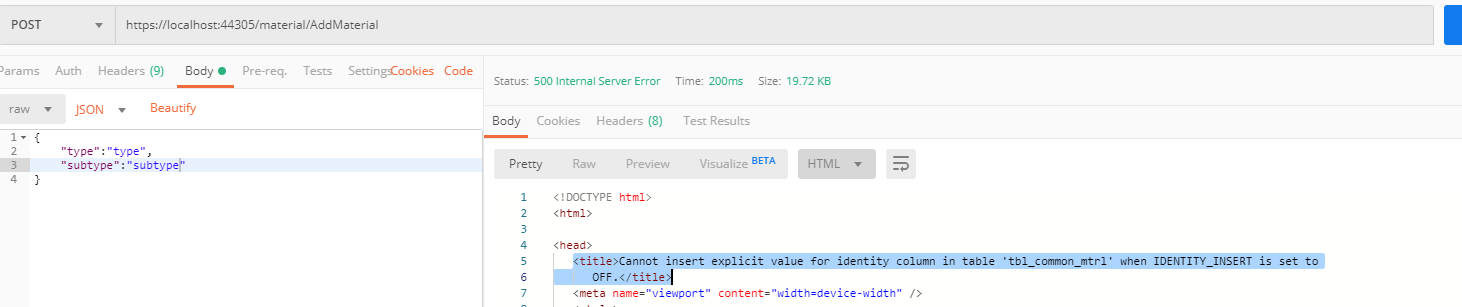
**Begin POST new record problem.**

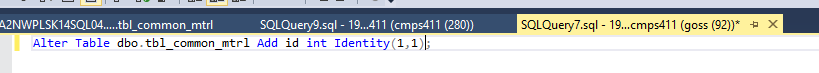
Why can’t I post a new record with axios or ajax? Let’s use Postman to try to determine the reason.



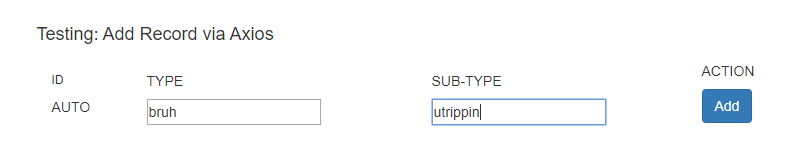
What does this mean? I did have to delete the original identity column today and add another because the identity column didn’t iterate when I used Axios to post a new record for Materials through the web browser. I feel like this is a SQL problem at the data level.

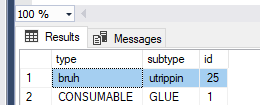
In order to best solve this problem, I need to be able to POST a new record to the Material controller through Postman first. That way, I can eliminate problems with jquery in the front end.

Figured it out. I ended up deleting the identity field id and making it again. Something went goofy the first time over the fact that a field named id existed, but wasn’t an identity field that auto incremented. The fix is to delete the non incrementing id field and then use a sql statement to make a field named id that is an identity field that increments. Works like a champ now.



Works through the user interface.

