your account:

   - Enable Email: **On**
   - Host: **smtp.gmail.com**
   - Port: **587**
   - Sender Email Address: **openidm@example.com**
   - Use Start TLS: **On**
   - Use SMTP Authentication: **On**
   - Username: **Your Gmail address**
   - Password: **Your Gmail password**



Configure the outbound email service

Enable Email

Host: smtp.gmail.com

Port: 587

Sender Email Address: openidm@example.com

Use STARTTLS

Use SMTP Authentication

Username: your_gmail_username@gmail.com

Password: •••••••••••••••

Task 2: Configure FakeSMTP

If you choose not to use Gmail, use the FakeSMTP utility instead.

1. Unzip the `/opt/forgerock/software/fakesmtp-latest.zip` file to your `/opt` directory.

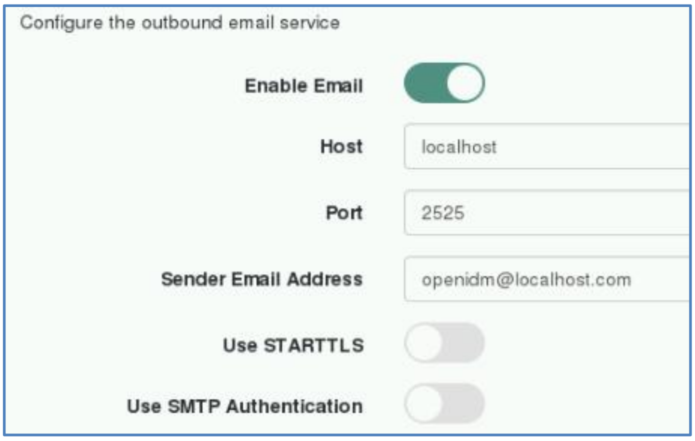
2. Start the FakeSMTP utility:

```
$ java -jar /opt/fakeSMPT-latest.jar
```

3. Set the Listening Port to 2525 then start the server.

4. Configure OpenIDM to use FakeSMTP:

- Enable Email: **On**
- Host: **localhost**
- Port: **2525**
- Sender Email Address: **openidm@example.com**



Configure the outbound email service

Enable Email ☑

Host: localhost

Port: 2525

Sender Email Address: openidm@localhost.com

Use STARTTLS □

Use SMTP Authentication □

5. When OpenIDM sends an email, it appears in the Last Message tab. Password emails are encoded so you need to use an Internet decoding service to see it.

Task 3: Verify the email service is working by sending mail over REST

Send email using the REST API by sending an HTTP POST to `/openidm/external/email`, to test that your configuration works. You pass the message parameters as part of the POST payload, URL encoding the content as necessary.

1. Open a new terminal window, if necessary.

2. Use the `curl` command to test email using the REST API.

```
$ curl --cacert self-signed.crt --header "Content-
Type: application/json" --header "X-OpenIDM-
Username: openidm-admin" --header "X-OpenIDM-
Password: openidm-admin" --request POST --data
'{"from":"openidm@example.com","to":"your_email@exam
ple.com","subject":"Test","body":"Test"}'
"http://localhost:8080/openidm/external/email?_actio
n=send" | jq .
```

You should get the following response:

```
{
  "status": "OK"
}
```

3. You should see the test email in your Gmail account or in FakeSMTP, depending on which you configured.

Task 4: Verify the external email service is active from the Felix console

1. Return to the terminal window where the Felix console is monitoring the OpenIDM instance.

2. Run the **`scr list`** command in the Felix console and verify the `external.email` bundle is active. For example, search the list for the following similar message:

   ```
   [ 19] [active ] org.forgerock.openidm.external.email
   ```

   Note that the starting number will likely be different for your environment.

Exercise 2: Enable email-based self-registration, password reset, and username retrieval

Goal

In this exercise you enable and configure the user registration self-service function to allow anonymous users to create their own accounts using email verification.

Objectives

Upon completion of this exercise, you should be able to:

- View the existing self-service files.
- Enable the user registration self-service function.
- Configure the email validation page.
- Verify that user registration is working.
- Enable and verify password reset.
- Enable and verify forgotten username retrieval.
- Enable Site Identification
- Reorder the user registration stages.

Requirements

Before starting this exercise, you need to complete the previous exercise: *Configure an outgoing email service for OpenIDM*.

References

The following references were used in the development of this exercise and might be helpful for future reference:

- Chapter 4.3, "Configuring User Self-Service" of the *OpenIDM Integrator's Guide*.

