

[Logout](#)

## PROJECT SPECIFICATION

## Create Your Own Private Blockchain

## Complete unfinished block.js implementation

CRITERIA	MEETS SPECIFICATIONS
Modify the <code>validate()</code> function to validate if the block has been tampered or not.	<ul style="list-style-type: none"><li>-Return a new promise to allow the method be called asynchronous.</li><li>-Create an auxiliary variable and store the current hash of the block in it (this represent the block object)</li><li>-Recalculate the hash of the entire block (Use SHA256 from crypto-js library)</li><li>-Compare if the auxiliary hash value is different from the calculated one.</li><li>-Resolve true or false depending if it is valid or not.</li></ul>

CRITERIA	MEETS SPECIFICATIONS
Modify the 'getBData()' function to return the block body (decoding the data)	<ul style="list-style-type: none"><li>-Use hex2ascii module to decode the data</li><li>-Because data is a javascript object use JSON.parse(string) to get the Javascript Object</li><li>• Resolve with the data and make sure that you don't need to return the data for the genesis block OR reject with an error.</li></ul>

### Complete unfinished blockchain.js implementation

CRITERIA	MEETS SPECIFICATIONS
Modify the '_addBlock(block)' function to store a block in the chain	<ul style="list-style-type: none"><li>• Must return a Promise that will resolve with the block added OR reject if an error happen during the execution.</li><li>• height must be checked to assign the previousBlockHash<ul style="list-style-type: none"><li>-Assign the timestamp &amp; the correct height</li><li>-Create the block hash and push the block into the chain array.</li></ul></li></ul> <p>Don't forget to update the <code>this.height</code></p>

CRITERIA	MEETS SPECIFICATIONS
Modify 'requestMessageOwnershipVerification(address)' to allow you to request a message that you will use to sign it with your Bitcoin Wallet (Electrum or Bitcoin Core)	<ul style="list-style-type: none"><li>• must return a Promise that will resolve with the message to be signed</li></ul>
Modify 'submitStar(address, message, signature, star)' function to register a new Block with the star object into the chain	<ul style="list-style-type: none"><li>• must resolve with the Block added or reject with an error.</li><li>• time elapsed between when the message was sent and the current time must be less than 5 minutes</li><li>• must verify the message with wallet address and signature: <code>bitcoinMessage.verify(message, address, signature)</code></li><li>• must create the block and add it to the chain if verification is valid</li></ul>
Modify the 'getBlockHeight(hash)' function to retrieve a Block based on the hash parameter	<ul style="list-style-type: none"><li>• must return a Promise that will resolve with the Block</li></ul>

CRITERIA	MEETS SPECIFICATIONS
Modify the 'getStarsByWalletAddress (address)' function to return an array of Stars from an owners collection	<ul style="list-style-type: none"><li>• must return a Promise that will resolve with an array of the owner address' Stars from the chain</li></ul>
Modify the 'validateChain()' function	<ul style="list-style-type: none"><li>• must return a Promise that will resolve with the list of errors when validating the chain</li><li>• must validate each block using validateBlock()</li><li>• Each Block should check with the previousBlockHash</li><li>• execute the <code>validateChain()</code> function every time a block is added</li><li>• create an endpoint that will trigger the execution of <code>validateChain()</code></li></ul>

### Test your App functionality

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Use 'POSTMAN' or similar service to test your blockchains endpoints and send screenshots of each call	<ul style="list-style-type: none"><li>• must use a GET call to request the Genesis block</li><li>• must use a POST call to requestValidation</li><li>• must sign message with your wallet</li><li>• must submit your Star</li><li>• must use GET call to retrieve starts owned by a particular address</li></ul>