

[< Back to Blockchain Developer](#)

Decentralized Star Notary

REVIEW

CODE REVIEW 7

HISTORY

▼ index.html 3

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"
6   <title>Star Notary</title>
7   <link rel="stylesheet" type="text/css" href="style.css">
8
9   <script src="https://cdn.jsdelivr.net/gh/ethereum/web3.js/dist/web3.m
10 </head>
11
12 <body>
13 <div class="container">
14   <h1>Star Notary</h1>
15
16   <div class="row">
17     <label class="w-1/2 bold" for="star-name">Star name:</label>
18     <input class="w-1/2" id="star-name" name="star-name" required>
19   </div>
20
21   <div class="row">
```

```

22   <label class="w-1/2 bold" for="star-story">Star story:</label>
23   <input class="w-1/2" id="star-story" name="star-story" required>
24 </div>
25
26 <div class="row">
27   <label class="w-1/2 bold" for="star-ra">ra:</label>
28   <input class="w-1/2" id="star-ra" name="star-ra" required>
29 </div>
30
31 <div class="row">
32   <label class="w-1/2 bold" for="star-dec">dec:</label>
33   <input class="w-1/2" id="star-dec" name="star-dec" required>
34 </div>
35
36 <div class="row">
37   <label class="w-1/2 bold" for="star-mag">mag:</label>
38   <input class="w-1/2" id="star-mag" name="star-mag" required>
39 </div>
40

```

SUGGESTION

For better user experience, it would be better if you have a separate input field for the token field can be confusing.

```

41   <button id="claim-button" onclick="claimButtonClicked()">Claim Star</
42
43 <h4>Get claimed star </h4>
44
45 <div class="row">
46   <label class="w-1/2 bold" for="star-tokenId">Token id:</label>
47   <input class="w-1/2" id="star-tokenId" name="star-tokenId" requirec
48 </div>
49
50 <div class="row">
51   <label id="star-token-result" class="w-full"></label>
52 </div>
53
54   <button id="info-button" onclick="infoButtonClicked()">Get star</butt
55 </div>
56
57 <script>
58   if (typeof web3 !== 'undefined') {
59     web3 = new Web3(web3.currentProvider) // what Metamask injected
60   } else {
61     // Instantiate and set Ganache as your provider
62     web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:8
63   }
64   // The default (top) wallet account from a list of test accounts
65   web3.eth.defaultAccount = web3.eth.accounts[0]
66   // The interface definition for your smart contract (the ABI)

```

```
67 var StarNotary = web3.eth.contract(  
68   [  
69     {  
70       "constant": true,  
71       "inputs": [  
72         {  
73           "name": "interfaceId",  
74           "type": "bytes4"  
75         }  
76       ],  
77       "name": "supportsInterface",  
78       "outputs": [  
79         {  
80           "name": "",  
81           "type": "bool"  
82         }  
83       ],  
84       "payable": false,  
85       "stateMutability": "view",  
86       "type": "function"  
87     },  
88     {  
89       "constant": true,  
90       "inputs": [  
91         {  
92           "name": "",  
93           "type": "uint256"  
94         }  
95       ],  
96       "name": "starsForSale",  
97       "outputs": [  
98         {  
99           "name": "",  
100          "type": "uint256"  
101        }  
102      ],  
103      "payable": false,  
104      "stateMutability": "view",  
105      "type": "function"  
106    },  
107    {  
108      "constant": true,  
109      "inputs": [  
110        {  
111          "name": "tokenId",  
112          "type": "uint256"  
113        }  
114      ],  
115      "name": "getApproved",  
116      "outputs": [  
117        {  
118          "name": "",  
119          "type": "address"  
120        }  
121      ]  
122    }  
123  ]  
124 )
```

```
121     ],
122     "payable": false,
123     "stateMutability": "view",
124     "type": "function"
125   },
126   {
127     "constant": false,
128     "inputs": [
129       {
130         "name": "to",
131         "type": "address"
132       },
133       {
134         "name": "tokenId",
135         "type": "uint256"
136       }
137     ],
138     "name": "approve",
139     "outputs": [],
140     "payable": false,
141     "stateMutability": "nonpayable",
142     "type": "function"
143   },
144   {
145     "constant": true,
146     "inputs": [
147       {
148         "name": "",
149         "type": "uint256"
150       }
151     ],
152     "name": "tokenIdToStarInfo",
153     "outputs": [
154       {
155         "name": "name",
156         "type": "string"
157       },
158       {
159         "name": "ra",
160         "type": "string"
161       },
162       {
163         "name": "dec",
164         "type": "string"
165       },
166       {
167         "name": "mag",
168         "type": "string"
169       },
170       {
171         "name": "story",
172         "type": "string"
173       }
174     ],
```