Detailed Tasks

Task 1: View the existing self-service configuration files

Out-of-the-box the user self-service functions are disabled for user registration, forgotten username and password reset. However, there is one configuration files in the OpenIDM configuration directory related to Knowledge-based Authentication (KBA) questions:

- `selfservice.kba.json` (KBA)

When you enable additional functions, the corresponding configuration files are created in the OpenIDM conf directory:

- `selfservice-registration.json` (User registration)
- `selfservice-reset.json` (Password reset)
- `selfservice-username.json` (Username lookup)

Note that KBA is a subset of the user self-service functions and can be enabled or disabled for the user registration and password reset functions.

1. Open a terminal window and change to the OpenIDM instance directory where you are testing the user self-service functions.

2. List the current self-service configuration files in the OpenIDM configuration directory.

```
$ ls conf/self*
conf/selfservice.kba.json
```

You should only find the `selfservice.kba.json` configuration file that contains the default KBA questions that are used when enabled on user registration and password reset.

3. View the contents of the file.

KBA will be revisted in a later exercise.

Task 2: Enable the user registration self-service function

Enable the user registration self-service function from the Admin UI.

Note that it is also possible to enable and configure the user registration self-service function by copying the `selfservice-registration.json` configuration file from the `samples/misc` directory, modifying the file and then copying the file to the OpenIDM configuration directory (similar to method used in the previous exercise to enable the email service).

1. Log in to the Admin UI as the administrator user.

2. Select **Configure** and then **User Registration** from the main menu.

The User Registration page appears where you can enable and configure the registration steps.

3. Enable user registration in the Admin UI.

   Note all options (steps) are enabled except for the reCAPTCHA for Registration option.

4. Return to the terminal window and view the contents of the `selfservice-registration.json` configuration file in the OpenIDM configuration directory.

5. Compare the stages in the configuration file with the user registration page in the Admin UI.

   Which stages are enabled?

   (Answer: Email Validation (`emailValidation`), User Details (`userDetails`), KBA Stage (`kbaSecurityAnswerDefinitionStage`), and Registration Form (`selfRegistration`))

   The configuration file references stages, which correspond to the registration steps in the Admin UI. These are also called options at times as some of the steps control the behavior of the user registration process. For example, the User Detail step allows you to modify the Identity Email Field associated with user registration.

Task 3: Configure the email validation stage

Edit the email validation step (stage) using the Admin UI and configure the outgoing email information.

1. Select Email Validation step on the User Registration page of the Admin UI.

   The Configure Validation Email window opens, which allows you to configure the options of the email validation step.

2. Edit the following settings:

   `Email From:` **`no-reply-openidm@example.com`**

   `Email Verification Link:`
   `http://localhost:8080/#register/`

3. Save the changes.

4. Verify the changes in the related configuration file.

5. Edit the email message to simplify the received mail for the lab. Note that this is to make it easier to cut-and-paste from the FakeSMTP application when testing.

   a. Select the Email Validation step on the User Registration page of the Admin UI.

   b. Remove the English and French email messages in the Email Message section.

   c. Add a simple English email message using the **`en`** locale and **`%link%`** localized text.

   d. Save the changes.

   e. Verify the changes in the related configuration file. For example:

```
"messageTranslations" : {
                "en" : "%link%"
            },
```

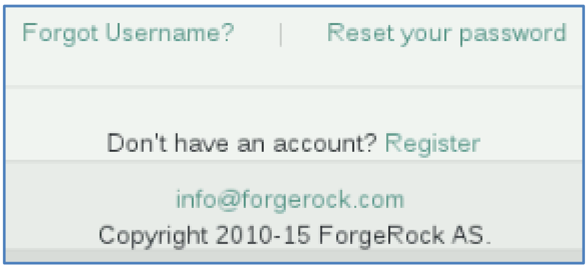
Task 4: Verify user registration is working

Use a new private browser window to test email-based user registration is working properly.

1. In a new incognito or private browser window, depending on the browser, go to the Self-Service UI page.

   `http://localhost:8080/`

2. Verify the login page contains a link to register. For example:



Forgot Username? | Reset your password

Don't have an account? Register

info@forgerock.com
Copyright 2010-15 ForgeRock AS.

