# 657.https://stackoverflow.com/questions/68118203/figuring-out-how-to-mint-nft-with-delayed-release-metadata-not-validating-on-op

**T:**Figuring out how to mint NFT with delayed release, metadata not validating on opensea

**Q:**I have a smart contract working on rinkeby which is minting ERC721 tokens. The goal is to allow minting for a week and then to release the metadata and images after a week has passed.  
  
When I deploy it, I use https://gateway.pinata.cloud/ipfs/ as the baseUri that I feed into the contract.  
  
Next, im simulating a delayed metadata/image release:  
  
 ● After the tokens are minted, I run a curl POST to add the image to  
IPFS and get its hash.  
  
  
 ● I then manually pin the image to pinata via UI and create metadata  
using the pinned image uri.  
  
  
 ● The image link in the metadata resolves in my browser.  
  
  
 ● Next, I run a second curl POST to add the JSON metadata to IPFS.  
  
  
 ● I manually pin the metadata to pinata via UI:  
https://gateway.pinata.cloud/ipfs/Qmf43nycxRXdwdq3GqT7SKU3MvRyBvbpjzufzZR8ASkMGb  
  
  
 ● At this point, Opensea shows no image and its metadata cannot  
validate:  
https://testnets.opensea.io/assets/0x1b566f6946feeb73e178c63a2992c2365c6cd55f/0  
- https://rinkeby-api.opensea.io/asset/0x1b566f6946feeb73e178c63a2992c2365c6cd55f/0/validate/  
  
  
The image\_url metadata property is failing in validation because ipfs links use a hash and im using the integer token value. I cant see to figure out how to bridge this gap since the hash is generated outside of the contract and it looks like the contract is what is generating this link in the metadata.  
  
Could anyone shine a light on my dilemna? Thank you! :D

**C1:**I was close... needed to run the OppenZeppelin \_setTokenURI for each minted token to accomplish this ...

1 **Answer**

**A1:**The answer lied in needing to structure metadata using a folder hash in IPFS before launch and then setBaseURI!

**C1:**Im doing exactly the same as you. can you please elaborate on folder hash and your solution ?

**C2:**Sure! Basically you just need an API which returns your metadata: ex - api.com/1 would return metadata for token 1. Once you have that, use the setBaseURI method in your contract to be the address of your api with a trailing slash - api.com Same story if you are using IPFS - you have to put all of your metadata in one folder and upload it all at once, this makes it so that the FOLDER has its own unique address. ipfs.io/ipfs/bafybeigpbly.../0 ipfs.io/ipfs/bafybeigpbly.../1 ipfs.io/ipfs/bafybeigpbly.../2 Note how the folder is the same.

**C3:**in the above comment https was removed from the links

**C4:**setbaseuri must be populated as such: h t t p s : / / api.com / <-- trailing slash is important

**C5:**thanks for this info! but in this case users might realize your pattern and sniff out all the metadata before the reveal ?