Chapter 3: Self Service Configuration and Customization

Course Map

Chapter 1: Identity Provisioning

Chapter 2: Synchronization and Reconciliation

Chapter 3: Self-Service

- Lesson 11: Configure User Self-Service
- Lesson 12: Customize the User Interface

Chapter 4: Workflow Engine

Chapter 5: Password Management and Policies

Chapter 6: Miscellaneous

Chapter 7: Production

Summary

- Configure Self Service:
  1. Self Service Relative URL
  2. Outgoing Email
  3. Self-registration, password reset, and forgotten username
  4. Knowledge-based
- Customize the User Interface
  1. Prepare your system for customization
  2. Customize the UI theme
  3. Customize Self Service screens
- Approximate time: 180 minutes with breaks

Module 11: Configure User Self Service

Module Objectives

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

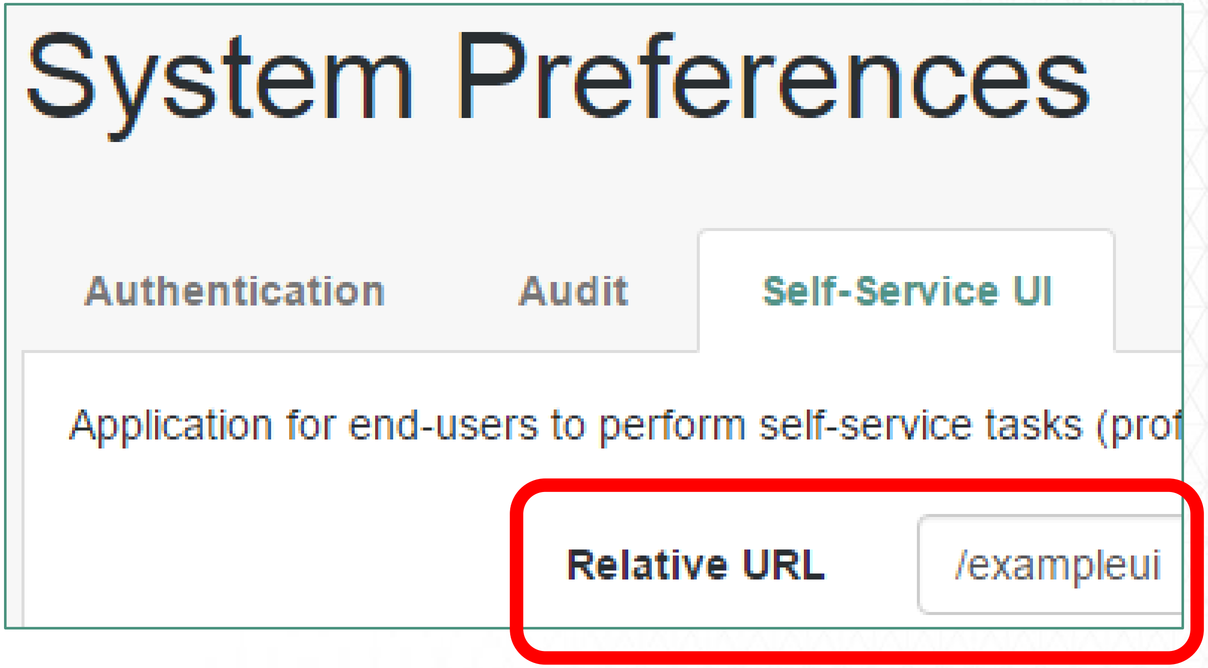
- Configure the self-service URL
- Configure the outbound email service
- Enable email-based self-registration
- Enable email-based password reset
- Enable email-based username retrieval
- Configure Google ReCAPTCHA options to protect from spam and abuse
- Implement knowledge-based authentication password reset

Configuring the Self Service URL

- The Self Service relative URL sets the context root
- For example, a new UI theme can bet set as the context root
- There are two ways to set the relative URL:
  - In the Admin UI
  - In the ui.context-selfservice.json file

Configuring the Relative URL in the Admin UI

1. In the Admin UI go to System Preferences, select the Self-Service UI tab
2. Enter the relative URL



System Preferences

Application for end-users to perform self-service tasks (prof

Relative URL /exampleui

Setting the Relative URL in the UI Configuration File

- The Relative URL is set in the UI configuration file
- Set the value of urlContextRoot to your relative URL

openidm/conf/ui.context-selfservice.json

```
{
    "enabled" : true,    "urlContextRoot" : "/exampleui",
    "defaultDir" : "&{launcher.install.location}/ui/...
    "extensionDir" : "&{launcher.install.location}/ui..."
}
```

Enable email-based self registration

- Each Self Service function (registration, password reset, and forgotten username) has the option to use email as part of the process
- You can configure an external outgoing email service to use in Self Service
- Steps:
  - Modify an external email configuration file external.email.json
  - Modify the Admin UI for email configuration
  - Verify with a test email

Modify the External Email Config File

- This class uses FakeSMTP to mimic an SMTP server
- To modify external.email.json:
  - Copy an existing email config file from samples/misc/external.email.json to your openidm directory
  - Modify the file for the parameters required by the server: host, port, auth, starttls
  - Move the file into the openidm/conf directory