3. Select the **Register** link.

4. Enter your Gmail address or any email address depending on whether you are using Gmail or FakeSMTP.

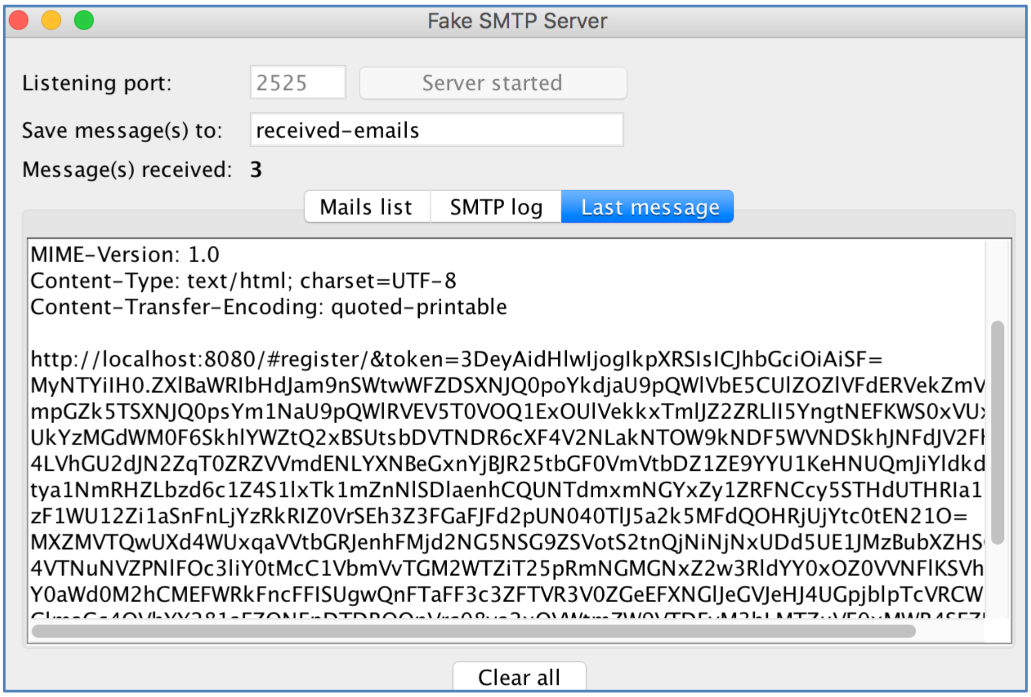
You should get a new browser page that indicates the email has been sent and to return to the login page, if everything is configured properly. If not configured properly, for example the email service is not configured, you will get an error message.

Task 5: Decode the encoded email in FakeSMTP

If you are using FakeSMTP, the email you receive will be encoded and will need to be decoded.

1. Access the email from the Fake SMTP Server and look for a message titled "Register a new account". In Gmail, the email will appear in your inbox.

2. Go to the **Last message** window of the FakeSMTP application to view the last message. For example:



Listening port: 2525

Save message(s) to: received-emails

Message(s) received: 3

```
MIME-Version: 1.0
Content-Type: text/html; charset=UTF-8
Content-Transfer-Encoding: quoted-printable

http://localhost:8080/#register/&token=3DeyAidHlwIjogIkpXRSIsICJhbGciOiAiSF=
MyNTYiIH0.ZXlBaWRIbHdJam9nSWtwWFZDSXNJQ0poYkdjaU9pQWlVbE5CUlZOZlVFdERVekZmV
mpGZk5TSXNJQ0psYm1NaU9pQWlRVEV5T0VOQ1ExOUlVekkxTmlJZ2ZRLlI5YngtNEFKWS0xVU
UkYzMGdWM0F6SkhlYWZtQ2xBSUtsbDVTNDR6cXF4V2NLakNTOW9kNDF5WVNDSkhJNFdJV2F
4LVhGU2dJN2ZqT0ZRZVVmdENLYXNBeGxnYjBJR25tbGF0VmVtbDZ1ZE9YYU1KeHNUQmJiYldkd
tya1NmRHZLbzd6c1Z4S1lxTk1mZnNlSDlaenhCQUNTdmxmNGYxZy1ZRFNCcy5STHdUTHRIa1
zF1WU12Zi1aSnFnLjYzRkRIZ0VrSEh3Z3FGaFJFd2pUN040TlJ5a2k5MFdQOHRjUjYtc0tEN21O=
MXZMVTQwUXd4WUxqaVVtbGRJenhFMjd2NG5NSG9ZSVotS2tnQjNiNjNxUDd5UE1JMzBubXZHS
4VTNuNVZPNlFOc3liY0tMcC1VbmVvTGM2WTZiT25pRmNGMGNxZ2w3RldYY0xOZ0VVNFlKSVh
Y0aWd0M2hCMEFWRkFncFFISUgwQnFTaFF3c3ZFTVR3V0ZGeEFXNGlJeGVJeHJ4UGpjblpTcVRCW
```

