Dependency matrix tool API

On this page

Dependency Matrix tool allows templates to access data from a Dependency Matrix, use diagrams to get data, row elements, column names, or relations between row and column elements from the matrix.

You can use the following Dependency Matrix tool API in the template.

Class DependencyMatrixTool

Use the following methods to use diagrams to get data.

- +getMatrix(diagram : Diagram) : Matrix
- +getMatrix(diagramName : String) : Matrix

Class Matrix

Use the following methods to get matrix row elements or nodes, column elements or nodes, and dependencies between these nodes/elements.

- +getRowNodes() : List<ElementNode>
- +getColumnNodes() : List<ElementNode>
- +getCellDependencies(rowNode: ElementNode, columnNode: ElementNode)
- +getRowElements() : List<Element>
- +getColumnElements() : List<Element>
- +getRelation(row: Element, column : Element) : List<Relation>

Class Relation

Describes dependency in a Dependency Matrix cell. Use the following methods for class relations.

- +getName() : String
- +getElement() : Element
- +getSemanticType() : String
- +getDirection() : String

Using Diagrams to Get Data from Dependency Matrix

A Dependency Matrix is a special diagram. You can retrieve it using the $Diagram variable. Thus, every method must accept a diagram instance or a diagram's name. Use the following methods to return a Dependency Matrix instance. You can retrieve rows and columns from the Dependency Matrix instances.

Getting Dependency Matrix Instances from Diagram Elements

getMatrix(diagram : Diagram) : Matrix
To get a Dependency Matrix instance from a specified diagram element, use the following code.

```
#set($matrix = $depmatrix.getMatrix($diagram))
```

Where the parameter is:

- diagram – a diagram element

Return an instance of Dependency Matrix.

For example:

```
#foreach($diagram in $project.getDiagrams("Dependency Matrix"))
#set($matrix = $depmatrix.getMatrix($diagram))
#end
```

Getting Dependency Matrix Instances from Diagram Names

getMatrix(diagramName : String) : Matrix
To get a Dependency Matrix instance from a specified diagram's name.

```
#set($matrix = $depmatrix.getMatrix($diagram))
```

Where the parameter is:

- diagram – a diagram's name

Return a Dependency Matrix instance.

Example code:

```
#set($diagram = "diagram name")
#set($matrix = $depmatrix.getMatrix($diagram))
```

Getting Rows and Columns

The Matrix consists of rows and columns. Each row or column is an instance of ElementNode object. Each ElementNode represents a single model Element. There might be multiple ElementNodes for the same model Element in a Dependency Matrix diagram, since the same model Element can be represented multiple times in a diagram.

Getting All Rows

getRowNodes() : List<ElementNode>

Use this method to retrieve a list of rows, ordered in the same way as actually displayed in the diagram. Nodes representing grouping packages, that do not belong to the actual matrix data, are excluded.

```
$matrix.getRowNodes()
```

The returned value is a list of ElementNodes. Method **getElement() : Element** can then be used to retrieve the model element represented by a row.

To print all row elements' names, for example, type the following code:

```
#foreach($rowNode in $matrix.rowNodes)
$rowNode.element.name
#end
```

Getting all Columns

getColumnNodes() : List<ElementNode>

Use this method to retrieve a list of columns, ordered in the same way as actually displayed in the diagram. Nodes representing grouping packages, that do not belong to the actual matrix data, are excluded.

```
$matrix.getRowNodes()
```

The returned value is a list of ElementNodes. Method **getElement() : Element** can then be used to retrieve the model element represented by a column.

To print all column elements' names, for example, type the following code:

```
#foreach($col in $matrix.columns)
$colNode.element.name
#end
```

Getting Dependencies between Rows and Columns

getCellDependencies(rowNode: ElementNode, columnNode: ElementNode)

Use this method to retrieve the relations in a specific matrix cell, which is identified by provided row and column nodes.

```
$matrix.getCellDependencies($rowNode, $columnNode)
```

Where the parameter is:

- rowNode – ElementNode of a row
- columnNode – ElementNode of a column

The returned value is a list of Relations.

The Relation class contains the following methods:

- **getSemanticType** : String
  Return the semantic type
- **getElement** : Element
  Return a relationship element or null if the relationship is not an element, for example, tag name.
- **getName** : String
  Return a relationship name. If the relationship is a tag name, it will return the tag name. If the relationship is NamedElement, it will return the element name; otherwise, return a human name.
- **getDirection** : String
  Return the direction.

To print the row, column, and its relationship name, for example, type the following code:

```
#foreach($row in $matrix.rowNodes)
   #foreach($col in $matrix.columnNodes)
      #foreach($rel in $matrix.getCellDependencies($row, $col))
         $row.element.name has $rel.name with $col.element.name
      #end
   #end
#end
```

To print rows, columns, and relations in a spreadsheet file format, for example, type the following code:

`#import('depmatrix','com.nomagic.reportwizard.tools.DependencyMatrixTool')`
 	
`#forpage ($diagram in $report.filterDiagram ($Diagram,["Dependency Matrix"]))`
 	
`#set($matrix=$depmatrix.getMatrix($diagram))`
 	
`$bookmark.create($diagram.name)`		

	`#forcol ($col in $matrix.columnNodes)$col.element.name#endcol`	

`#forrow($row in $matrix.rowNodes)$row.element.name`	`#forcol ($col in`
	
	`$matrix.columns)#if(!$report.isEmpty($matrix.getCellDependencies($row,$col)))`
	
	`->#end#endcol`
		`#end row`

`#endpage`		