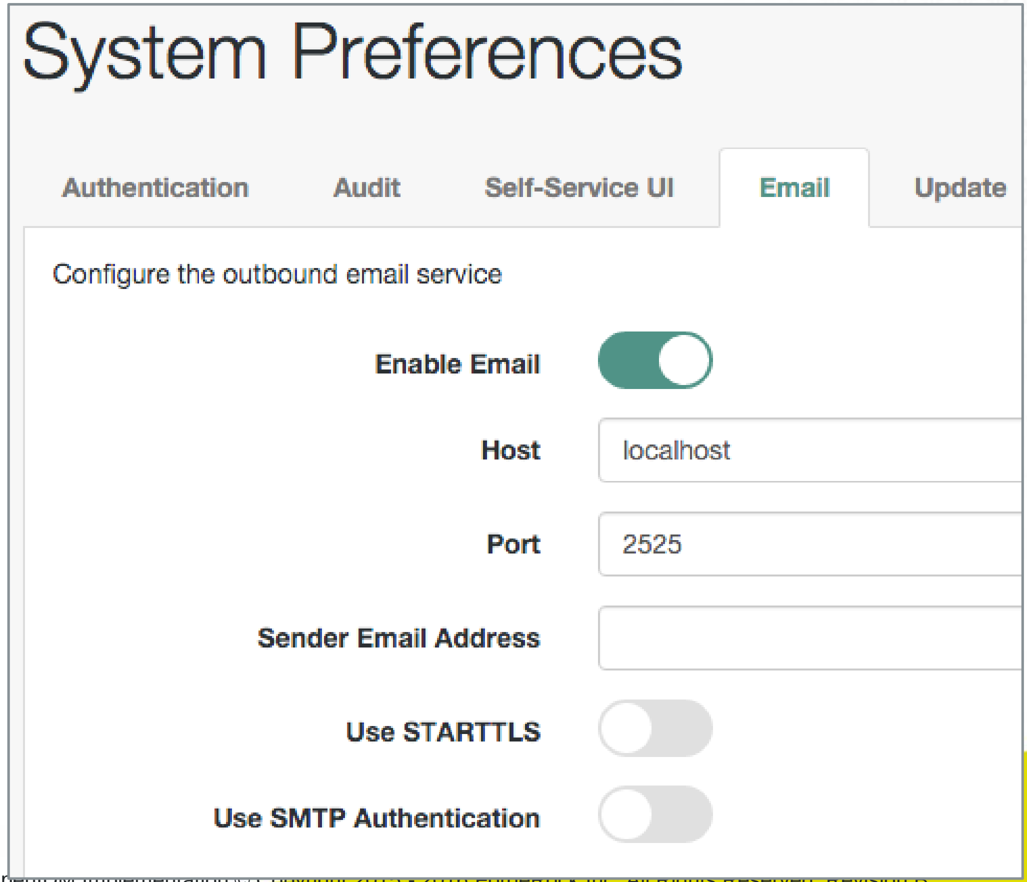
Edit external.email.json

- The external email file is initially configured for smtp.google.com
- Change for your SMTP server (FakeSMTP on localhost in this example)

```
"host" : "localhost",
"port" : "2525",
"debug" : false,
"auth" : {
    "enable" : false,
    "username" : "openidm-admin",
    "password" : "openidm-admin"
},
"starttls" : {
    "enable" : false
}
```

Configure Email in the Admin UI

- Enabling email in the Admin UI creates the conf/external.email.json file
- Authorization data is encrypted when the file is created in the conf directory



System Preferences

Configure the outbound email service

Enable Email: on

Host: localhost

Port: 2525

Sender Email Address:

Use STARTTLS: off

Use SMTP Authentication: off

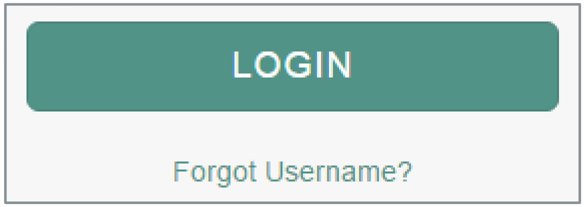
Enable Email-based Self Service

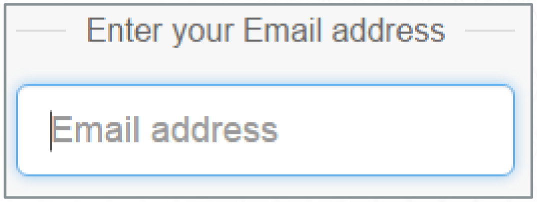
- To add email authorization to user registration, password reset, and forgotten password set the appropriate parameters to true in conf/ui-configuration.json
- You can also use the Admin UI
- Once enabled each function generates an email for completion
- Setting to false allows users to complete the function in the UI

```
"configuration" : {
    "selfRegistration" : true,
    "passwordReset" : true,
    "forgotUsername" : true,
...
```

Example: Forgotten Username

- After enabling Forgotten username and configuring Email, log into Self Service
- Click the Forgot Username link
- Enter a valid email address or other identifying information



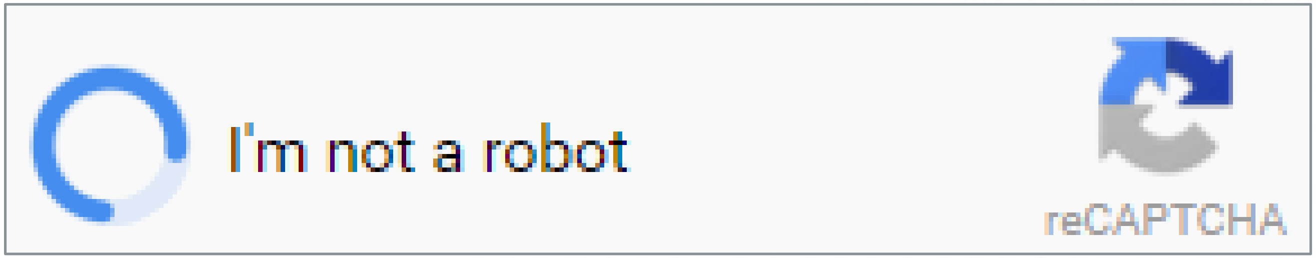


```
From: info@admin.org
To: user1@example.com
Message-ID: <550722779.0.1461627727828.JavaMail.name@host>
Subject: Account Information - username
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit

<h3>Username is:</h3><br />user1
```