Note that the link is encoded using the "quoted-printable" Content Transfer Encoding method. You need to copy the contents of the message and convert (decode) it to HTML (as a client would normally do):

a. Copy the http portion of the message to the clipboard. Note that this is the `%link%` value sent by OpenIDM. For example, (encoded view):

```
http://localhost:8080/#register/&token=3DeyAidHlwIjo
gIkpXRSIsICJhbGciOiAiSF=
MyNTYiIH0.ZXlBaWRIbHdJam9nSWtwWFZDSXNJQ0poYkdjaU9pQW
lVbE5CUlZOZlVFdERVekZmV=
mpGZk5TSXNJQ0psYm1NaU9pQWlRVEV5T0VOQ1ExOUlVekkxTmlJZ
2ZRLnhJVnJDa2NHUmZDS0pD=
ZVU0SkJaY3pidU8zeGFqcVEtSkRYQXBsazlTTXhNbGNWcm5uN21X
dDNHWC1BUS1IWV9hcGhGVUh=
BUGZKSFk1dW10NFpjQ0Q3dHdHWHptbW5weUxWUUlBdlpBNVNGV2w
2ZHEwZ19EWFB2SHB1b212S2=
Q5ZEZqazBsWEhpZFhsM3RRS2d0dXpoYlljOTAweVBvTmphcFBueT
Y5SmUwdy54QWlpeU43djY1O=
UlmVkZBVTJVTkZRLmRzeE5UQnlhLVFKaU0xQTRFdnF2Wm5aZ1dHb
ExuZF9RVEM0V3RVVm1CQ29V=
S2dpOHp3OS1DdW9SN0JFVmNwT2J1cFgtc3dNbFVRS3FrUVk0Q2k0
NHhMcm9YNFY0UzZhZzVGRDN=
zdFZ6WjBYcmdJOXh2R0dXbkxFWWEwSDA2c2gxcEpaY2d6Sk5PcjV
zNi1zc0RvVm9qMWpuSUhiSU=
lnQWhZTURQVGI3VVR4dGxoZUxKVi1fUXYzZ0tkSi1rbDVQMHRUQ1
BDb3RnbFNuQU0zTzdyOXlkb=
W5sVlRkbWh2Z2hZdng1Yy1HbVh3TTd1allQSEJoWHRYQ1ZKRXZlN
2Z2LWdiNUFBNTY0NlR1MGJk=
TVFGNGh2LXpJaS1wUEJUbzBGd203eDlsZ1FXTWRGNlJnU1M2R1R1
czgwN1B4TU5XaHgzLmhFYnJ=
hVWVRMlp4Ykh3c08yM1h2Q2c.hxdE5k45cCPNlpoJCyn4TQufcwV
eDh3N-3I9rynmJFA&code=
=3D0b52b8f2-69a2-42ce-9af8-29a3b7c12b19
```

b. Search the Internet for a converter. For example, search on the string:

