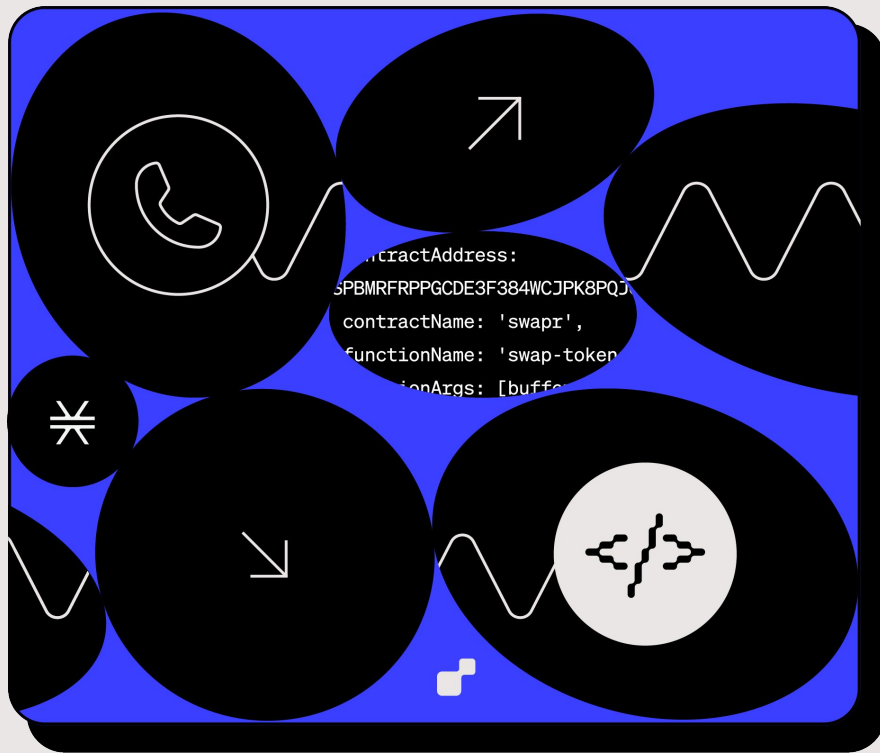


Hiro Monthly Developer Calls

#1 – April 28th, 2022





Purpose

We are
**developer
obsessed**

Strive to
engage

with the developer
community directly

Invite and
inspire
developers of
diverse backgrounds
to Stacks & Hiro

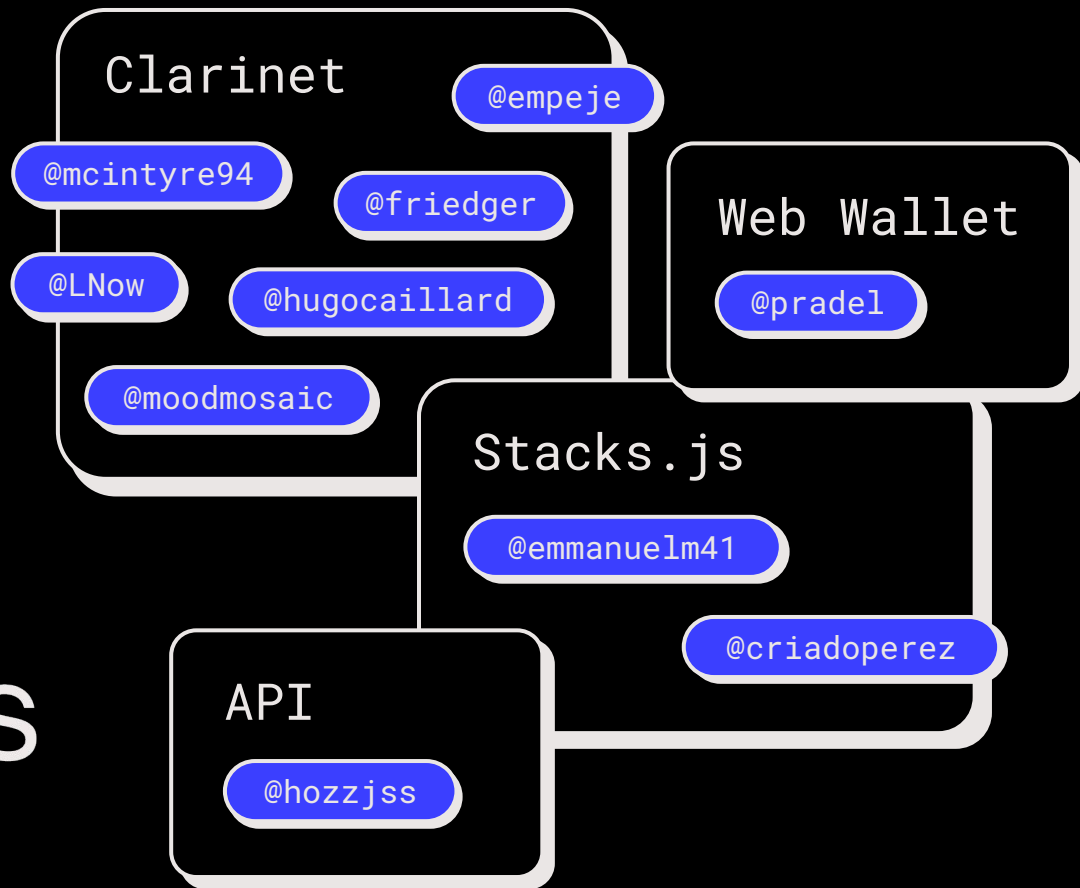
Listen
to your needs
carefully & improve
the DevX