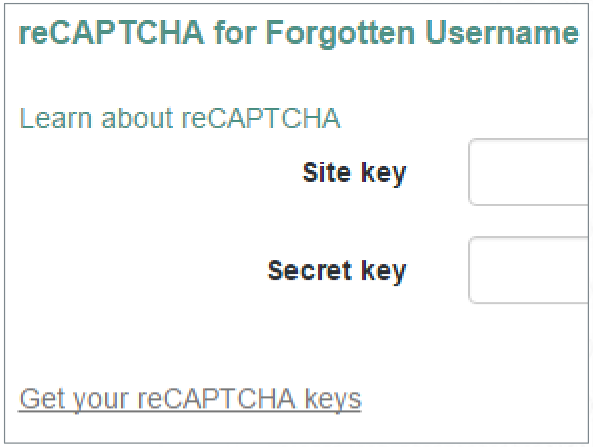
Enable Google reCAPTCHA

- reCaptcha presents users with a graphical image that they must select before proceeding with password reset, user registration, or forgotten username.
- You can configure reCAPTCHA for any or all of the Self Service functions.
- reCAPTCHA requies Google site keys.



Steps to Enable reCAPTCHA

1. Select Enable reCAPTCHA for password reset, user registration, or forgotten username.
2. Select Get your reCAPTCHA keys.
3. Register the site. You can use localhost/localhost for testing.
4. Record your keys and enter in the OpenIDM reCAPTCHA screen.
5. Test in the Self Service UI.



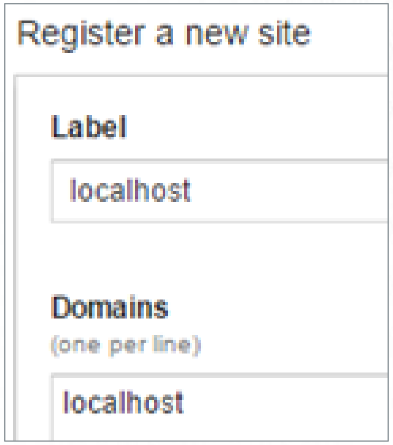
reCAPTCHA for Forgotten Username

Learn about reCAPTCHA

Site key

Secret key

Get your reCAPTCHA keys



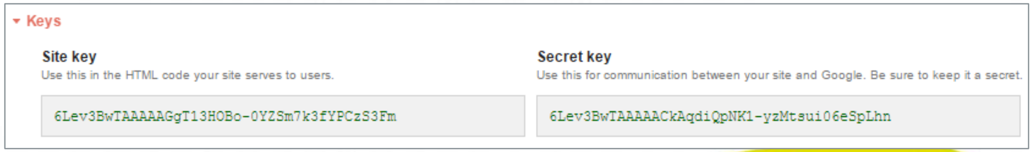
Register a new site

Label

localhost

Domains
(one per line)

localhost



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.

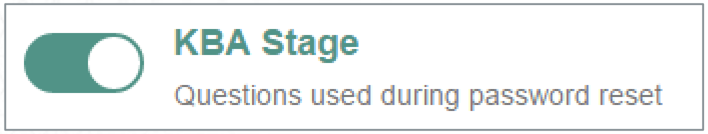
6Lev3BwTAAAAACkAqdiQpNK1-yzMtsui06eSpLhn

Knowledge-Based Password Reset

- You can enable security questions for registration, password reset, or forgotten username.
- The OpenIDM Admin enables KBA for user registration or password reset.
- Users craft their own security questions in Self Service Profile/Security Questions.
- An initial set of security questions can be configured by the Admin in selfservice.kba.conf.

Enabling and Configuring KBA

- To enable KBA, the Admin simply selects KBA Stage:



