

Modules - sms: Configurations

These configuration items are available within the sms module:

```
# Enable property for SMS GET script driver
idm.sec.sms.script.get.enabled
#
# Login that will be used for composes URL with gateway
idm.sec.sms.script.get.login
#
# Password that will be used for composes URL with gateway
idm.sec.sms.script.get.password
#
# Script that be used for composes URL
idm.sec.sms.script.get.scriptCode
#
# Timeout for communication with gateway
idm.sec.sms.script.get.timeout
#
# Phone number that will be used for overriding all recipient number (can be
used for testing)
idm.sec.sms.script.get.overrideNumber
```

It is also necessary to define with sender will be used. These are now available:

- **defaultScriptSmsSender** - simple sms gateway with check http result codes,
- **defaultHtmlSmsSender** - check http page with response (OK/ERROR must contains the page),
- **defaultSmsBranaSmsSender** - implementation of [SmsBrana](#).
- **defaultEmailSmsSender** - gateway which sends SMS by sending emails to addresses of the form "usersphone@gateemail.tld". E.g. <https://www.sms-operator.cz/>

If you want to set up one of these senders, create new configuration property:

```
# sender implementation
idm.sec.sms.notification-sender.sms.impl=<sender>
```

More information about this configuration can be found in [configuration](#).

For defaultEmailSmsSender, it's required to configure the email suffix, where the emails are sent, e.g.:

```
# Suffix for sending SMS by emails. The final recipient's address is
composed as "usersphone" + suffix
idm.sec.sms.script.get.sms.email.sender.suffix=@test.cz
```

Modules - sms: Script for compose URL

For a composition of URL for a given gateway, create a script from the System category. As parameters add these variables:

- **login** - login for connect to sms gateway,
- **password** - password as GuardedString (don't forget call method asString) for SMS gateway,
- **number** - number with real recipient,
- **message** - message as plain text, don't log this message, it is possible that this message will contain password as plaintext



Attributes like message, number, etc. may contain some characters that aren't passing GET method. So try to encode attributes with `URLEncoder.encode`

Example script

For this script is necessary add these authorities:

- CLASS: java.net.URLEncoder
- CLASS: java.nio.charset.StandardCharsets
- CLASS: sun.nio.cs.UTF_8

```
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;

def messageEncoded = URLEncoder.encode(message,
StandardCharsets.UTF_8.name());
def numberEncoded = URLEncoder.encode(number,
StandardCharsets.UTF_8.name());

return "https://something.tld/example_gateway.php?login=" + login +
"&password=" + password.asString() + "&message=" + messageEncoded +
"&number=" + numberEncoded + "&someTestAttribute=test123";
```

From:
<https://wiki.czechidm.com/> - **CzechIdM Identity Manager**

Permanent link:
https://wiki.czechidm.com/tutorial/adm/modules_sms

Last update: **2023/08/18 13:32**