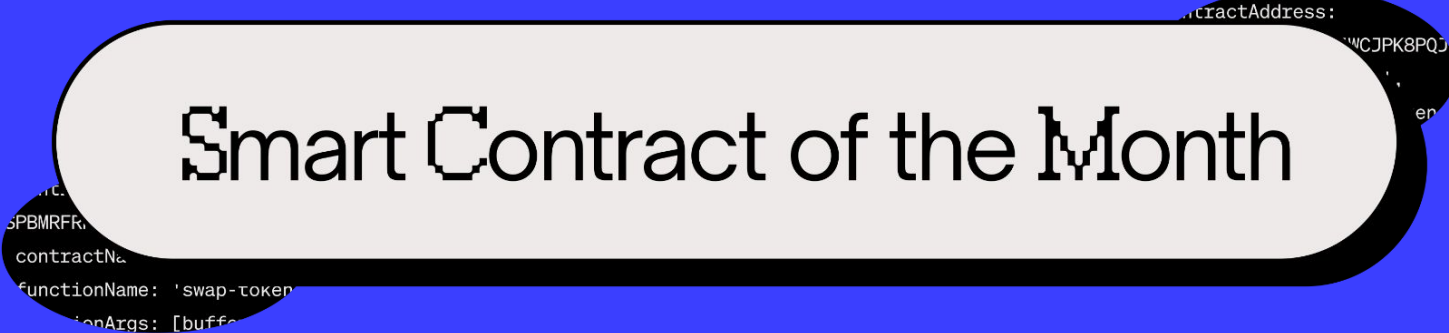
Shoutouts & Thanks

April 2022 Contributors

Thank you!



Smart Contract of the Month



Smart Contract of the Month

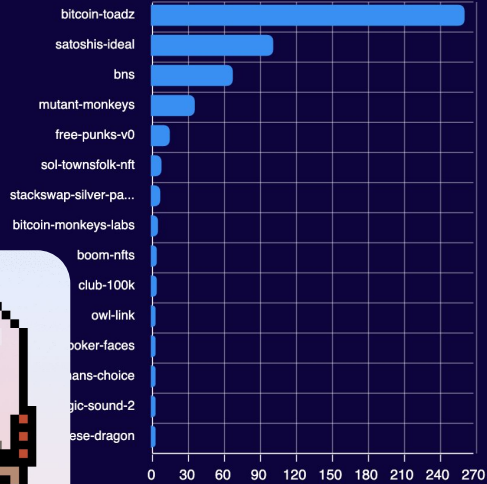


Bitcoin Toadz

- A derivative project of CrypToadz built on Stacks, using Clarity smart contracts
- Adheres to the SIP-009 NFT standard defined as a Clarity trait
- 6969 total available
- Mint price = 1 STX. Mint opened on April 21st
- Design based on the original CrypToadz with modified traits and new backgrounds. Each NFT is unique.
- After the end of the sale, a 3D collection is planned under a CC0 license for free use within the ecosystem.
- Royalties are set to 5% of the secondary sales.

⇒ [Link to smart contract in Stacks Explorer](#)

Top NFTs by # mints last 100 Blocks



Bitcoin Toadz

Bitcoin Toadz #1169

6969 out of 6969

Rarity Score: 116

ATTRIBUTE	FLOOR	RARITY SCORE	RARITY %
Background	15	+100	5.88 %
Body	15	+200	4.88 %
Head	10	+30.75	2.58 %
Mask	15	+21.85	4.02 %
Eyes	15	+10.0	9.33 %
Tail	10	+100	100.00 %

15.00

Buy Item

Place Bid

0.00000000

View PPS Image

@BitcoinToadz

bitcointoadz.com

BITCOIN TOADZ SMART CONTRACT

```

900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200

```

```

1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500

```

f

30 functions

⊗

28 variables

=x

4 maps

🔗

1 token






Functions Requiring a Transaction to Call

 burn	→
 buy-in-ustx	→
 claim	→
 claim-five	→
 claim-ten	→
 claim-three	→
 freeze-metadata	→
 list-in-ustx	→
 set-artist-address	→