KBA Stage

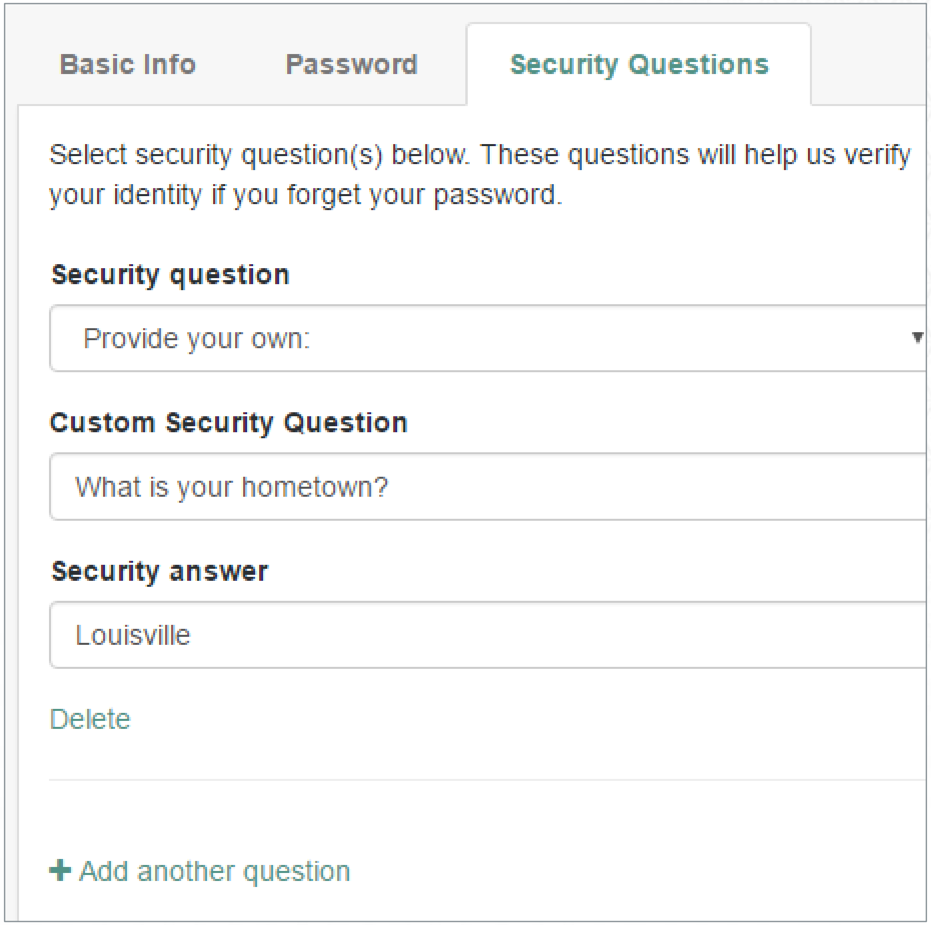
Questions used during password reset

- The Admin sets initial security questions in conf/selfservice.kba.conf:

```
     "kbaPropertyName" : "kbaInfo",
     "questions" : {
         "1" : {
             "en" : "What's your favorite color?",
             "en_GB" : "What's your favorite colour?",
             "fr" : "Quelle est votre couleur préférée?"
         },
         "2" : {
             "en" : "Who was your first employer?"
         }
     }
```

Configuring Security Questions

- Once KBA is enabled, users can customize questions and answers
- In Self Service go to Profile/Security Questions
- Select **Provide your own** and then enter your custom question and answer



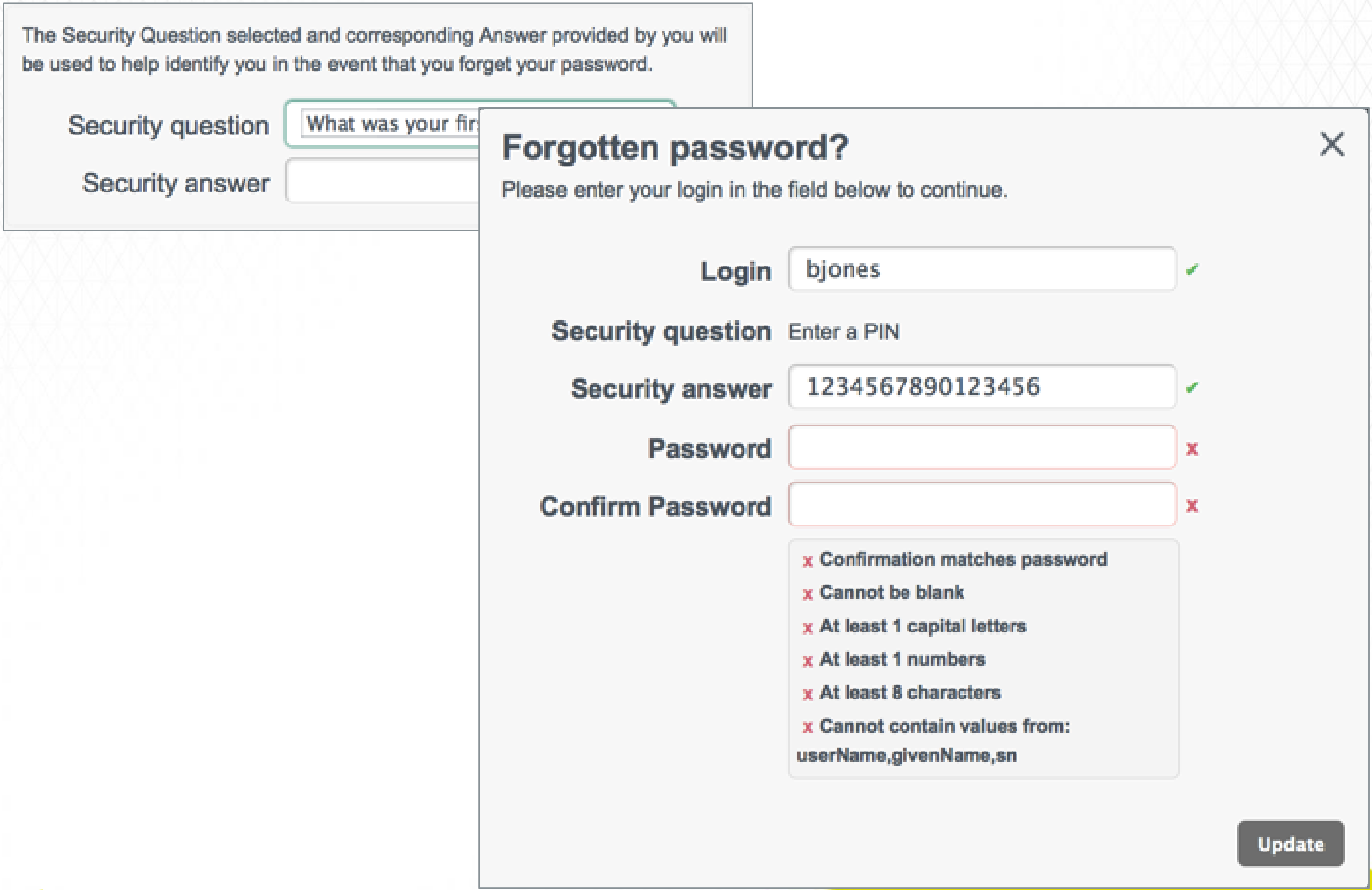
Select security question(s) below. These questions will help us verify your identity if you forget your password.