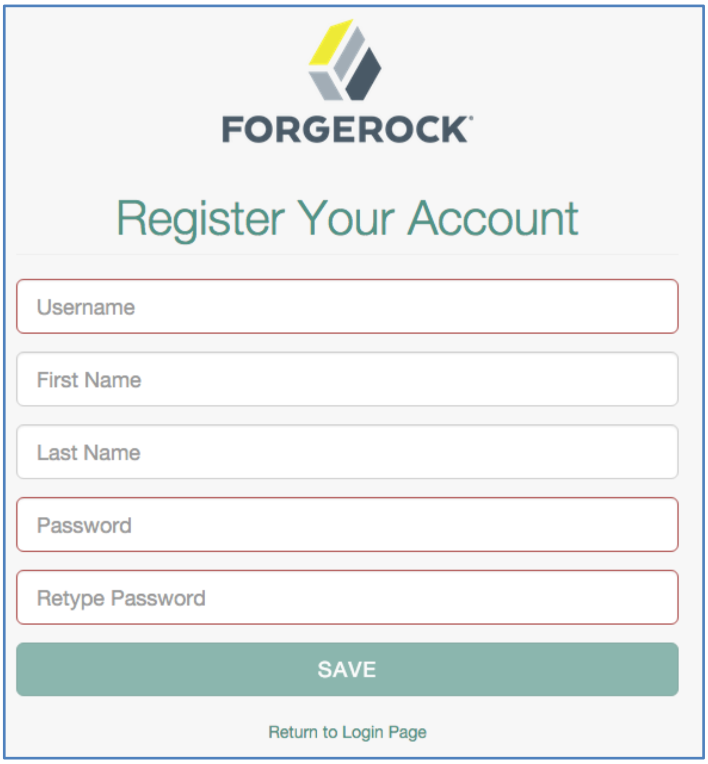
`quoted-printable to HTML`

c. Copy the message to the converter and decode the message. For example, (decoded view):

```
http://localhost:8080/#register/&token=eyAidHlwIjogI
kpXRSIsICJhbGciOiAiSFMyNTYiIH0.ZXlBaWRIbHdJam9nSWtwW
FZDSXNJQ0poYkdjaU9pQWlVbE5CUlZOZlVFdERVekZmVmpGZk5TS
XNJQ0psYm1NaU9pQWlRVEV5T0VOQ1ExOUlVekkxTmlJZ2ZRLnhJV
nJDa2NHUmZDS0pDZVU0SkJaY3pidU8zeGFqcVEtSkRYQXBsazlTT
XhNbGNWcm5uN21XdDNHWC1BUS1IWV9hcGhGVUhBUGZKSFk1dW10N
FpjQ0Q3dHdHWHptbW5weUxWUUlBdlpBNVNGV2w2ZHEwZ19EWFB2S
HB1b212S2Q5ZEZqazBsWEhpZFhsM3RRS2d0dXpoYlljOTAweVBvT
mphcFBueTY5SmUwdy54QWlpeU43djY1OUlmVkZBVTJVTkZRLmRze
E5UQnlhLVFKaU0xQTRFdnF2Wm5aZ1dHbExuZF9RVEM0V3RVVm1CQ
29VS2dpOHp3OS1DdW9SN0JFVmNwT2J1cFgtc3dNbFVRS3FrUVk0Q
2k0NHhMcm9YNFY0UzZhZzVGRDNzdFZ6WjBYcmdJOXh2R0dXbkxFW
WEwSDA2c2gxcEpaY2d6Sk5PcjVzNi1zc0RvVm9qMWpuSUhiSUlnQ
WhZTURQVGI3VVR4dGxoZUxKVi1fUXYzZ0tkSi1rbDVQMHRUQ1BDb
3RnbFNuQU0zTzdyOXlkbW5sVlRkbWh2Z2hZdng1Yy1HbVh3TTd1a
llQSEJoWHRYQ1ZKRXZlN2Z2LWdiNUFBNTY0NlR1MGJkTVFGNGh2L
XpJaS1wUEJUbzBGd203eDlsZ1FXTWRGNlJnU1M2R1R1czgwN1B4T
```

```
U5XaHgzLmhFYnJhVWVRMlp4Ykh3c08yM1h2Q2c.hxdE5k45cCPNl
poJCyn4TQufcwVeDh3N-3I9rynmJFA&code=0b52b8f2-69a2-
42ce-9af8-29a3b7c12b19
```

3. Copy the registration link to the browser where you made the request. You should now get a registration page. For example:



FORGEROCK

Register Your Account

Username

First Name

Last Name

Password

Retype Password

SAVE

Return to Login Page

4. Complete the registration and save.

5. Select a security question, answer and save. Your account should be successfully registered.

   Note that you are asked a security question because by default the KBA stage is enabled when you enable user registration.

6. Return to login page link and log in as the new user using the username and password entered during registration.

7. View the user profile information, including the security questions.

8. Log out when done and close the browser window.

9. Return to the Admin UI and verify the new user is added to the repository as a managed/user.

a. Select Manage and then User to list all of the managed users in the repository.

b. Select the test user and view the user's information.

10. Return to the dashboard when done viewing the managed user.

Task 6: Enable and verify password reset

In this task you enable password notification using a new user account.

1. As the Admin user, create a new user account named user1 with a password of Password1.

2. After the user is created, enable password reset. Open **Configure/Password Reset**.

3. Click the **Enable Password Reset** button.

4. Log out as the openidm-admin user and then log in as `user1/Password1` to verify that the account is working correctly.

5. Log out as `user1` and on the login page, select **Reset your password**.

6. Enter **Username**, **First name**, and **Last Name** and click **Submit**.

7. In the FakeSMTP window, the email appears. Follow the instructions in Task 4 Step 6 to view the email message. If you are using Gmail, just check your inbox.

