

Analytics designer provides a Table script API and a Data Source script API to help analytic application developers add script custom specific logic into their analytic applications.

Besides the common widget script API like `getVisible()` and `setVisible()`, the main Table script API is listed below:

### **addDimensionToColumns**

```
addDimensionToColumns(dimension: string|Dimension, position?: integer): void
```

Adds the dimension to the column axis at the specified position. If no position is specified, the dimension is added as the last dimension of the column axis.

*Example:*

```
Table_1.addDimensionToColumns("Location_4nm2e04531");
```

### **addDimensionToRows**

```
addDimensionToRows(dimension: string|Dimension, position?: integer): void
```

Adds the dimension to the row axis at the specified position. If no position is specified, the dimension is added as the last dimension of the row axis.

*Example:*

```
Table_1.addDimensionToRows("Location_4nm2e04531");
```

### **getDataSource**

```
getDataSource(): DataSource
```

Returns the data source of the table. If the table has no data source, `undefined` is returned. Refer to the section on the data-related script API.

### **getDimensionsOnColumns**

```
getDimensionsOnColumns(): Dimension[]
```

Returns the dimensions on the column axis.

### **getDimensionsOnRows**

```
getDimensionsOnRows(): Dimension[]
```

Returns the dimensions on the row axis.

### **getPlanning**

```
getPlanning(): Planning
```

Returns the planning object of the table. If the table data source isn't of type planning, `undefined` is returned.

Refer to the section on Planning.

### **getSelections**

```
getSelections(): Selection[]
```

Returns the selections of the chart. You can use the elements of the returned array with the function `DataSource.getData()` to get the value of a cell. For more information about the `Selection`, see <https://help.sap.com/doc/958d4c11261f42e992e8d01a4c0dde25/2019.8/en-US/doc/Selection.html>.