Security question: Provide your own:

Custom Security Question: What is your hometown?

Security answer: Louisville

Configure security questions



The Security Question selected and corresponding Answer provided by you will be used to help identify you in the event that you forget your password.

Security question: What was your fir

Security answer:

Forgotten password?

Please enter your login in the field below to continue.

Login: bjones ✓

Security question: Enter a PIN

Security answer: 1234567890123456 ✓

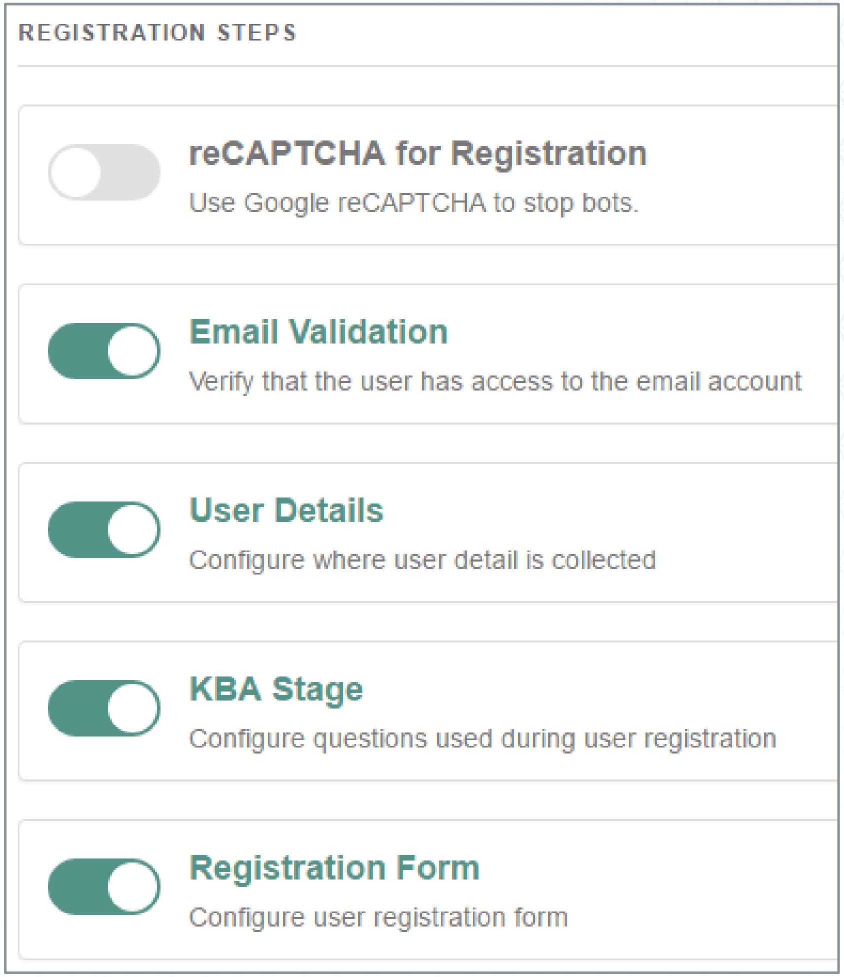
Password: ✗

Confirm Password: ✗

- ✗ Confirmation matches password
- ✗ Cannot be blank
- ✗ At least 1 capital letters
- ✗ At least 1 numbers
- ✗ At least 8 characters
- ✗ Cannot contain values from: userName,givenName,sn

Re-order User Registration Stages

- You can re-order the registration stages that are presented to a user with the Admin GUI.
- To re-order registration steps, go to Configuration/User Registration and drag the steps to the desired order.
- Be careful to ensure the order of stages makes sense to users and does not generate system errors.



REGISTRATION STEPS

○ **reCAPTCHA for Registration**
Use Google reCAPTCHA to stop bots.

● **Email Validation**
Verify that the user has access to the email account

● **User Details**
Configure where user detail is collected

● **KBA Stage**
Configure questions used during user registration

● **Registration Form**
Configure user registration form

Lab: Configure User Self Service

- Perform the following exercises in the Student Workbook:
  1. Configure outgoing email
  2. Enable email-based self-registration, password reset, and forgotten username lookup
  3. Enable Google reCAPTCHA
  4. Implement knowledge-based password reset
- Approximate time: 30 minutes

Module 11: Configure user self-service

End of module

Module 12: Customize the End User Interface

Module Objectives

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

- Prepare your system for customization
- Customize the UI theme
- Customize Self Service

Introduction

- The OpenIDM UI is fully customizable for branding and locale
- You can also modify Self Service templates to change theme, branding, and information gathered
- You can apply free/open source themes, logos, and other resources
  - http://www.wrapbootstrap.com
- Customization requires an understanding of HTML, Javascript/jQuery, XML, and other common web development technologies

Preparing OpenIDM for Customization

- Customizations are deployed to an extensions directory for Admin and Self Service
  - Admin: openidm_home/ui/admin/extensions
  - Self Service: openidm_home/ui/selfservice/extension
- OpenIDM automatically looks in the extension directory, if present, for customizations
- Templates are stored in:
  - admin/extensions/default/templates
  - selfservice/extension/default/templates