8. Once you have decrypted the email, send it to the address shown in the email to reset your password.

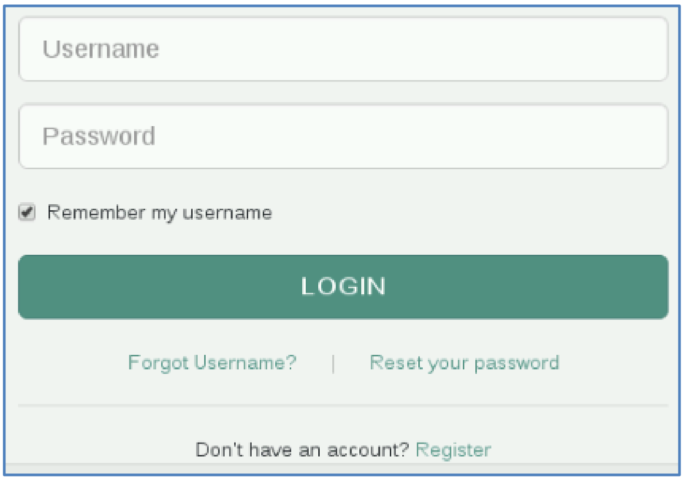
Task 7: Configure forgotten username retrieval

1. Log into OpenIDM as the `openidm-admin`:`openidm-admin` user.

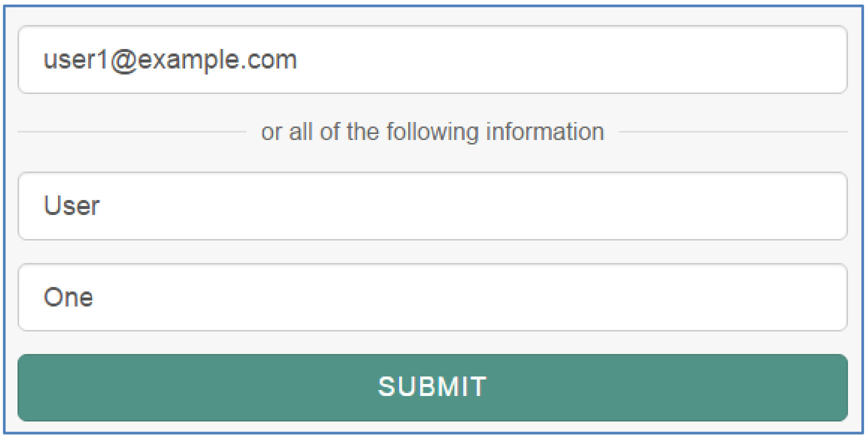
2. Open **Configure/Forgotten Username** and click **Enable Forgotten Username Retrieval**.

3. Choose whether you want to email the forgotten username or display it directly to the user, or both. In this example, the forgotten username is emailed to the user.

4. Test by logging out as the admin and logging back into Self Service with a user account you created earlier. Click the Forgot Username? link.



5. Enter the user's email or the first and last name, or both. This example shows the `user1` user created earlier.



user1@example.com

or all of the following information

User

One

6. Open Gmail or your FakeSMTP window and examine the Last Message tab. The email with the forgotten username appears.

Task 8: Reorder the user registration stages

It is possible to reorder the stages in the Admin UI or related configuration file to change the order of steps presented to the user while performing user registration.

However, even though you can reorder the stages, be careful in the order as some steps before others will not make sense to the end user and/or to the OpenIDM system. For example, you will get internal errors.

In this task, you reorder the steps using the Admin UI, view the configuration file changes, view the user registration page as a user and then put the steps back to their original order.

1. Return to the Admin UI page where you can edit the user registration steps. For example, the following is the original order:

   reCAPTCHA for Registration (not enabled)

   Email Validation (enabled)

   User Details (enabled)

   KBA Stage (enabled)

   Registration Form (enabled)

2. Reorder the stages by selecting any stage and dragging the stage before or after another stage.

   For example, drag the User Details step before the Email Validation step.

3. Go to a terminal window and view the related configuration file and compare the order to the Admin UI. They should match still.

4. Open a new incognito or private browser window, go to the Self-Service UI page and select the register link.

   The first register page should match the stage that is first in the user registration steps. For example, if you moved the User Details step before the Email Validation step, then you should see the page to enter registration information.

   Please note that this is just to demonstrate that you can reorder the steps; however, the order might not make sense to the end user or the sytem itself. For example, end users might see errors on the user registration page and system administrators will see errors in the Felix console.

5. Change the order of the user registration steps back to their original order (as noted in the first step of this task).

Exercise 3: Enable Google reCAPTCHA

Goal

In this exercise you configure Google reCAPTCHA for user registration, password reset, and forgotten username.

Objectives

Upon completion of this exercise, you should be able to:

- Register for Google reCAPTCHA
- Test reCAPTCHA for Forgotten Username

Requirements

This exercise assumes you have set up a test user (user1) and configured password reset, user registration, and forgotten username.

References

The following references were used in the development of this exercise and might be helpful for future reference:

- Chapter 4.3, "Configuring User Self-Service" of the *OpenIDM Integrator's Guide*.

