Для добавления кнопки, при нажатии которой формируется и скачивается документ электронной таблицы формата xlsx, нужно:

* скачать библиотеку json-as-xlsx и импортировать оттуда xlsx;

import xlsx from "json-as-xlsx";

* создать кнопку;

<button class="btn btn-info ms-2" @click="downloadXLSX">

Скачать отчёт

</button>

* Создать метод «downloadXLSX» и вызвать в нём функцию fetchReports();

function downloadXLSX() {

fetchReports();

}

* Если запрос требует определённых данных, то в метод можно подать нужные аргументы предварительно вычленив их;

function downloadXLSX() {

let ec = document

.getElementById("selececol")

.options[

document.getElementById("selececol")

.selectedIndex

]

.value

.toString(

10

);

fetchReports(ec);

}

* Создать метод асинхронную функцию fetchReports с нужным набором аргументов.
* Нужно запросить нужные данные с помощью метода axios.get(). В аргументе указывается путь API. Потом нужно забрать данные из промиса;

let response = await axios.get(`/api/journals/${subjectId}/school/`);

let res = response.data; //забор данных из промиса

* Далее нужно сформировать два json-словаря: один с данными, другой с параметрами документа;
* Структура словаря с данными:
  + Лист:

sheet: "Название",

* + Названия столбцов:

columns: []

* + - Столбец:

({ label: name, value: alias });

* + Содержимое

content: []

* + - Заполнение:

{ alias: "Andrea" }, { alias: "Luis" }

Структура может масштабироваться в зависимости от нужных полей.

* Структура словаря с параметрами документа:

{

fileName: title, // Name of the resulting spreadsheet

extraLength: 3, // A bigger number means that columns will be wider

writeMode: "writeFile", // The available parameters are 'WriteFile' and 'write'. This setting is optional. Useful in such cases https://docs.sheetjs.com/docs/solutions/output#example-remote-file

writeOptions: {}, // Style options from https://github.com/SheetJS/sheetjs#writing-options

RTL: false, // Display the columns from right-to-left (the default value is false)

}

* Последнее, что нужно сделать – подать в метод xlsx () словарь с данными, а потом с параматрами.

xlsx(data, settings);

* Пример:

async function fetchReports(subjectId) {

let response = await axios.get(`/api/journals/${subjectId}/school/`);

let res = response.data;

let data1 = response["school-journal"];

journal.value = data1;

let header = new Array(); // { label: "User", value: "user" },

let headeralias = new Array();

header.push("ФИО");

headeralias.push("user");

header.push("Предмет");

headeralias.push("suj");

for (var i = 0; i < lesDates.value.length; i++) {

header.push(lesDates.value[i]);

headeralias.push("d" + i.toString());

}

console.log(header, headeralias);

//forming a header

let data = { sheet: "Etuds", columns: [], content: [] };

for (var i = 0; i < header.length; i++) {

data.columns.push({ label: header[i], value: headeralias[i] });

}

//forming content

for (let key0 in groupedStudents.value) {

for (let key1 in groupedStudents.value[key0].lessons) {

let currlist = {

[headeralias[0]]: groupedStudents.value[key0].students\_name,

[headeralias[1]]: key1,

};

for (let i = 2; i < header.length; i++) {

currlist[headeralias[i]] = resMark(key0, key1, header[i]); //etud\_id, lessons\_name, date

}

data.content.push(currlist);

}

}

data = [data];

let ectitle =

document.getElementById("selececol").options[

document.getElementById("selececol").selectedIndex

].text;

let settings = {

fileName: "Отчёт для " + ectitle, // Name of the resulting spreadsheet

extraLength: 3, // A bigger number means that columns will be wider

writeMode: "writeFile", // The available parameters are 'WriteFile' and 'write'. This setting is optional. Useful in such cases https://docs.sheetjs.com/docs/solutions/output#example-remote-file

writeOptions: {}, // Style options from https://github.com/SheetJS/sheetjs#writing-options

RTL: false, // Display the columns from right-to-left (the default value is false)

};

xlsx(data, settings);

}