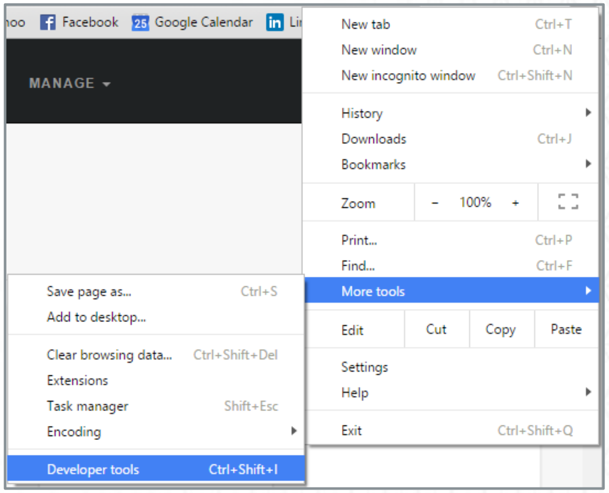
Setting Up the extension directory

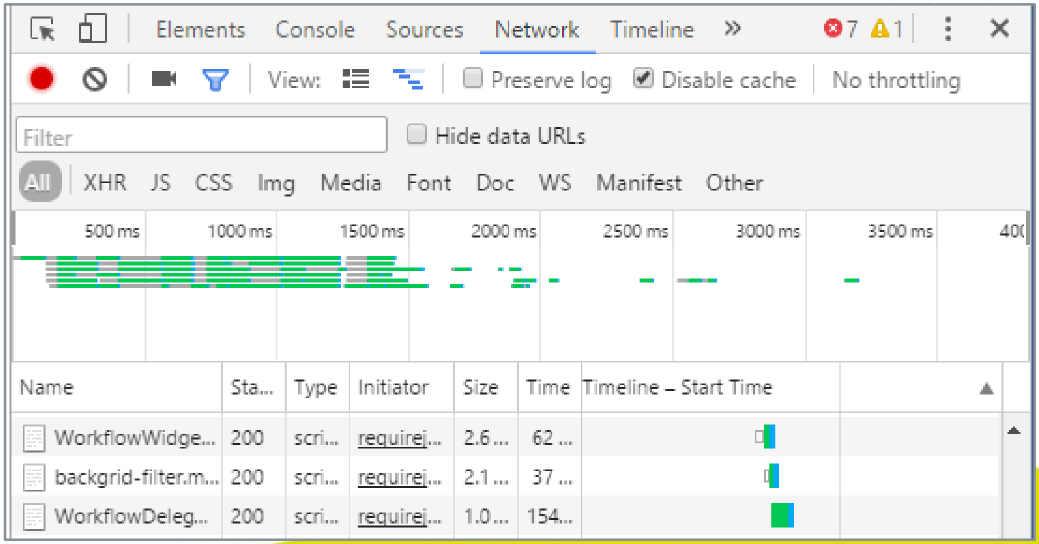
- Change to the ui directory in your OpenIDM project
- Create the extension directory
- Copy contents of the existing selfservice/default directory to the extension directory

```
$ cd /opt/openidm/ui
$ mkdir selfservice/extension
$ cp -r selfservice/default/. extension
```

Optional: Use Google Developer Tools to Monitor Customizations

- Start OpenIDM in Chrome, go to Tools/MoreTools/Developer tools
- Select Network tab
- Select Disable Cache
- Monitor activity while working in OpenIDM





□ Preserve log ☑ Disable cache No throttling

□ Hide data URLs

Name	Sta...	Type	Initiator	Size	Time	Timeline – Start Time
WorkflowWidge...	200	scri...	requirej...	2.6 ...	62 ...	
backgrid-filter.m...	200	scri...	requirej...	2.1 ...	37 ...	
WorkflowDeleg...	200	scri...	requirej...	1.0 ...	154...	

Customize the UI Theme

- You can change the UI theme by copying theme elements to the extension directory and modifying the appropriate files
- This section covers:
  - Changing the logos
  - Changing the UI language
  - Changing account labels

Change the Login Logo

- To change the login logo for Self Service:
  1. Copy your logo file to ui/selfservice/extension/images
  2. Make a note of the dimensions in pixels
  3. Edit the OpenIDM conf/ui-themeconfig.conf file to use the new image:

```
"settings" : {
  "loginLogo" : {
    "src" : "images/logoFR.png",
    "title" : "ForgeRock",
    "alt" : "ForgeRock",
    "height" : "99px",
    "width" : "92px"
  },
```