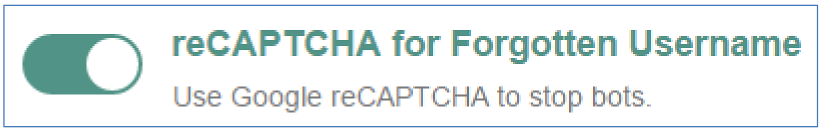
Detailed Tasks

Task 1: Register for Google reCAPTCHA

In this example you enable reCAPTCHA for forgotten username. The procedure for enabling it for password reset and user registration are the same.

1. Log in to OpenIDM as the `openidm-admin:openidm-admin` user and select **Configure/Forgotten Username**.

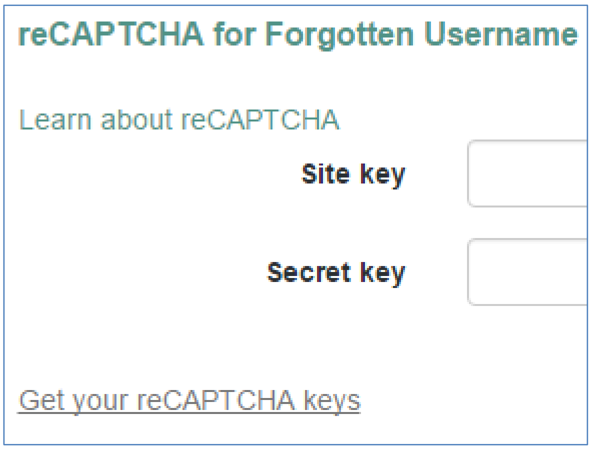
2. Click the button to enable reCAPTCHA:



reCAPTCHA for Forgotten Username

Use Google reCAPTCHA to stop bots.

3. Click the edit button to the right, then select **Get your reCAPTCHA keys**. Note: You need a Google account to retrieve the keys.



reCAPTCHA for Forgotten Username

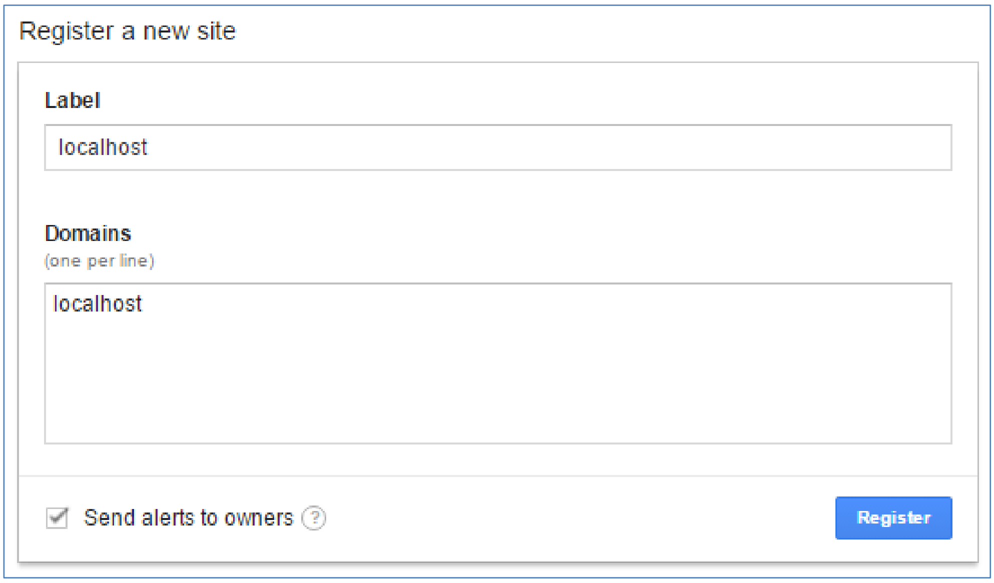
Learn about reCAPTCHA

Site key

Secret key

Get your reCAPTCHA keys

4. Enter **localhost** for the **Label** and **Domains** fields. You can use **localhost** for testing purposes. Then click **Register**.



Register a new site

Label

localhost

Domains

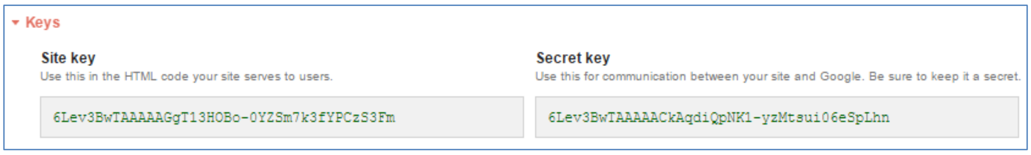
(one per line)

localhost

☑ Send alerts to owners

5. Make a note of your keys. This example uses:

- Site Key: `6Lev3BwTAAAAAGgT13HOBo-0YZSm7k3fYPCzS3Fm`
- Secret Key: `6Lev3BwTAAAAACkAqdiQpNK1-yzMtsui06eSpLhn`



Keys

Site key

Use this in the HTML code your site serves to users.