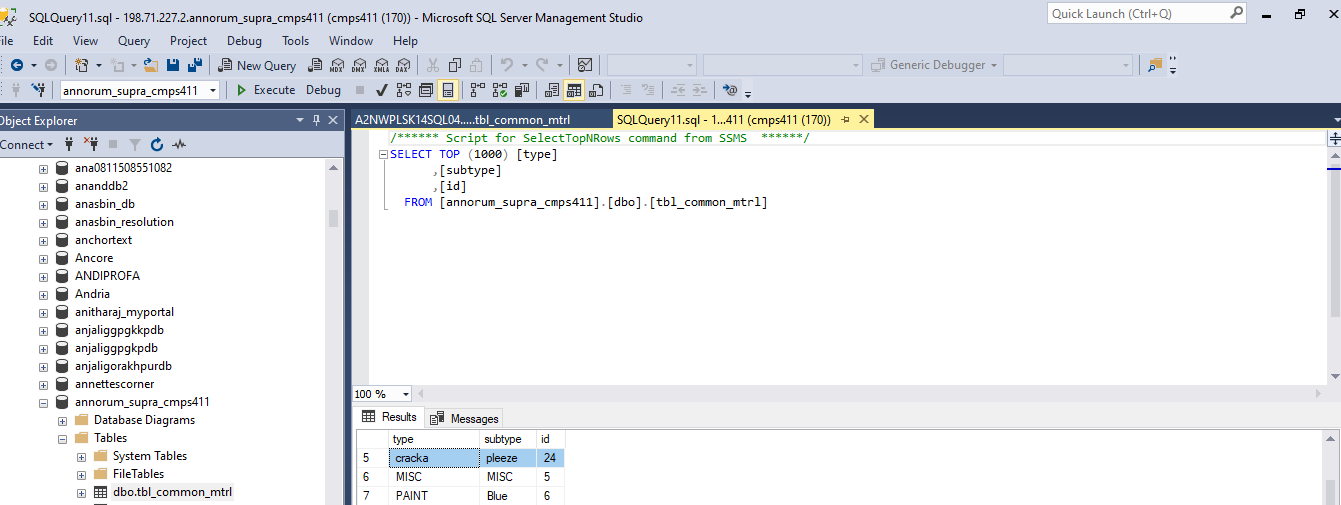


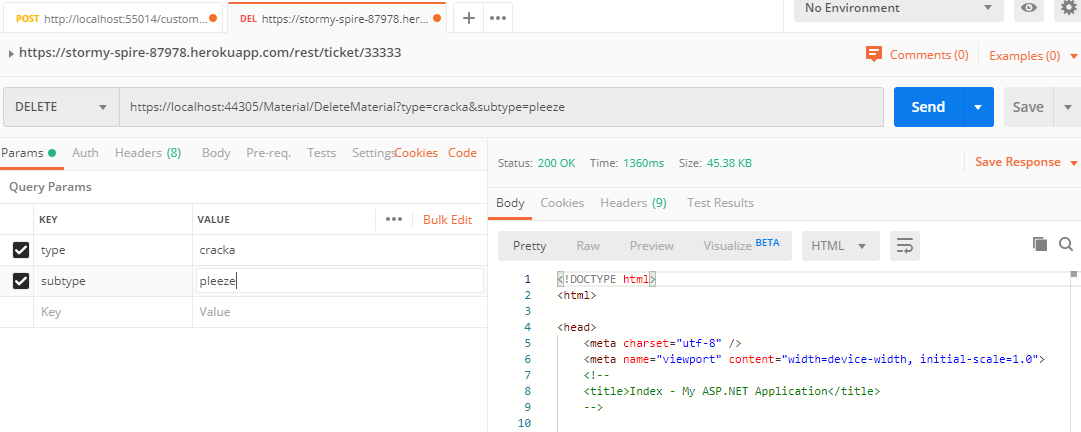


**End POST new record problem.**

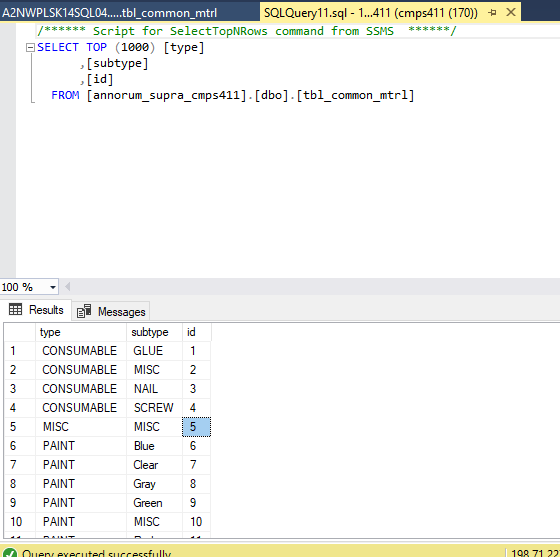
**Begin DELETE a record row problem.**

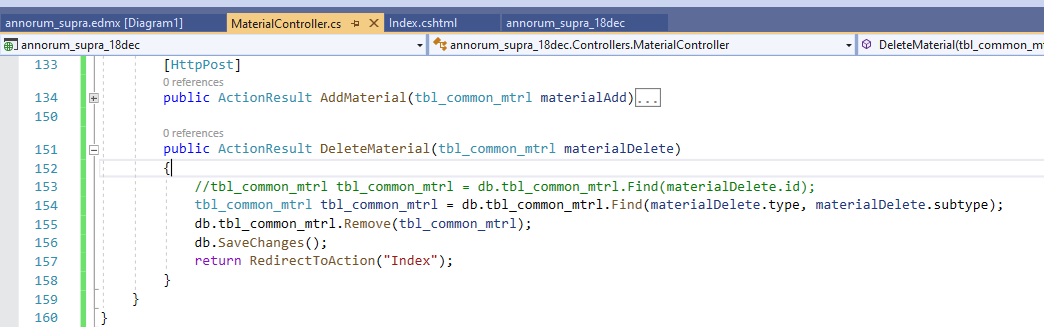
I can delete records when the two keys are passed through Postman, but I can’t delete them from the user interface. Why?



