

User properties - CustomerID

user.username.minimum.length

This property defines the minimum length for a username.

Default is 4.

Example:

```
user.username.minimum.length = 4
```

user.username.pattern.N

These properties define username check patterns. A username must match all the defined patterns. The property name begins with user.username.pattern. and ends with the pattern index number. The pattern format follows the Java regular expression syntax. You can learn more about it at the [java.util.regex.Pattern API documentation](#). The actual matching is done against the whole username string and "." is appended to the pattern as both prefix and postfix.

Default is [a-zA-Z]

Example:

```
user.username.pattern.1 = [a-zA-Z] user.username.pattern.2 = @
```

In the example above, the username must contain some character from the range a-z or A-Z and also @-character. So, a username of "admin@" would be valid, but "admin" not, because it doesn't contain @-character.

user.username.characters

This property defines those characters that can be used in the username. All the characters in the username must match the given regular expression. The difference to the user.username.pattern.N property is that the matching is done character by character, not to the whole username string. The pattern format follows the Java regular expression syntax. You can learn more about it at the [java.util.regex.Pattern API documentation](#).

Default is [a-zA-Z0-9@]|[_-]

Example:

```
user.username.characters = [a-zA-Z]
```

In the example above, every character in the username must be in the range a-z or A-Z. So, a username of "admin" would be ok, but a username of "4dm1n" would not, because it contains numbers.