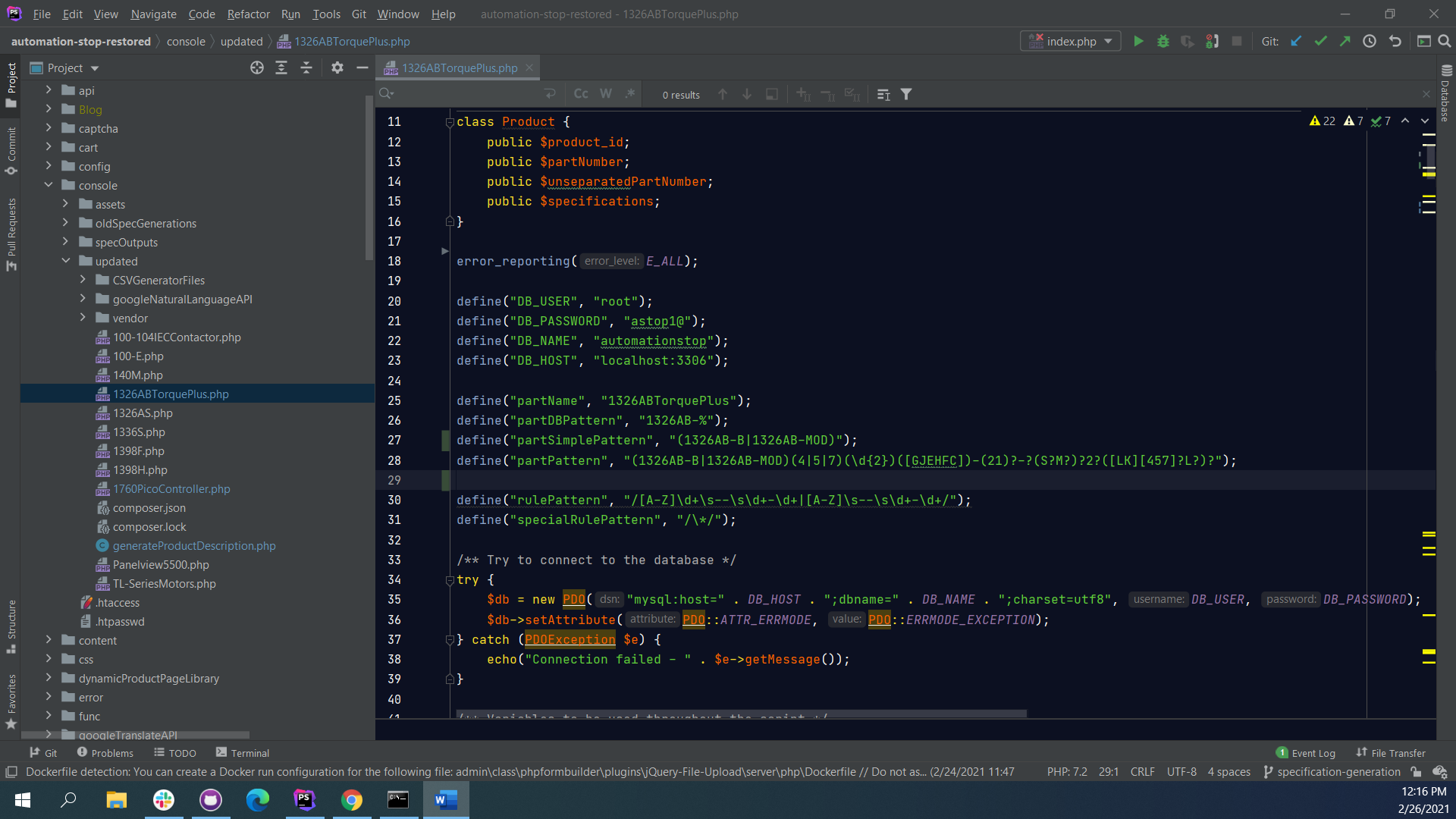
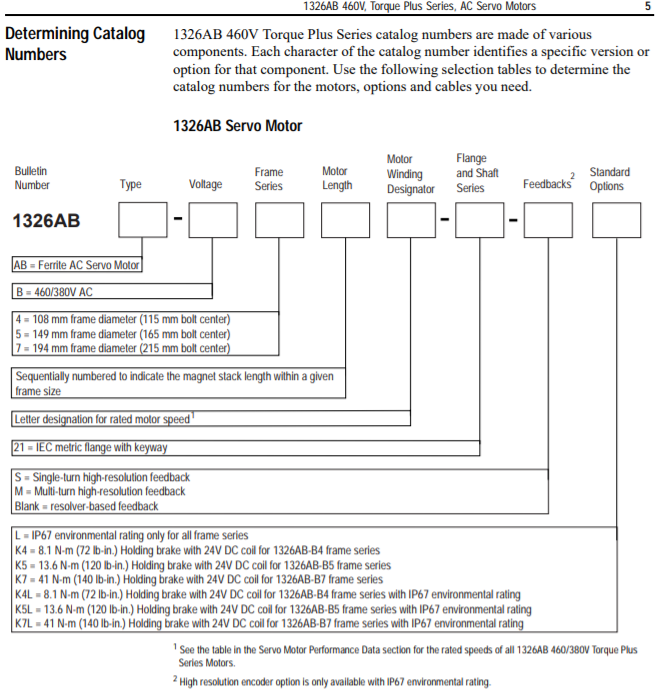
Copy one of the existing generator scripts and rename it to whatever product you are trying to generate for. (the 1326ABTorquePlus is the best one to do this with) These scripts can be found in the specification-generation branch of the automation-stop-restored github repository under automation-stop-restored/console/updated