Change the Self Service Banner Logo

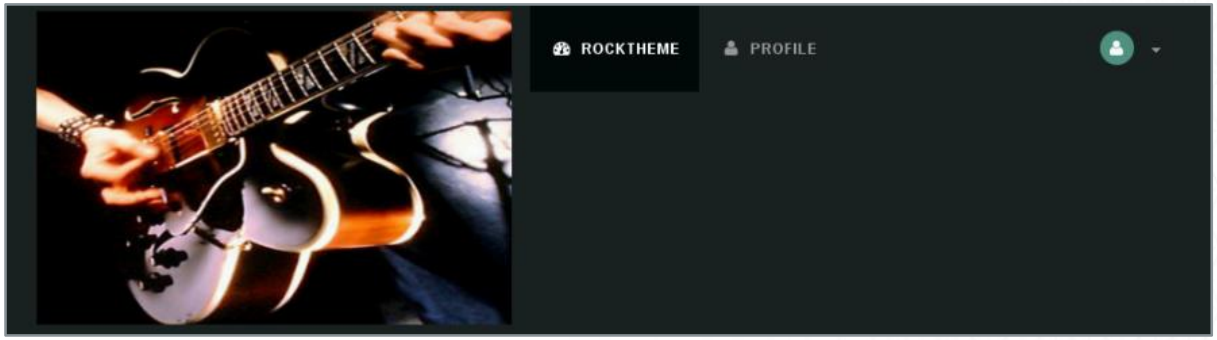
- To change the branding/banner logo for Self Service:
  1. Copy your logo file to ui/selfservice/extension/images
  2. Edit the OpenIDM conf/ui-themeconfig.conf file to use the new image:

```
"settings" : {
  "logo" : {
    "src" : "images/GuitarScreehshot.png",
    "title" : "ForgeRock",
    "alt" : "ForgeRock"
  },
```

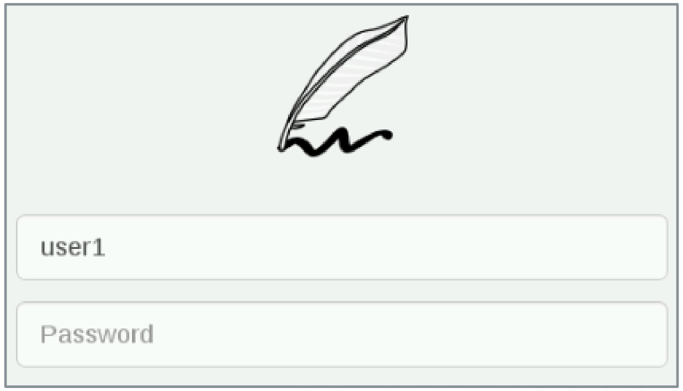
View the Changes

- Once your changes are made they are visible in the Self Service UI:

Dashboard



Login



Change the UI Language

- translation.json file defines the language used for text-based UI components
- Each locale has a separate translation.json file
- Translated strings are reflected in the UI
- Use Google Translate to translate strings

ui/selfservice/extension/locales/fr

```
...
"LoginTemplate" : {
  "loginRemember" : "Mémoriser mon nom d'utilisateur",
  "noAccountQuestion" : "Ne pas avoir un compte?",
  "registerAccount" : "Registre",
  "problemLoggingQuestion" : "Ayant des problèmes de connexion?",
  "resetPassword" : "Réinitialisez votre mot de passe",
  "loginWithOpenAM": "Connexion avec OpenAM",
  "forgotUsername" : "Identifiant oublié?"
},
...
```

Steps for Changing UI Language

- This example changes the language for the Self Service login page:

1. Copy the en (English) locale directory to fr (French)

```
$ cd selfservice/extension/locales
$ cp -r en fr
```

2. Open fr/translation.json and search for LoginTemplate

```
"loginRemember" : "Remember my
username",
```

3. Use Google Translate to translate the strings into French

```
"loginRemember" : "Mémoriser mon nom
d'utilisateur",
```

4. Search for copyright and translate that

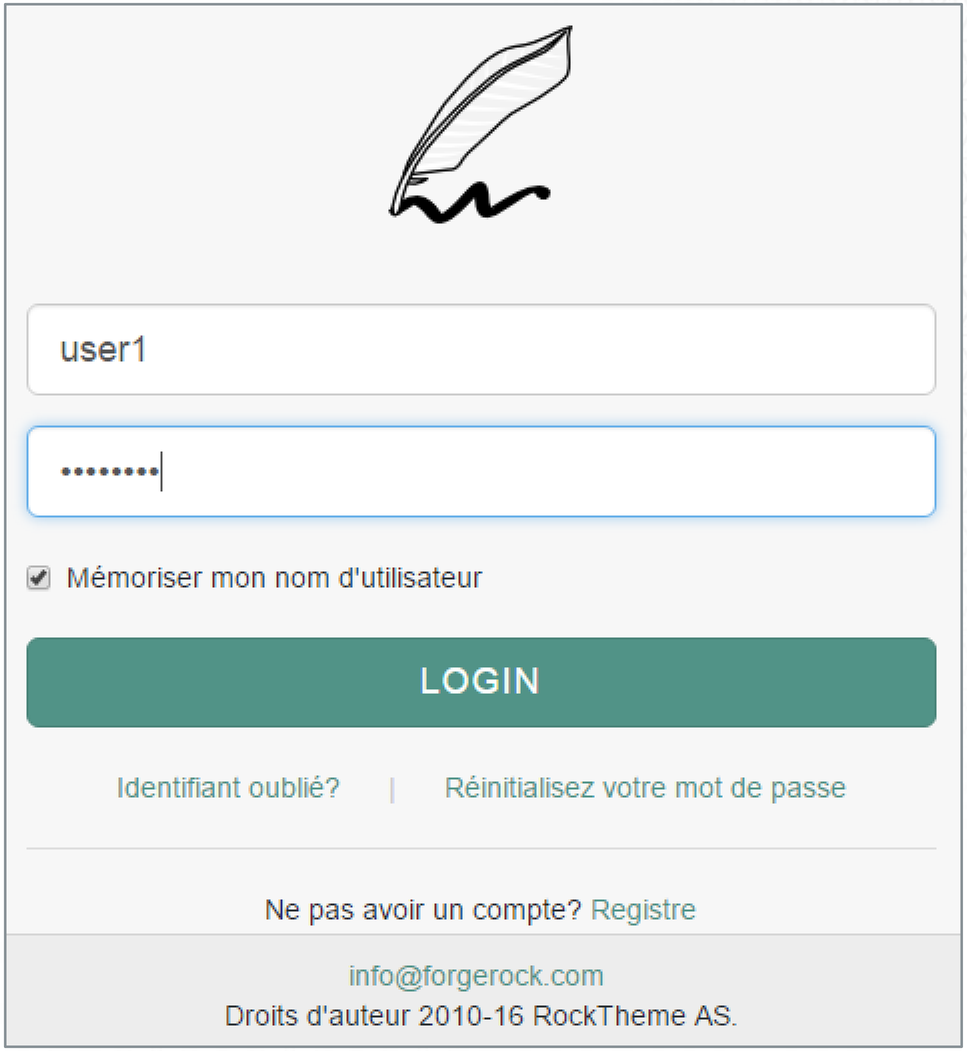
```
"copyright" : "Droits d'auteur 2010-
16 RockTheme AS.",
```

