import \* as jsPDF from 'jspdf';

import \* as autoTable from 'jspdf-autotable'

/\*export to excel \*/

uri = 'data:application/vnd.ms-excel;base64,';

template = '<html xmlns:o="urn:schemas-microsoft-com:office:office" xmlns:x="urn:schemas-microsoft-com:office:excel" xmlns="http://www.w3.org/TR/REC-html40"><head><!--[if gte mso 9]><xml><x:ExcelWorkbook><x:ExcelWorksheets><x:ExcelWorksheet><x:Name>{worksheet}</x:Name><x:WorksheetOptions><x:DisplayGridlines/></x:WorksheetOptions></x:ExcelWorksheet></x:ExcelWorksheets></x:ExcelWorkbook></xml><![endif]--></head><body><table>{table}</table></body></html>'

base64(s: any) {

return window.btoa(s);

}

format(s, c) {

return s.replace(/{(\w+)}/g, function (m, p) {

return c[p];

})

}

tableToExcel(temp, worksheetName) {

var table = temp,

ctx = { worksheet: worksheetName, table: table },

href = this.uri + this.base64(this.format(this.template, ctx));

return href;

};

public exportexcel(selectedformat: any) {

console.log(selectedformat, "selectedformate");

this.myText = 'RDmi Infrastructure' + "-" + this.DeploymentName + "-"+this.TenantIDClicked;

var template = '<table>', thead = '', tbody = '', caption = '<h4>'+this.myText+'</h4>';

// removal of headerName if having the corresponding values are empty.

var array: any = this.api.getDataAsCsv().split('\n')[0].split(',');

var aa = this.api.getDataAsCsv().split('\n').slice(1);

var header = array.map(function (name, index) {

var count = 0;

aa.map(function (str, ia) {

str.split(',').map(function (value, i) {

if (value.length == 2 && (index == i)) {

count += 1;

}

});

});

if (aa.length == count) {

name = false;

}

return name;

});

var indexes = header.map(function (value, index) {

if (!value) { return index; }

}).filter(function (value) {

if (value !== undefined) { return value; }

});

var body = aa.map(function (str) {

return str.split(',').map(function (value: any, index) {

if (indexes.indexOf(index) !== -1) {

value = false;

}

return value;

})

}).map(function (arr) {

return arr.filter(function (name) { return name; })

});

header = header.filter(function (value) {

return value;

});

var finalRes = [];

finalRes.push(header);

body.map(function (arr) {

finalRes.push(arr);

});

finalRes.map(function (row, index) {

var tBodyRow = '';

row.map(function (name, i) {

name = name.substr(1, name.length - 2);

name = name.replace('"', '');

var str;

if (index == 0) {

str = '<th>' + name + '</th>';

thead = thead.concat(str);

} else {

str = '<td>' + name + '</td>';

tBodyRow = tBodyRow.concat(str);

}

});

if (index == 0) {

thead = '<thead><tr>' + thead + '</tr></thead>';

} else {

tbody = tbody.concat('<tr>' + tBodyRow + '</tr>');

}

});

tbody = '<tbody>' + tbody + '</tbody>';

if (selectedformat == 'Excel') {

var exportHref = this.tableToExcel(template + caption +thead + tbody + '</table>', this.TenantIDClicked);

var a = document.createElement('a');

a.download = this.TenantIDClicked;

a.href = exportHref;

setTimeout(function () {

a.click();

}, 100);

}

/\*export to PDF\*/

if (selectedformat == 'PDF') {

var that = this;

var columns = [];

var rows = [];

// removal of headerName if having the corresponding values are empty.

var array: any = this.api.getDataAsCsv().split('\n')[0].split(',');

var aa = this.api.getDataAsCsv().split('\n').slice(1);

var header = array.map(function (name, index) {

var count = 0;

aa.map(function (str, ia) {

str.split(',').map(function (value, i) {

if (value.length == 2 && (index == i)) {

count += 1;

}

});

});

if (aa.length == count) {

name = false;

}

return name;

});

var indexes = header.map(function (value, index) {

if (!value) { return index; }

}).filter(function (value) {

if (value !== undefined) { return value; }

});

var body = aa.map(function (str) {

return str.split(',').map(function (value: any, index) {

if (indexes.indexOf(index) !== -1) {

value = false;

}

return value;

})

}).map(function (arr) {

return arr.filter(function (name) { return name; })

});

header = header.filter(function (value) {

return value;

});

var finalRes = [];

finalRes.push(header);

body.map(function (arr) {

finalRes.push(arr);

});

finalRes.map(function (row, index) {

var rowData = {};

row.map(function (name, i) {

name = name.substr(1, name.length - 2);

name = name.replace('"', '');

var obj = {

title: name,

dataKey: name

}

if (index == 0) {

columns.push(obj);

} else {

var rowDataKeys = columns.map(function (obj) {

return obj.dataKey;

});

rowData[rowDataKeys[i]] = name;

}

});

if (index) {

rows.push(rowData);

}

})

// Only pt supported (not mm or in)

var doc = new jsPDF({

orientation: 'l',

unit: 'mm',

format: 'a3',

compress: true,

fontSize: 15,

lineHeight: 1,

autoSize: true,

printHeaders: true

});

console.log(autoTable, jsPDF);

doc.autoTable(columns, rows, {

styles: { overflow: 'linebreak' },

margin: { top: 20 },

addPageContent: function (data) {

doc.text("RDmi Infrastructure" + " " + that.DeploymentName + "-" + that.TenantIDClicked, 150, 10);

}

});

doc.save(this.TenantIDClicked);

}

}

}