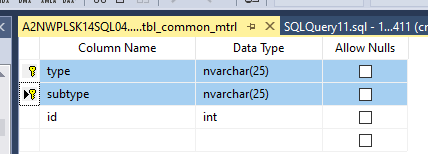


Gone from the database.

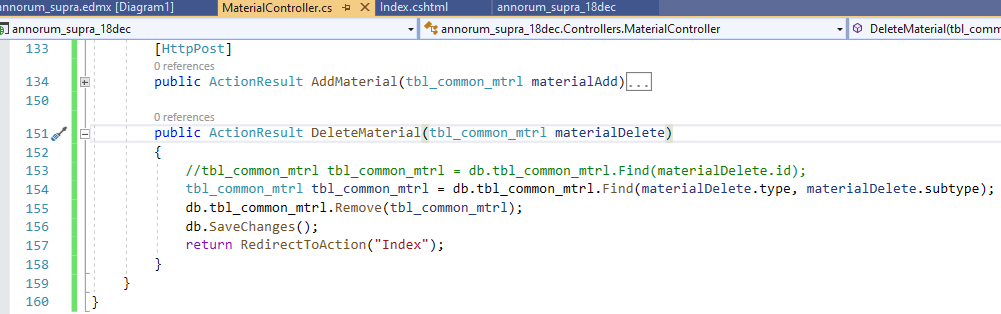




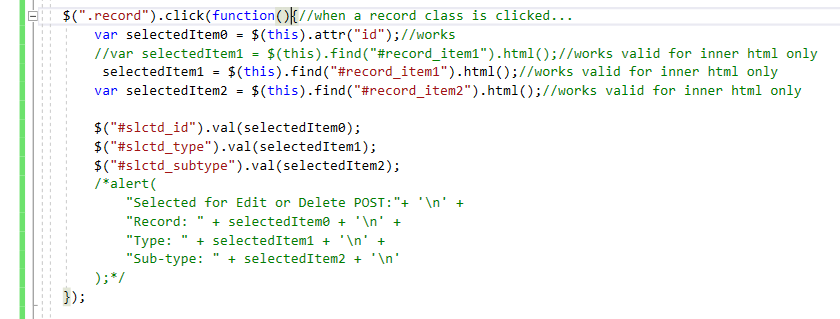
This method in the controller uses key fields to delete records…



…this table has a composite key made up of type and subtype. I tried to delete a record using the identity field of id first, but it didn’t work. No biggie. I just need to figure out how to pass type and subtype to the controller and route through the user interface…



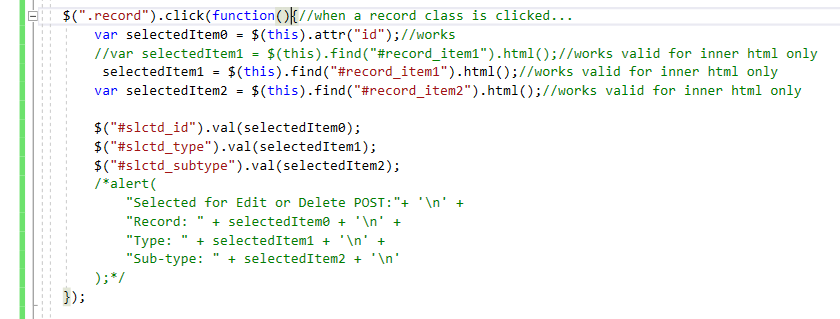
Found it. I’m relying on this record click event to determine the record that has been clicked on…



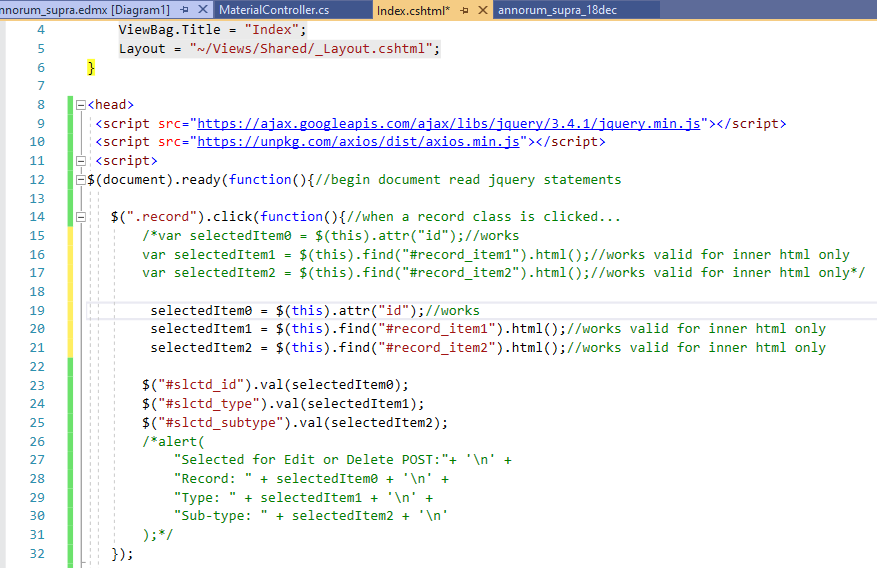
My confirm delete event is in another function…