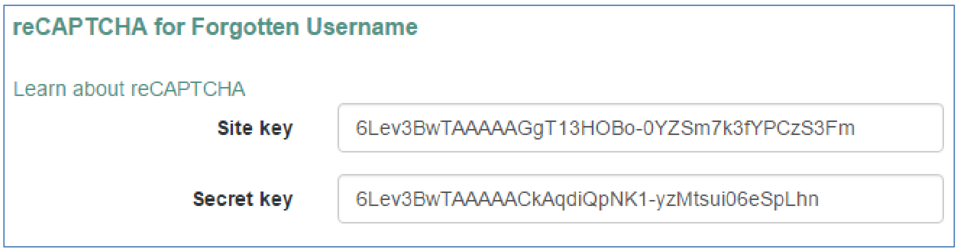
6Lev3BwTAAAAAGgT13HOBo-0YZSm7k3fYPCzS3Fm

Secret key

Use this for communication between your site and Google. Be sure to keep it a secret.

6Lev3BwTAAAAACkAqdiQpNK1-yzMtsui06eSpLhn

6. Go back to the edit screen for reCAPTCHA and enter the Site key and Secret key, then save your changes:



reCAPTCHA for Forgotten Username

Learn about reCAPTCHA

Site key: 6Lev3BwTAAAAAGgT13HOBo-0YZSm7k3fYPCzS3Fm

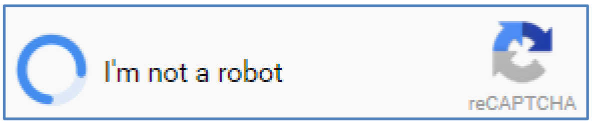
Secret key: 6Lev3BwTAAAAACkAqdiQpNK1-yzMtsui06eSpLhn

Task 2: Test in Self Service

1. Log out as the `openidm-admin` user then open the self-service URL (localhost:8080).

2. Before logging in, click **Forgot Username?**

3. The reCAPTCHA widget appears, which prevents you from proceeding until you click the **I'm not a robot** button:



4. Once you select **I'm not a robot**, you are directed to the **Retrieve Your Username** page where you can complete the process.

Exercise 4: Implement knowledge-based authentication password reset

Goal

In this brief exercise you configure and enable security questions for password reset.

Objectives

Upon completion of this exercise, you should be able to:

- Add security questions to a user's account.
- Enable KBA.
- Test KBA for password reset.

Requirements

You need a test user (user1) created earlier.

References

The following references were used in the development of this exercise and might be helpful for future reference:

- Chapter 4.3, "Configuring User Self-Service" of the *OpenIDM Integrator's Guide*.

Detailed Tasks

Task 1: Enable KBA

The OpenIDM Administrator can enable KBA for password reset or user registration. Just select Enable KBA in the Configuration page for password reset or user registration.

Task 2: Set up security questions

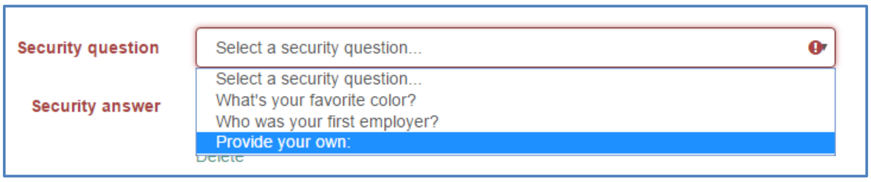
Users can configure their own security questions in **Profile/Security Questions**.

Visit http://goodsecurityquestions.com for advice on creating security questions.

1. Log in to Self Service as `user1:Password1`.

2. Select **Profile/Security Questions**.

3. Click **Add another question** your questions and answers. To enter your own question, select Provide your own and then enter the question in the field provided.


Security question
Security answer
Select a security question...
Select a security question...
What's your favorite color?
Who was your first employer?
Provide your own:
Delete

4. When done entering your questions and answers, save your changes.

Task 3: Test password reset with security questions

1. Log out of OpenIDM and log back in with the `user1` account.

2. Instead of logging in, click **Password Reset**.

3. When notified that the email was sent, open it in Gmail or the FakeSMTP utility and decrypt it using an external service.

4. When you access the decrypted email you are presented with security questions before being allowed to change the password.