5. Edit your project's ui-configuration.json file, change the lang property to fr

```
"lang": "fr",
```

View the Changes in the Self Service UI

- Changes are visible as soon as they are made
- You may need to refresh your browser window



user1

•••••••••

☑ Mémoriser mon nom d'utilisateur

Ne pas avoir un compte? Registre

info@forgerock.com

Droits d'auteur 2010-16 RockTheme AS.

Customize a Template

- Templates can be customized to meet a particular business need
- This example adds an Employee ID field to the user registration screen:
  - Add a new form-group stanza to userDetails-initial.html
  - Edit translation.json to add the employee ID number field

Add a Form-Group Stanza

1. Open userDetails-initial.html

```
$ cd selfservice/extension/templates/
user/process/registration
$ gedit userDetails-initial.html
```

2. Add the Employee ID field:

```
<div class="form-group">
  <label class="sr-only" for="input-employeeNum">{{t'common.user.employeeNum'}}
  </label>
  <input type="text" placeholder="{{t'common.user.employeeNum'}}" id="input-
  employeeNum" name="user.employeeNum" class="form-control input-lg"/>
</div>
```

Add the Form to translation.json

1. Open translation.json
2. Locate the place in the file to add the form information (right after Last Name)
3. Enter the form information
4. Save and close the file

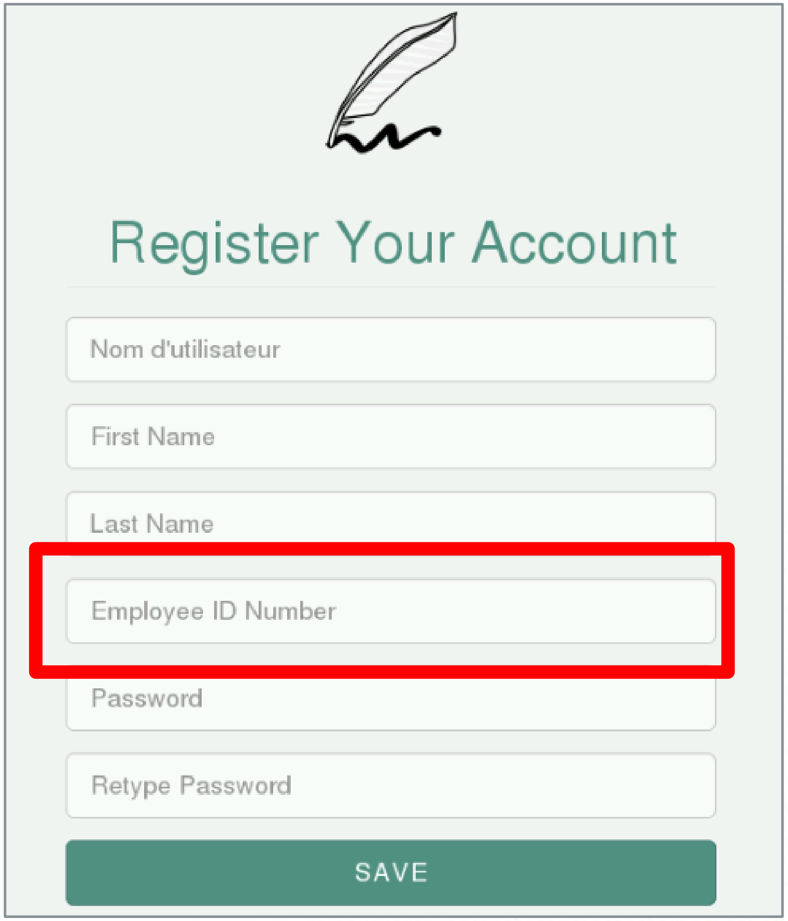
```
$ cd selfservice/extension/locales/en
$ gedit translation.json
```

```
...
"givenName" : "First Name",
"familyName" : "Family Name",
"sn" : "Last Name",
"basicInfo" : "Basic Info",
...
```

```
...
"givenName" : "First Name",
"familyName" : "Family Name",
"sn" : "Last Name",
"employeeNum" : "Employee ID Number",
"basicInfo" : "Basic Info",
...
```

Modified Template in Self Service

- Once your changes are made you see Employee ID added to the self-registration form:



Register Your Account

Nom d'utilisateur

First Name

Last Name

Employee ID Number

Password

Retype Password

Lab: Customize the User Interface

- Perform the following exercises in the Student Workbook:
  1. Prepare your system for customization
  2. Customize the UI theme
  3. Customize Self Service
- Approximate time: 60 minutes

Module 12: Customize the end-user interface

End of module

Chapter 3: Self-service

End of chapter