```
175     "payable": false,  
176     "stateMutability": "view",  
177     "type": "function"  
178   },  
179   {  
180     "constant": false,  
181     "inputs": [  
182       {  
183         "name": "from",  
184         "type": "address"  
185       },  
186       {  
187         "name": "to",  
188         "type": "address"  
189       },  
190       {  
191         "name": "tokenId",  
192         "type": "uint256"  
193       }  
194     ],  
195     "name": "transferFrom",  
196     "outputs": [],  
197     "payable": false,  
198     "stateMutability": "nonpayable",  
199     "type": "function"  
200   },  
201   {  
202     "constant": false,  
203     "inputs": [  
204       {  
205         "name": "from",  
206         "type": "address"  
207       },  
208       {  
209         "name": "to",  
210         "type": "address"  
211       },  
212       {  
213         "name": "tokenId",  
214         "type": "uint256"  
215       }  
216     ],  
217     "name": "safeTransferFrom",  
218     "outputs": [],  
219     "payable": false,  
220     "stateMutability": "nonpayable",  
221     "type": "function"  
222   },  
223   {  
224     "constant": true,  
225     "inputs": [  
226       {  
227         "name": "tokenId",  
228         "type": "uint256"
```

```
229     }
230   ],
231   "name": "ownerOf",
232   "outputs": [
233     {
234       "name": "",
235       "type": "address"
236     }
237   ],
238   "payable": false,
239   "stateMutability": "view",
240   "type": "function"
241 },
242 {
243   "constant": true,
244   "inputs": [
245     {
246       "name": "owner",
247       "type": "address"
248     }
249   ],
250   "name": "balanceOf",
251   "outputs": [
252     {
253       "name": "",
254       "type": "uint256"
255     }
256   ],
257   "payable": false,
258   "stateMutability": "view",
259   "type": "function"
260 },
261 {
262   "constant": false,
263   "inputs": [
264     {
265       "name": "to",
266       "type": "address"
267     },
268     {
269       "name": "approved",
270       "type": "bool"
271     }
272   ],
273   "name": "setApprovalForAll",
274   "outputs": [],
275   "payable": false,
276   "stateMutability": "nonpayable",
277   "type": "function"
278 },
279 {
280   "constant": false,
281   "inputs": [
282     {
```

```
283         "name": "from",
284         "type": "address"
285     },
286     {
287         "name": "to",
288         "type": "address"
289     },
290     {
291         "name": "tokenId",
292         "type": "uint256"
293     },
294     {
295         "name": "_data",
296         "type": "bytes"
297     }
298 ],
299 "name": "safeTransferFrom",
300 "outputs": [],
301 "payable": false,
302 "stateMutability": "nonpayable",
303 "type": "function"
304 },
305 {
306     "constant": true,
307     "inputs": [
308         {
309             "name": "owner",
310             "type": "address"
311         },
312         {
313             "name": "operator",
314             "type": "address"
315         }
316     ],
317     "name": "isApprovedForAll",
318     "outputs": [
319         {
320             "name": "",
321             "type": "bool"
322         }
323     ],
324     "payable": false,
325     "stateMutability": "view",
326     "type": "function"
327 },
328 {
329     "anonymous": false,
330     "inputs": [
331         {
332             "indexed": true,
333             "name": "from",
334             "type": "address"
335         },
336         {
```

```
337         "indexed": true,  
338         "name": "to",  
339         "type": "address"  
340     },  
341     {  
342         "indexed": true,  
343         "name": "tokenId",  
344         "type": "uint256"  
345     }  
346 ],  
347 "name": "Transfer",  
348 "type": "event"  
349 },  
350 {  
351     "anonymous": false,  
352     "inputs": [  
353         {  
354             "indexed": true,  
355             "name": "owner",  
356             "type": "address"  
357         },  
358         {  
359             "indexed": true,  
360             "name": "approved",  
361             "type": "address"  
362         },  
363         {  
364             "indexed": true,  
365             "name": "tokenId",  
366             "type": "uint256"  
367         }  
368     ],  
369     "name": "Approval",  
370     "type": "event"  
371 },  
372 {  
373     "anonymous": false,  
374     "inputs": [  
375         {  
376             "indexed": true,  
377             "name": "owner",  
378             "type": "address"  
379         },  
380         {  
381             "indexed": true,  
382             "name": "operator",  
383             "type": "address"  
384         },  
385         {  
386             "indexed": false,  
387             "name": "approved",  
388             "type": "bool"  
389         }  
390     ],
```



