

```

// Please refer to this article for script examples: https://www.crowcanyon.help/article/709/

// viewName parameter has the name of the current list view.

// objListView parameter has reference to NITRO list view object.

var scriptManager = {

};

// Use this function to fetch data for any columns in addition to columns in the configured list
view.

// These column values can be used in other funtions like
UpdateListItemsValues/UpdateListItemValues.

// viewColumns parameter has the columns names from the configured list view.

scriptManager.GetColumnsToFetch = function(viewColumns, viewName, objListView){

    var loadColumnNames = [];

    return loadColumnNames;

};

// This function is called after fetching data for the view configured in the list view.

// It can be used to modify list column values before this data is processed to render the list
view.

// Use function GetColumnsToFetch to specify additional columns to fetch data for use in this
function.

// listItems parameter has data for all items in the specified view.

scriptManager.UpdateListItemsValues = function(listItems, viewName, objListView) {

    var internalName = "LkTest3_x003a_RequesterEmail";

    var itemsTotalCount = listItems.get_count();

    for (var i = itemsTotalCount - 1; i >= 0; i--) {

        var currentItem = listItems.getItemAtIndex(i);

        var itemProperties = currentItem.get_fieldValues();

        var lookupField = itemProperties[internalName];

        //

        // Check if the lookup field exists and has a display value

        if (lookupField && lookupField.$Dr_1) {

            if (lookupField.$Dr_1 === "SB Region") {

                lookupField.$Dr_1 = "Southern Region";

            }

        }

    }

}

```

```
// Replace display value
    console.log("Updated display value to 'Southern Region' for item at index", i);
}
}
else {
    console.log("Lookup field is missing or invalid for item index", i);
}
}
return listItems;
};

// This function is called to decide whether to show the item in the list view or not.
scriptManager.IsItemVisible = function(listItem, viewName, objListView){
    return true;
};

// This function is called to post list view loaded.
scriptManager.PostRenderListView = function(objListView){
    return true;
};

return scriptManager;
```