5.10.3.4 Ranges

If the variable supports a range of variable values, you can set the variable value as follows:

Example:

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {from: "5", to: "7"});
```

If the variable supports excluding a range of variable values, you can set the variable value as follows:

Example:

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {exclude: true, from: "5", to: "7"});
```

5.10.4 Getting Variable Values

You can get the values of a variable with the method `getVariableValues()`. It returns an array of all variable values as `VariableValue` objects. The script API definition is as follows:

```
getVariableValues(variable: string | VariableInfo): VariableValue[]
```

Example:

In the following example, the variable values of the variable `V_Country` are retrieved:

```
var values = Table_1.getDataSource().getVariableValues("V_Country");
```

Each value in the array is an instance of either `SingleVariableValue`, `MultipleVariableValue`, or `RangeVariableValue`, which all inherit from `VariableValue`. To work with such an instance, that is, to access its type-specific properties, cast the instance to the corresponding type first, using the `type` property. The following example shows how:

```
var value = Table_1.getDataSource().getVariableValues("V_Country")[0];
switch (value.type) {
  case VariableValueType.Single:
    var singleValue = cast(Type.SingleVariableValue, value);
    console.log(singleValue.value); // you can access the 'value' property now
    break;
  case VariableValueType.Multiple:
    var multiValue = cast(Type.MultipleVariableValue, value);
    console.log(multiValue.values); // you can access the 'values' property now
    break;
  case VariableValueType.Range:
    var rangeValue = cast(Type.RangeVariableValue, value);
    console.log(rangeValue.from); // you can access the 'from' property now
    console.log(rangeValue.to);   // you can access the 'to' property now
    // further range properties: 'less', 'lessOrEqual', 'greater', 'greaterOrEqual'
    break;
  default:
    break;
}
```

Note: What you set with `setVariableValue()` isn't necessarily the same that's returned from `getVariableValues()`.