# 542.https://stackoverflow.com/questions/70203931/error-transaction-reverted-function-call-failed-to-execute

**T:**Error: Transaction reverted: function call failed to execute

**Q:**I am currently working on an nft marketplace and wanna integrate a 2.5% fee on every sale. I have read and reasearched online on how to calcuate percentage in solidity and was successful on that part but the problem is that result its a huge number i.e 2.5 \* 150/100 is supposed to return 3.75but i get this figure instead: 3750000000000000000000000000000000000  
  
My test  
  
const auctionPrice = ethers.utils.parseUnits("1", "ether");const [\_, userAddress] = await ethers.getSigners(); let fee = (2.5 \* auctionPrice) / 100; const commission = ethers.utils.parseUnits(fee.toString(), "ether"); let commissionValue = commission.toString(); console.log(commission.toString()); console.log(fee); await market .connect(userAddress) .createMarketSale(nftContractAddress, 1, commissionValue, { value: auctionPrice, });  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The Result  
  
250000000000000000000000000000000 1 failing 1) NFTMarket Should create and execute market sales: Error: Transaction reverted: function call failed to execute at NFTMarket.createMarketSale (contracts/NFTMarket.sol:165) at processTicksAndRejections (node:internal/process/task\_queues:96:5) at runNextTicks (node:internal/process/task\_queues:65:3) at listOnTimeout (node:internal/timers:526:9) at processTimers (node:internal/timers:500:7) at HardhatNode.\_mineBlockWithPendingTxs (node\_modules\hardhat\src\internal\hardhat-network\provider\node.ts:1602:23) at HardhatNode.mineBlock (node\_modules\hardhat\src\internal\hardhat-network\provider\node.ts:435:16) at EthModule.\_sendTransactionAndReturnHash (node\_modules\hardhat\src\internal\hardhat-network\provider\modules\eth.ts:1494:18)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So my question is there any way to turn this number (3750000000000000000000000000000000000) into a decimal or whole number? I have searched but couldnt find any thing online talking about this

1 **Answer**

**A1:**In fact there is no problem with your calculation. The problem is the use of the function ethers.utils.parseUnits().  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Under the hood, all transactions in Ethereum are paid in wei, that is the smallest denomination of ether. So, if you send 1 ETH to some contract, you are actually sending 1,000,000,000,000,000,000 wei, because 1 ETH = 10^18 wei.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
In the tests, when you use the function ethers.utils.parseUnits("1", "ether"), you are atually saing "I want to send 1 ETH, so please convert this 1 ETH to wei to make my life easier". Then this function simply adds 18 zeros to your "1" string.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Finally, when you perform your calculation (2.5 \* auctionPrice) / 100, the result are already in the right format. So you don't have to call perseUnits again, because you are adding more 18 zeros! =D  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**But why does it through an error when I want to transfer that amount to the contract owner ``` payable(owner).transfer(listingPrice); ```

**C2:**A bit late on the conversation, but, Were you able to solve it? If so, can you share your solution?, otherwise, Can you post the contract (or at least the portion of it) that it's throwing that error.