

Validation configuration - CustomerID

Validation can be configured in a configuration file called `validation.json`. The format of the JSON file has the following structure:



Figure 1: Structure of validation.json

The configuration consists of a JSON Object, with attributes starting with a `validation.` prefix and ending with the fully qualified field name, like `user.ssn`, resulting in a configuration property like `validation.user.ssn`.

This configuration property's value in turn is a JSON Array, which makes it possible to configure multiple validators to the same field. For example, one could be interested in configuring constraints for amount of characters in input data, making it required and enforcing uniqueness within the database and in this case one would make use of three validators to satisfy all these criteria.

Within the JSON Array a number of validator components are configured in JSON Object format. The system requires that each validator configuration has at least a class attribute defined. This attribute value is used to select the validation logic. Additional attributes can be used to configure each validator, which can be of *numeric*, *textual* or *list* type. Internally these are passed to validator's setter methods as instances of `java.lang.Long`, `java.lang.String` and `java.util.List` respectively.

For example, if a registration system were implemented where users could register with attributes like first name, last name, email, login and a unique alias for discussion boards and the following rules would apply to each of these.

- 1) First name and last name must begin with a capital letter and be at most 20 characters in length, containing only ASCII alphabetic letters.
- 2) Email must be valid,
- 3) login must be unique
- 4) alias must be unique, contain 5-8 lowercase ASCII alphabetic letters, numbers, underscore or hyphen

In this case, the following configuration would apply.

Example:

```

{
  "validation.user.firstname" : [
    { "class" : "com.ubisecure.customerid.logic.validators.RegexpValidator",
      "pattern" : "^[A-Z]{1}[a-z]{1,19}",
      "errorKey" : "general.ui.errors.invalid.firstname"
    }
  ],
  "validation.user.surname" : [
    { "class" : "com.ubisecure.customerid.logic.validators.RegexpValidator",
      "pattern" : "^[A-Z]{1}[a-z]{1,19}",
      "errorKey" : "general.ui.errors.invalid.surname"
    }
  ],
  "validation.user.alias" : [
    { "class" : "com.ubisecure.customerid.logic.validators.RegexpValidator",
      "pattern" : "^[a-z0-9_-]{5,8}",
      "errorKey" : "general.ui.errors.invalid.alias" },
    { "class" : "com.ubisecure.customerid.logic.validators.UniqueAttributeValidator",
      "errorKey" : "general.ui.errors.existing.alias",
      "attrName" : "user.alias" },
    { "class" : "com.ubisecure.customerid.logic.validators.RequiredValidator",
      "errorKey" : "general.ui.errors.required.alias" }
  ]
}

```

In the example configuration above, the email and login attributes were omitted because there is already sufficient built-in logic. Attributes with built-in validation are listed in following table.

Attribute name	Built-in validation
password	Validation is performed based on data entered in first name, surname and login fields in addition to the data entered in the password field. There must not exist similar occurrences of three consecutive letters in the other mentioned fields compared to the password field.
mobile	Mobile numbers must begin with a '+' character, contain only numbers or hyphens and be at most 20 characters in length. With specific configuration verification SMS are sent to mobile devices which must be entered during registration before it's possible to proceed.
email	Email addresses must be properly formatted and unique. With certain configuration email addresses may also be verified by sending email to the given address.

Table 1 - Built-in validation

Currently following classes of validators can be configured in `validation.json`.

com.ubisecure.customerid.logic.validators.CombinedEmailValidator

Performs a unique attribute and email format validation to given attribute.

Configuration parameters	Description	Mandatory	Multivalue
class	com.ubisecure.customerid.logic.validators.CombinedEmailValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.FixedSSNValidator

Validates if the input value is a valid Finnish personal identification number.

Configuration parameters	Description	Mandatory	Multivalue
class	com.ubisecure.customerid.logic.validators.FixedSSNValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.FixedCompanyIdValidator

Validates if the input value is a valid Finnish company identifier.

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.FixedCompanyIdValidator	Yes	No

com.ubisecure.customerid.logic.validators.RealnameValidator

Validates if input can be considered a human first name or last name.

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.RealnameValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.RegexpValidator

Validates if input matches defined regular expression.

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.RegexpValidator	Yes	No
pattern	Regular expression to validate against	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.RequiredValidator

Validates if the input has been entered.

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.RequiredValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.UniqueAttributeValidator

Fails validation if a user exists in database with attribute defined by the configuration parameter attrName. However if there is a match whose database id matches the user who is registering, the validation will succeed.

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.UniqueAttributeValidator	Yes	No
attrName	The attribute name that uniquely identifies a user	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.UniqueEmailValidator

Fails validation if a user exists in database with the entered email address, unless the registering user matches the user in the registration (which implies that the user must be logged in).

Configuration parameters	Description	Mandatory	Multivalued
class	com.ubisecure.customerid.logic.validators.UniqueEmailValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.UniqueLoginValidator

Fails validation if a user exists in database with the entered login, unless the registering user matches the user in the registration (which implies that the user must be logged in).

Configuration parameters	Description	Mandatory	Multivalue
class	com.ubisecure.customerid.logic.validators.UniqueLoginValidator	Yes	No
errorKey	Defines a language resource key representing error message that should be shown to user.	No	No

com.ubisecure.customerid.logic.validators.UsernameValidator

Reads its configuration from `eidm2.properties` and validates if the input conforms to that configuration.

Configuration parameters	Description	Mandatory	Multivalue
class	com.ubisecure.customerid.logic.validators.UsernameValidator	Yes	No