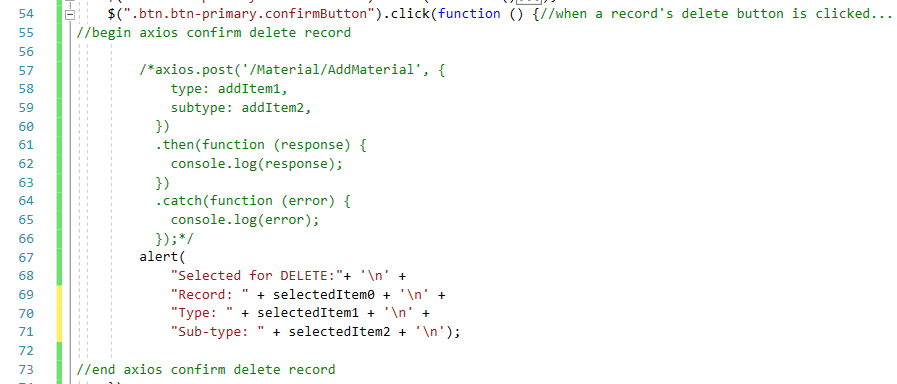


It’s a scope problem. If I remove var from the variable declarations in the record click event, I will be able to access what I need from any other function…

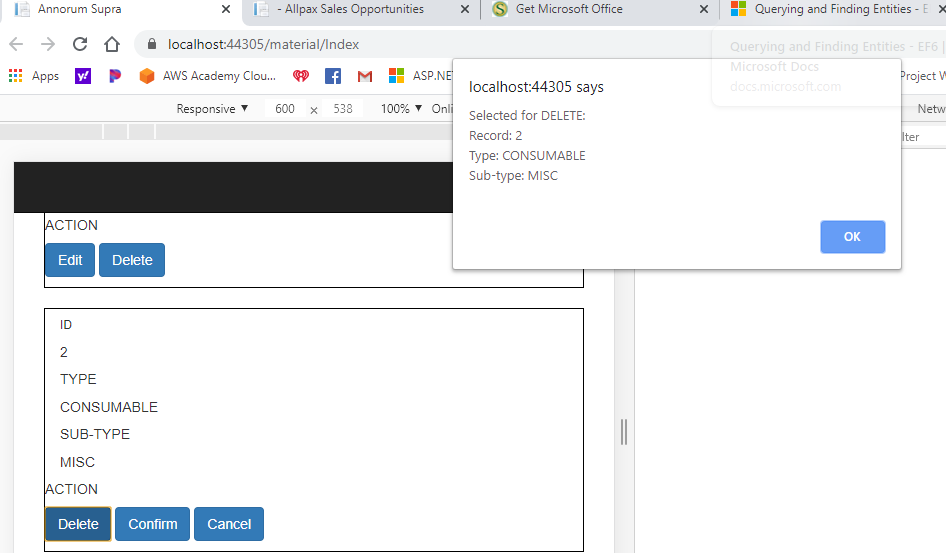


Let’s test that idea further…

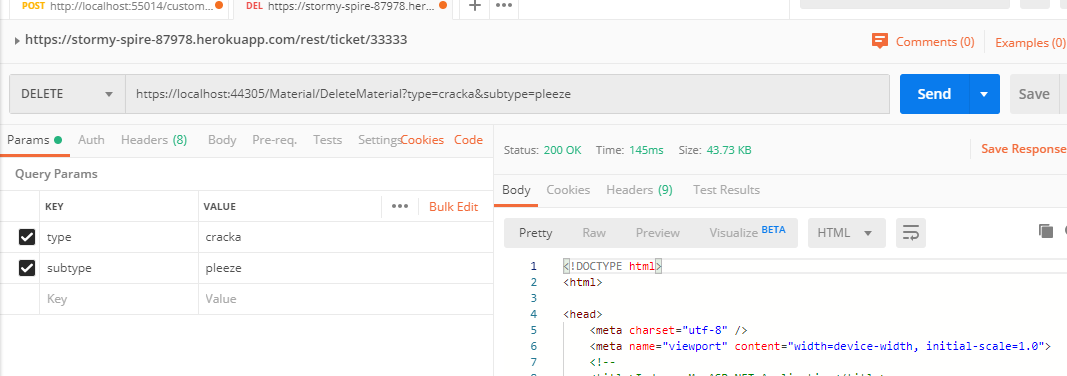




Works. Baby steps are the way to go when it comes to debugging. Another part is documentation with screen shots.

