1. Measuring the Database Performance:

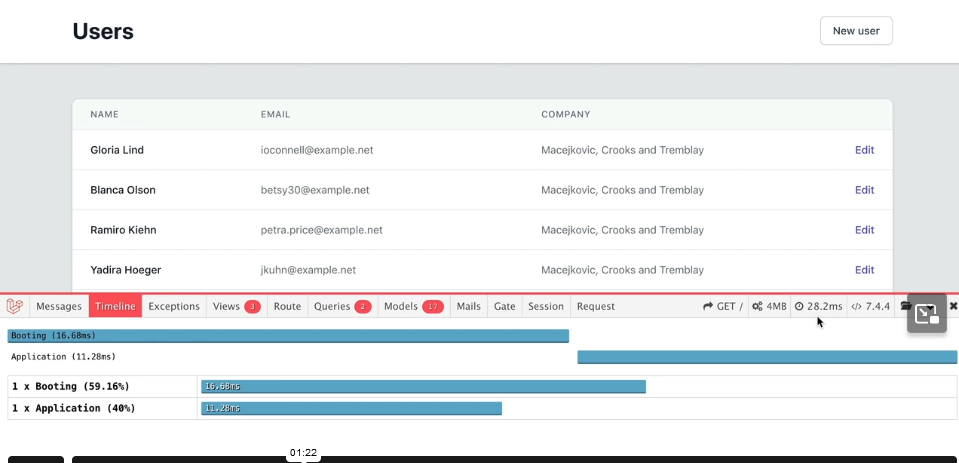
If you are not doing this then you are flying blind as you are incapable of identifying problem requests where too many queries are being run, too much data is being parsed, high complexity is occurring, etc.

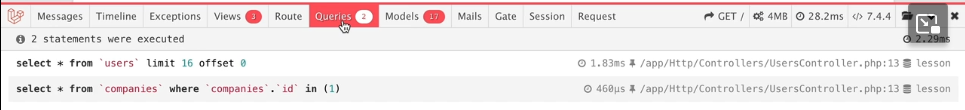
Recommended to apply the Laravel Debug Bar tool, allowing you to see all queries being run on the system. Install it as a dev dependency as we only need it locally:



We do not want this running in production as sensitive database info will be exposed.

The total request duration is highly important and should optimally be below 200ms. The other important factor is the number of queries being run, as this is a key point for optimising.



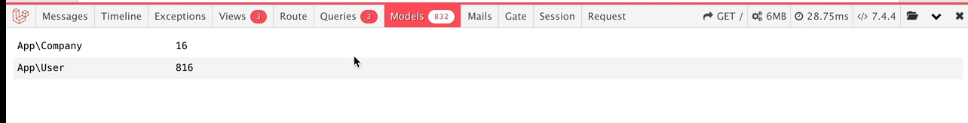


This is important for avoiding the n+1 problem, where lazy loading is applied to eloquent relationships and a query is generated for each record where a relationship is present. The better approach is eager loading via the ‘with’ method, to pre-load all relationships, generating only 1 or 2 queries.

When running orders by clauses, it is highly likely that we will be needing an index. This vastly improves performance and should be done on the database layer.



The amount of memory your application uses is directly correlated to it’s performance. We can just this more succinctly by checking the models tab:



We should only load the models that we are showing on the page. In other words, pagination is exceptionally important.

1. Minimize Memory Usage: