

```

1 // Please refer to this article for script examples:
  https://www.crowcanyon.help/article/709/
2 // viewName parameter has the name of the current list view.
3 // objListView parameter has reference to NITRO list view object.
4 var scriptManager = {
5 };
6 // Use this function to fetch data for any columns in addition to columns in the
  configured list view.
7 // These column values can be used in other funtions like
  UpdateListItemsValues/UpdateListItemValues.
8 // viewColumns parameter has the columns names from the configured list view.
9 scriptManager.GetColumnsToFetch = function(viewColumns, viewName, objListView){
10     var loadColumnNames = ["title", "Status", "AssignedTo"];
11     return loadColumnNames;
12 };
13 // This function is called after fetching data for the view configured in the list view.
14 // It can be used to modify list column values before this data is processed to render
  the list view.
15 // Use function GetColumnsToFetch to specify additional columns to fetch data for use in
  this function.
16 // listItems parameter has data for all items in the specified view.
17 scriptManager.UpdateListItemsValues = function(listItems, viewName, objListView){
18     //console.log("listItems:", listItems);
19     return listItems;
20 };
21 // This function is called after fetching items for the view configured in the list view.
22 // Script code can be used to modify list column values before this data is processed to
  render the list view.
23 // Use function GetColumnsToFetch to specify additional columns to fetch data for use in
  this function.
24 // This function is called for each item in the view.
25 scriptManager.UpdateListItemValues = function(listItem, viewName, objListView){
26     //console.log("listItem: ", listItem);
27     return listItem;
28 };
29 // This function is called to decide whether to show the item in the list view or not.
30 scriptManager.IsItemVisible = function(listItem, viewName, objListView){
31     console.log("IsItemVisible listItem:", listItem);
32     // debugger;
33     const itemStatus = listItem.get_item("Status");
34     const itemAssignedTo = listItem.get_item("AssignedTo");
35     console.log("itemStatus:", itemStatus);
36     console.log("itemAssignedTo", itemAssignedTo);
37     if (itemAssignedTo && itemAssignedTo === "Brian"){
38         return true;
39     }
40     else {
41         return false;
42     }
43 };
44 // This function is called to post list view loaded.
45 scriptManager.PostRenderListView = function(objListView){
46     return true;
47 };
48 return scriptManager;
49

```