

Description:

Why AutoComplete filter by value field and does not filter using text field.

Solution:

In EJ2 AutoComplete, the search and filter works based on the value field alone. Mapping text field for AutoComplete will only update the text to list items in popup but functionalities like sorting and filtering will be performed based on the value field. So, you can bind only the value field based on the component design. To filter items in AutoComplete based on both text and value fields, use **predicate** of **dataManager** using the filtering event. The filtered data can be again updated using the **updateData** method.

Refer to the following code for filtering using both fields.

```
filtering: function(e)
{
    e.preventDefaultAction=true;
    var predicate = new Predicate('Name', 'contains', e.text);
    predicate = predicate.or('Code', 'contains', e.text);
    var query = new Query();
    //frame the query based on search string with filter type.
    query = (e.text != "") ? query.where(predicate) : query;
    //pass the filter data source, filter query to updateData method.
    e.updateData(this.dataSource, query);
}
```

As an alternative, the ComboBox component can also be used instead of AutoComplete to bind both fields and set the allowFiltering property of ComboBox to true.

Also if you want to trigger the query based on the text field, pass the text field in **updateData** method as shown below,

```
e.updateData(this.dataSource, query, {text: "Name"});
```