# 622.https://stackoverflow.com/questions/69152527/code-sample-for-parsing-metadata-for-solana-nft-and-updating-the-metadata

**T:**Code sample for parsing metadata for Solana NFT and updating the metadata

**Q:**On Solana, NFT metadata is stored in accounts which are owned by the shared contract Token Metadata Program at address metaqbxxUerdq28cj1RbAWkYQm3ybzjb6a8bt518x1s.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I need a clear, concise code example for how I can use some existing library to obtain the metadata for a particular NFT. Let's use this NFT for example: a SolStone 4itgFt6tSotypyVAaUkLJzpGQ5KXsJNhwpKBANMv49mf  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The furthest I've gotten so far is copying over the metaplex library and using a call like so  
  
 await getProgramAccounts(connection, METADATA\_PROGRAM\_ID, 'finalized');  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
however this will load the metadata for all NFTs in existence. I instead need to use some filter instead of 'finalized' as the commitment argument.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
In the metaplex codebase they have an example of filtering by creator's address I believe.  
  
filters: [ { memcmp: { offset: 1 + // key 32 + // update auth 32 + // mint 4 + // name string length MAX\_NAME\_LENGTH + // name 4 + // uri string length MAX\_URI\_LENGTH + // uri 4 + // symbol string length MAX\_SYMBOL\_LENGTH + // symbol 2 + // seller fee basis points 1 + // whether or not there is a creators vec 4 + // creators vec length i \* MAX\_CREATOR\_LEN, bytes: whitelistedCreators[j].info.address, } }]  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So in theory I should be able to update this filter to instead filter by the NFT's own address, but I'm not sure how to get the bit offset correct here to make the filter accurate.  
  
After I am able to read the nft metadata, next I need a code example for updating the metadata. Specifically the URI field.

1 **Answer**

**A1:**You're almost there on most of this!  
  
The address 4itgFt6tSotypyVAaUkLJzpGQ5KXsJNhwpKBANMv49mf is in fact a token mint, as seen on the explorer: https://explorer.solana.com/address/4itgFt6tSotypyVAaUkLJzpGQ5KXsJNhwpKBANMv49mf  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
This means that you'll use a very similar call as that example you found, but instead of filtering on the whitelistedCreators array, you'll filter on the position of the mint, which is at byte 1 + 32, assuming we can trust the comment in the code. That gives a function call of:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
getProgramAccounts(connection, METADATA\_PROGRAM\_ID, { filters: [ { memcmp: { offset: 1 + // key 32 // update auth bytes: mintAddress.toBytes(), }, }, ],})  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
For updating the metadata, you'll likely want to use the UpdateMetadata instruction. It looks like the Metaplex JS library has a utility function to do just that: https://github.com/metaplex-foundation/metaplex/blob/9d5a5c6d668cd9c597cff8c63dfba00dee2f72f0/js/packages/common/src/actions/metadata.ts#L481  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]