The scripts are set up so that you should be able to fully generate specs for a product by only changing a couple of defined variables at the top of the script.  Go ahead and change partName to whatever the name of the product you’re working with is. Make sure that this value is the same as the name of the file you are using just without the .php extension. Once you are done with that run the script just to make sure all references to ‘1326ABTorquePlus’ are gone and your product name is there instead.

The next thing you’re going to want to do is figure out what the part numbers you are going to be working with start with. You can find this in a couple different ways such as looking through the product manual, searching the products table in the database, or browsing through the website you are generating specs for. Most of the time the product manual will give you something like this:  Here you can see that all 1326AB products start with the characters ‘1326AB’ and the torque plus series in particular is denoted by the letter ‘B’ thus the starting prefix for the torque plus series is ‘1326AB-B’. Now that we have this we can go ahead and set the partSimplePattern to whatever your part prefix is (1326AB-B) and set partDBPattern to the product line prefix with a ‘%’ SQL wildcard character attached to the end (1326AB- %). You’re also going to want to change partPattern to “” for now. At this point you should be able to run the script and select the radio button to see all of the products in the 1326AB product line in a table. The partDBPattern handles what the database query gets from the database and the partSimplePattern handles what products the script tries to generate specs for. I like to see the entire product line when working on these just to see if there is any additional information that can be matched up between the manual and the products we currently support but this can easily be changed to suit your preferences.

Now we’re going to start working on our CSV Generator file. Go to the CSVGeneratorFiles directory and copy the 1326ABTorquePlusspec\_default.csv file and rename it to your products name. Each column can either be a specification or a rule column. Rule columns are signified by a series of characters in the format (Letter) – (number)-(number) where the letter can be any character you want, the first number should be a 1 and the second character is the length of the rule. The ‘rules’ in the rule columns are essentially just the part of the part number that corresponds to an individual set of specifications. In the case where you have rules of varying lengths for a single set of specifications just use the shortest length value. Each rule column will correspond to every single non-rule column that comes after it until it runs into the next rule column. The first three specifications need to be the Manufacturer, Series, and Product Type in that order from left to right. Use the ‘-‘ character whenever a specification for a given rule should be blank.

The script uses dash ‘-‘ characters to separate individual rules when reading the products part number. Sometimes you will get lucky and every part of the part number that corresponds to a set of specifications will already be separated by dashes. In this case you can leave the partPattern variable blank and the script will automatically work fine. Often this is not the case, however, and you will need to make a regular expression that has a capture group for every part of the part number that corresponds to a set of specifications. <https://regex101.com/> is a great resource for making this, you basically just need to make sure that you take into account where there are already dash characters in the part number and it will put dashes between the capture groups you specify. Set the partPattern variable equal to this regular expression. If you want to see where the script ends up putting these dashes then change $product->unseparatedPartNumber to $product->partNumber on line 324 just make sure you change this back when you’re done or else it will insert the separated part number for the MPN spec which we do not want.