How to improve Performance for Datatables in Laravel?

Datatable is a powerful and an effective solution for creating table listings and adding interactive functionalities for it. It provides all the functionalities like pagination, searching, sorting out of the box with such an ease that one need not do any kind of tough work to get them work as needed.

But to make it work in the most effective and efficient way, there is need to make some decisions regarding its use. No doubt, It is able to handle lakhs of records really smoothly but wrong way of using or ill-way of using it might turn out to be problematic instead of actually being a solution.

This post will help you people to analyse your problem and then choose the best way to get this really helpful and powerful solution implemented in your application.

Analysing the Requirement

While being into web development, many developers like me need to show the data in tabular form and then provide some actions over them so as to manipulate the data row-by-row (like add, edit or delete) or implement some other operations like sorting and searching. The best solution in terms of ease and speed is Datatables. It works like charm as far as the number of records are few thousands, but as the number grows say above 20,000 or so, the problem starts.

So a page with around more than 20,000 records was taking a lot of time(say, around 1 minute, which is actually really long in consideration to today’s fast generation) even after making use of Datatables with the simplest configuration.

I am sure many of the developers must have faced the same problem and must have spent a lot of hours to find the correct solution at one place.

Data tables usage approaches

After struggling at various forums and going through many discussions about it , I came up with 2 solutions to it both following different approaches; first one being on Client side.

Client side

Initially the approach that was being used was on the data that was being fetched in the controller itself and was passed to the view further and then was shown and manipulated there as required. Lets just make steps to to make it simple and comprehensive:

- Fetch data in controller
- Manipulate data in controller itself to show it as required in view
- After loading all the data, datatables paginate the data in view

Now, the issue with this approach was that it took really long to load the page as on page load all the data that was around 20 thousand records were fetched that further degraded the speed. And therefore led to frustrated user experience.

The Code snippet is as follows:

Controller function to call view initially is as below-

Code snippet of function called on ajax is as follows:

The view snippets should be like as given below:

The datatable configuration on this page is:

The Pros of this approach is that, the time for operations like searching, sorting and pagination is noticeably small which made these really fast for users.

The Cons for it is that it takes really long to load the page. No doubt, this approach decreased the load time to good level but was still very slow considering user satisfaction. So, I after doing the needful R&D, the other solution we moved on to was Server Side solution.

The time for request can be seen in the picture given below:

2. Server side

This Solution was really effective because it restricted all the manipulations and operations to controller and view just loaded the HTML. Moreover, the data pagination was implemented on server side which further reduced the overall execution time.

Just to make comparison with the above mentioned approach, the steps for this approach are as follows:

- Fetch data in controller

● Manipulate data and paginate the data in controller

● Return data plus some meta-data like total records and records filtered count.

Now the ajax request is specified in datatables to get the data of the table. Moreover, every operation such as searching, pagination, and sorting is done on server side with appropriate ajax request.

The Controller function to call view remains same as in the former case.

The only change exists in method called via ajax and datatable configuration as below:

(Tip: To show the proper page numbers and allow paging work properly it is very important to send the recordsTotal, recordsFiltered data)

And the Datatable configuration is below:

The Pros of this approach is that the total execution time is equally divided among the server side operations and client side operations, thereby, reducing the waiting time to load the page.

The Cons of this approach as such is nothing except that now the operations that were given by datatables out of the box, now needs to be coded on the server side.

The Given below image will show the drastic difference (from 27.53s to 2.11s only) this approach made to the request time:

Which one to Choose?

Both the approaches are having their own advantages and disadvantages, but depending upon the scenario we can choose one to follow, so that we could eliminate any disadvantages.

From the above discussion, it can be easily analyzed that the first approach is really efficient when the number of records to be viewed are less than around 15000 records whereas the second one is applicable when the data set to be operated is very large in size, let’s say, above 20000 thousand.

About Author

The author is a Web Developer at with Experience in **Laravel** and Passionate about learning emerging frontend technologies like jquery, angularjs.