Kentico offers two services which provide communication and synchronization of content and objects between servers. Both services are disabled by default. Enable the services only if you plan to use them.

- You can enable Staging in **Settings -> Versioning & Synchronization -> Staging -> Enable staging service**.
- You can enable REST in **Settings -> Integration -> REST -> Service enabled**.

Staging

The weakest spot of the Staging service is in the authentication process. If potential attackers obtained the user name and password for the service, they could stage the administrator and gain absolute power over the system.

You can secure the staging service using two authentication options:

- **User name and password**
- **X.509 certificate**

The recommended option is to use the X.509 certificates for authentication, as certificates generally provide better security. See Using X.509 authentication for more details.

REST

The REST service provides access to the objects in Kentico, so a potential attacker could obtain or modify any data in the system.

You can secure the REST service by using **SSL** along with Basic authentication. See Authenticating REST requests and Configuring SSL for more details.

You can also use the **Hash parameter authentication** for authenticating individual REST requests. You only need to generate the hash in the administration interface and add the hash to URL. This URL then serves a particular REST request without the need of further authentication.

The REST service should optimally check the correct authentication with every request. However, because of other services in Kentico (e.g., chat), which need some HTTP context within WCF, the checks are not performed every time. You can change this behavior by changing the *aspNetCompatibilityEnabled* key to **false** in the **<system.serviceModel>** section of the web.config file:

```
<serviceHostingEnvironment aspNetCompatibilityEnabled="false" />
```

Note that setting this key also **disables the chat** functionality.

Best practice

The best practice with REST is to **assign a dedicated user** to the service, grant the user **permissions only for the desired objects**, configure access through **SSL** and disable the **aspNetCompatibilityEnabled** mode.