```
391     "name": "ApprovalForAll",
392     "type": "event"
393 },
394 {
395     "constant": false,
396     "inputs": [
397         {
398             "name": "_name",
399             "type": "string"
400         },
401         {
402             "name": "_ra",
403             "type": "string"
404         },
405         {
406             "name": "_dec",
407             "type": "string"
408         },
409         {
410             "name": "_mag",
411             "type": "string"
412         },
413         {
414             "name": "_story",
415             "type": "string"
416         },
417         {
418             "name": "_tokenId",
419             "type": "uint256"
420         }
421     ],
422     "name": "createStar",
423     "outputs": [],
424     "payable": false,
425     "stateMutability": "nonpayable",
426     "type": "function"
427 },
428 {
429     "constant": false,
430     "inputs": [
431         {
432             "name": "_tokenId",
433             "type": "uint256"
434         },
435         {
436             "name": "_price",
437             "type": "uint256"
438         }
439     ],
440     "name": "putStarUpForSale",
441     "outputs": [],
442     "payable": false,
443     "stateMutability": "nonpayable",
444     "type": "function"
```

```

445     },
446     {
447         "constant": false,
448         "inputs": [
449             {
450                 "name": "_tokenId",
451                 "type": "uint256"
452             }
453         ],
454         "name": "buyStar",
455         "outputs": [],
456         "payable": true,
457         "stateMutability": "payable",
458         "type": "function"
459     },
460     {
461         "constant": true,
462         "inputs": [
463             {
464                 "name": "_ra",
465                 "type": "string"
466             },
467             {
468                 "name": "_dec",
469                 "type": "string"
470             },
471             {
472                 "name": "_mag",
473                 "type": "string"
474             }
475         ],
476         "name": "checkIfStarExist",
477         "outputs": [
478             {
479                 "name": "",
480                 "type": "bool"
481             }
482         ],
483         "payable": false,
484         "stateMutability": "view",
485         "type": "function"
486     }
487 ]
488 )
489 // Grab the contract at specified deployed address with the interface
490
491 // Rinkeby
492 //var starNotary = StarNotary.at('0xe757C17BDC5A0decC59eb31F3e1Fa29C2
493
494 // ganache
495 var starNotary = StarNotary.at('0xca10a17423c486b3f4f87b064cf622b9242
496
497 function claimButtonClicked () {
498     web3.eth.getAccounts(function(error, accounts) {

```

```

499     if (error) {
500         alert(error)
501         return
502     }
503
504     var account = accounts[0]
505
506     var name = document.getElementById('star-name').value
507     var ra = document.getElementById('star-ra').value
508     var dec = document.getElementById('star-dec').value
509     var mag = document.getElementById('star-mag').value
510     var story = document.getElementById('star-story').value
511
512     var tokenId = document.getElementById("star-tokenId").value;

```

AWESOME

Good! Now we have the tokenId as additional input!

```

513
514     starNotary.createStar.sendTransaction(name, ra, dec, mag, story,
515         from: account
516     ), function(error, result) {
517         if (!error) {
518             alert(result)
519             var starClaimedEvent = starNotary.Transfer()
520             starClaimedEvent.watch(function(error, result) {
521                 if (!error) {
522                     alert('star claimed')
523                 } else {
524                     alert('watching if error on star claim')
525                 }
526             })
527         } else {
528             alert(error)
529         }
530     })
531 })
532 }
533
534 function infoButtonClicked () {
535     web3.eth.getAccounts(function(error) {
536         if (error) {
537             alert(error)
538             return
539         }
540
541         var tokenId = document.getElementById("star-tokenId").value;
542
543         starNotary.tokenIdToStarInfo.call(tokenId, function(error, result

```

AWESOME

Works as expected!

```
544         if (!error) {  
545             alert(result)  
546             document.getElementById('result').innerHTML = result  
547         } else {  
548             alert(error)  
549         }  
550     })  
551 })  
552 }  
553 </script>  
554 </body>  
555 </html>
```

- ▶ smart_contracts/test/StarNotaryTest.js 3
- ▶ README.md 1
- ▶ style.css
- ▶ smart_contracts/truffle-config.js
- ▶ smart_contracts/migrations/2_deploy_contracts.js
- ▶ smart_contracts/migrations/1_initial_migration.js
- ▶ notes.txt

RETURN TO PATH

Rate this review

