

Column Validation

Servoy provides an opportunity to implement validation rules at the column level. There are several built-in validation rules, which may be implemented at design-time. Additionally, custom validation rules may be written in as a JavaScript method which is bound to a column. Servoy also allows the contribution of a column validator by a java plugin.

In Servoy 2020.9 by default the validation will only be called when `databaseManager.validate(record)` is call or when `saveData()` is called, which does call `validate(record)` itself also. This behavior can be set back by a property `"servoy.execute.column.validators.only.on.validate_and_save"` (which is configurable in the ui in the Column validation part of the table editor). If set back or before 2020.9 the behavior is that a validation event occurs at the moment a record's value for a column changes. This may be the result of a user's action or some code which is executed. When validation fails, a [Servoy Exception](#) is raised for [Invalid Input](#), which may be trapped in a solution's `onError` event handler. With 2020.9 it is better to use the new validation system see: [Data/Record/Column validation](#)

Numeric Range Validation

Servoy provides built-in numeric validation for Integer and Number data types. Providing upper and lower bounds will automatically enforce that any value entered is between (inclusive) the range provided. Providing only a lower bound will enforce that any value entered is greater-than-or-equal-to the bound. Providing only an upper bound will enforce that any value entered is less-than-or-equal-to the bound.

Size/Length Validation

Servoy provides built-in validation for the size/length of a value in a column. This rule is applicable to Text and Media data types. Setting the length property for Text columns will enforce that value entered has a length of characters which is less-than-or-equal-to the length specified in the rule. Setting the size property for Media columns will enforce that value entered has a size, measured in number of bytes, which is less-than-or-equal-to the size specified in the rule.

RegEx Validation

Servoy offers the flexible pattern matching capability of [Regular Expressions](#) as a means to apply validation rules to Text columns. Providing a RegEx value will enforce that any value entered into the Text column must match on the expression. RegEx is an excellent way to match on patterns, such as phone numbers, email addresses, and much more. [RegExLib](#) is a useful site containing user-generated libraries of expressions to suit many needs.

Email Validation

Servoy provides a built-in email validation rule, which enforces that any Text column matches a pattern which is similar to email addresses. This pattern is ideal for most use cases. However, developers may implement their own RegEx validation to ensure an exact match on the pattern of their choice.

Identifier Validation

A build in validator to make sure that the values are javascript based identifiers (so can be used in scripting or for solution model stuff)

Custom Validation

Apart from the built-in validation rules, Servoy allows developers to author business logic to enforce their own validation rule for a column. A Global Method may be bound to a column, such that when a validation event occurs for the column, the method is invoked. The value that is entered is passed into the method and a developer may then execute any evaluation of the value before returning a boolean value; true indicates that validation is successful.

```
/**
 * Custom Validation rule: Must be Dog or Cat (case insensitive)
 * @param {Object} obj The value that will be validated
 * @returns {Boolean} True when successful
 * @properties={typeid:24,uuid:"655B9F0E-A1A2-4B0B-84CD-8E299546DB57"}
 */
function validateColumn(obj) {
    return 'Dog'.equalsIgnoreCase(obj) || 'Cat'.equalsIgnoreCase(obj);
}
```

Column Validator from Java Code (plugin)

A Column validator can be contributed by a java plugin. See [Providing converters and validators from plugins](#) for more information.