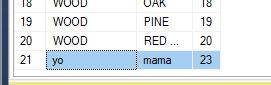


So, now that I’ve proven with a pop up that I can get the right information, all I need to do now is link it up to what I was able to do with Postman in the beginning. All I did was pass values for a composite key, type and subtype…

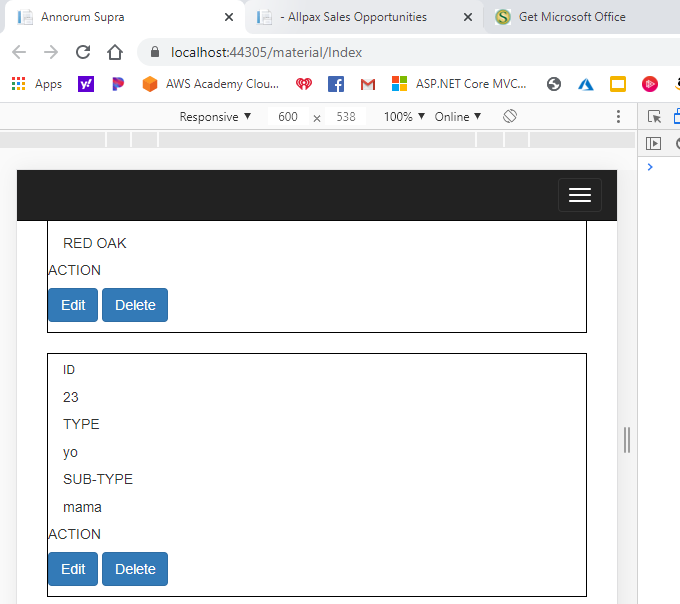


This record was deleted in an earlier step, so let’s try to use the OI to get rid of this test record…

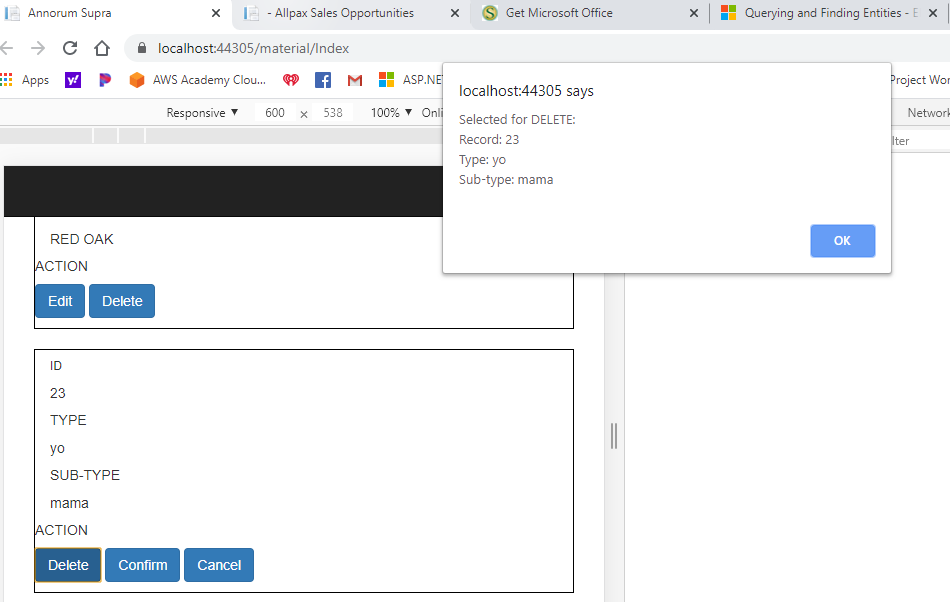
It is shown in the sql database and…



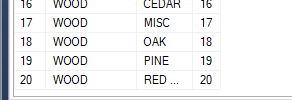
In the OI.



Let’s press the delete button in the OI.



Gone from the database.



This is the logic that made it work…



**End DELETE a record row problem.**

**Begin EDIT a record**

**End EDIT a record.**