If you want to allow visitors to register and sign in to your website, you can choose from a number of authentication methods. Kentico supports forms authentication, Windows AD authentication and third party authentication. You can use only one of these options or a mix of them and let the users choose the one that is the most convenient for them.

Internal authentication

Configuring forms authentication

The **forms authentication** stores user names and passwords in the database and requires users to register to your site before they can log in. This is the default option.

You can allow users to register and sign in to your site by using the Registration form web part. See Using the Registration form and Custom registration form web parts.

Configuring multi-factor authentication

Multi-factor authentication uses a combination of forms authentication and one other security factor (for example a passcode generated by a mobile application or sent by SMS, email, etc.). Utilize the default passcode solution or implement your own custom solution.

External authentication

Configuring Windows Active Directory authentication

The Windows AD authentication gets user identity from the network credentials and automatically creates a corresponding user in the database, including the user’s roles (if they exist in the Kentico database). Users are not required to enter their user names and passwords when logging in to Kentico.

Configuring mixed-mode authentication

You can also use the forms and Windows AD authentication at the same time in a mixed mode. The users will be able to choose which authentication method they will use when registering/logging in to your website.

Claims-based authentication

Hand over the authentication process to an external service called identity provider and provide your users with the comfort of single sign-on mechanism.

Third-party authentication services

Third-party authentication services provide an alternative way for users to log in and register to your site. This way, they do not need to go through the registration process and create a new user name and password for your site. Instead, they can use the same user name and password that they already use on a popular site or service (such as Microsoft accounts, Facebook, LinkedIn, Yahoo!, etc.). Even new users can log in to your site like this, in which case the system creates a new user account in the database for them. To learn how to manage the data of imported users, see Managing users coming through a third-party authentication service.

Kentico supports the following authentication providers:

- Windows Live ID
- OpenID
- Facebook Connect
- LinkedIn

Configuring custom external authentication

If you want to retrieve user and role information from an external source (such as a custom database), you need to configure the system.

Kentico allows you to write a custom authentication provider. In this way, the submitted user name and password are checked against an external user profile source/authentication source. If the user is successfully authenticated, the user account is automatically created/updated in the Kentico database, without copying the user password.

You can integrate your custom authentication provider with Kentico by handling the system's security events.

Other related tasks

Sharing user accounts between sites - learn how to share user accounts among all sites running on one Kentico installation.

Configuring single sign-on - learn how to enable users to authenticate once and gain access to multiple websites, which are either running on a single domain or on different domains.

Accessing current user data in code

When the user is authenticated, a *UserInfo* object representing the current user is stored in the session variable **CMSCurrentUser** and is accessible through the **CMS.Membership.MembershipContext.AuthenticatedUser** property. All operations after authentication then use the user profile and user roles assigned to this object.

```
// Gets the user name of the current user
string userName = CMS.Membership.MembershipContext.AuthenticatedUser.UserName;
```