



Figure 50: Variable Values Are Applied to the Model of the Application or the Widget

Note: This method isn't validating the specified variable values neither at runtime nor at design time. All values and value combinations which are accepted in the Prompt dialog will be supported. All other combinations might lead to errors or inconsistent state.

### 5.10.3.1 Single Variable Values

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

*Example:*

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {value: "5"});
```

or, alternatively,

```
Table_1.getDataSource().setVariableValue("VAR_NAME", "5");
```

If the variable supports excluding a single variable value, you can set the variable value as follows:

*Example:*

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

### 5.10.3.2 Multiple Variable Values

If the variable supports multiple values, you can set the variable values as follows:

*Example:*

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {values: ["5", "7"]});
```

If the variable supports excluding multiple values, you can set the variable value as follows:

*Example:*

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {exclude: true, values: ["5", "7"]});
```

### 5.10.3.3 Comparisons

If the variable supports comparison operations <, <=, >, and >= you can set the variable value as follows:

*Example:*

```
Table_1.getDataSource().setVariableValue("VAR_NAME", {less: "5"});
Table_1.getDataSource().setVariableValue("VAR_NAME", {lessOrEqual: "5"});
Table_1.getDataSource().setVariableValue("VAR_NAME", {greater: "5"});
Table_1.getDataSource().setVariableValue("VAR_NAME", {greaterOrEqual: "5"});
```