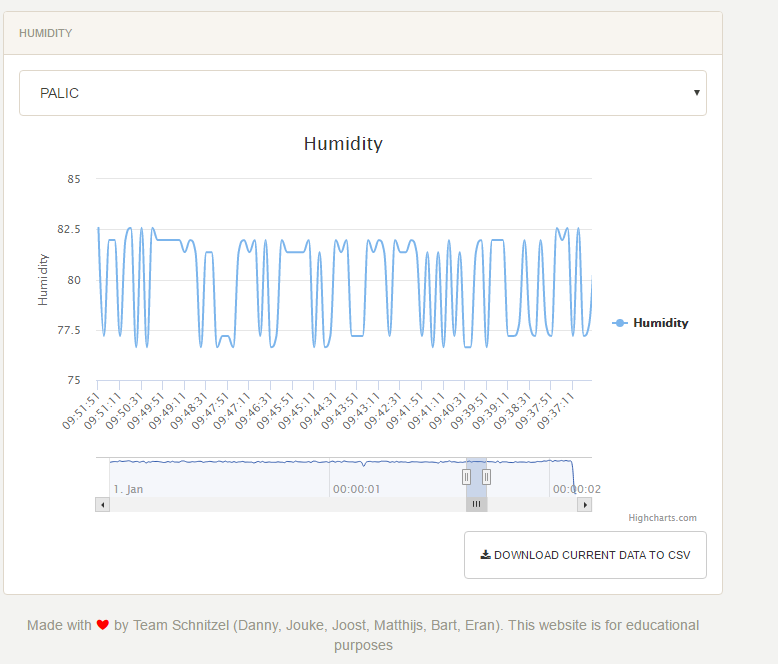
## Using data offline (CSV download)

One of the requirements for the project was the ability to use data offline so it can be used in the field. This paragraph contains the explanation of the implementation.

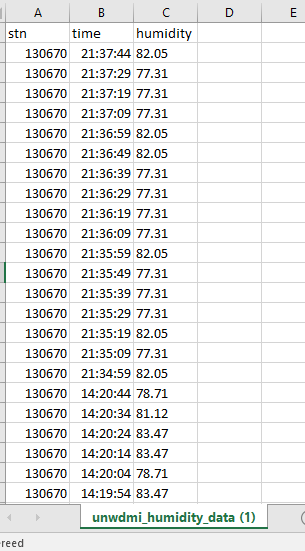
For an example we take the humidity view. This view contains the humidity data of the last 60 minutes. The view contains a download to csv button. This button uses the following URL route /humidity/{id}/download.

The ID of the station is known within the page. When the Serbian station Palic is selected it uses the corresponding ID in this case 130670 so the URL becomes /humidity/130670/download.



Humidity view with the ‘Download current data to csv’ button

As explained earlier in the paragraph ‘receiving the humidity data’. In the controller, which contains all the logic, we read a file and get all the data in an array. For the presentation the array is converted to JSON and loaded with JavaScript. For the download to csv we use the array and wrote a function which parses this array to a comma separated csv file. The route /humidity/{id}/download calls the downloadData() function of the Humidity controller.



Download to CSV logic Actual csv output

In this function. The right headers are set so it doesn’t output on the page. An array is received with the calculateData($id) function. The fields that we want in the headers are set for example: stn, time, humidity. Every value in the array is then converted to a row with a comma separation.

When the conversion is done we close the output fclose($output); The CSV is downloaded.