N SERIES
TIDYING UP
WITH MARIE KONDO

 set-base-uri	→
 set-mint-limit	→
 set-price	→
 toggle-pause	→
 transfer	→
 unlist-in-ustx	→

Read-Only Functions

 get-balance		→
 get-last-token-id		→
 get-listing-in-ustx		→
 get-mint-limit		→
 get-mints		→
 get-owner		→
 get-paused		→
 get-price		→
 get-token-uri		→

Implementing NFT Trait and defining constants

```
;; bitcoin-toadz

(impl-trait 'SP2PABAF9FTAJYNFZH93XENAJ8FVY99RRM50D2JG9.nft-trait.nft-trait)

(define-non-fungible-token bitcoin-toadz uint)

;; Constants
(define-constant DEPLOYER tx-sender)
(define-constant COMM u1000)
(define-constant COMM-ADDR 'SPNWZ5V2TPWQGQVDR6T7B6RQ4XMGZ4PXTEE0VQ0S)

(define-constant ERR-NO-MORE-NFTS u100)
(define-constant ERR-NOT-ENOUGH-PASSES u101)
(define-constant ERR-PUBLIC-SALE-DISABLED u102)
(define-constant ERR-CONTRACT-INITIALIZED u103)
(define-constant ERR-NOT-AUTHORIZED u104)
(define-constant ERR-INVALID-USER u105)
(define-constant ERR-LISTING u106)
(define-constant ERR-WRONG-COMMISSION u107)
(define-constant ERR-NOT-FOUND u108)
(define-constant ERR-PAUSED u109)
(define-constant ERR-MINT-LIMIT u110)
(define-constant ERR-METADATA-FROZEN u111)
(define-constant ERR-AIRDROP-CALLED u112)
(define-constant ERR-NO-MORE-MINTS u113)
```

NFT Name

SIP-009 NFT Trait

ERROR CODES

Defining Internal Variables and Maps

```
;; Internal variables
(define-data-var mint-limit uint u6969)
(define-data-var last-id uint u1)
(define-data-var total-price uint u1000000)
(define-data-var artist-address principal 'SPJJYJVZ4H7B34GG8D3SSN70WVWDYSHCC9E9ZV4V)
(define-data-var ipfs-root (string-ascii 80) "ipfs://ipfs/QmRt5hZo26w1VPP5gUTrnkxjef2XbbLETdx")
(define-data-var mint-paused bool false)
(define-data-var premint-enabled bool false)
(define-data-var sale-enabled bool false)
(define-data-var metadata-frozen bool false)
(define-data-var airdrop-called bool false)
(define-data-var mint-cap uint u500)

(define-map mints-per-user principal uint)
(define-map mint-passes principal uint)
```

Artist's STX Address

IPFS Hash

Uses post conditions to ensure correct mint.

Post conditions



SP1MKMHN...TGPySTKW
Transfers no more than 3 STX

Defining the Minting Functions

```
(define-public (claim)
  (mint (list true)))

(define-public (claim-three) (mint (list true true true)))

(define-public (claim-five) (mint (list true true true true true)))

(define-public (claim-ten) (mint (list true true true true true true true true true true)))
```

Default Minting Function

```
;; Default Minting
(define-private (mint (orders (list 25 bool)))
  (mint-many orders))

(define-private (mint-many (orders (list 25 bool)))
  (let
    (
      (last-nft-id (var-get last-id))
      (enabled (asserts! (<= last-nft-id (var-get mint-limit)) (err ERR-NO-MORE-NFTS)))
      (art-addr (var-get artist-address))
      (id-reached (fold mint-many-iter orders last-nft-id))
      (price (* (var-get total-price) (- id-reached last-nft-id)))
      (total-commission (/ (* price COMM) u10000))
      (current-balance (get-balance tx-sender))
      (total-artist (- price total-commission))
      (capped (> (var-get mint-cap) u0))
      (user-mints (get-mints tx-sender))
    )
    (asserts! (or (is-eq false (var-get mint-paused)) (is-eq tx-sender DEPLOYER)) (err ERR-PAUSED))
    (asserts! (or (not capped) (is-eq tx-sender DEPLOYER) (is-eq tx-sender art-addr) (>= (var-get mint-cap) (+ (len orders)
    (map-set mints-per-user tx-sender (+ (len orders) user-mints))
    (if (or (is-eq tx-sender art-addr) (is-eq tx-sender DEPLOYER) (is-eq (var-get total-price) u00000000))
      (begin
        (var-set last-id id-reached)
        (map-set token-count tx-sender (+ current-balance (- id-reached last-nft-id)))
      )
      (begin
        (var-set last-id id-reached)
        (map-set token-count tx-sender (+ current-balance (- id-reached last-nft-id)))
        (try! (stx-transfer? total-artist tx-sender (var-get artist-address)))
        (try! (stx-transfer? total-commission tx-sender COMM-ADDR))
      )
    )
    (ok id-reached)))
```

Default Minting Function

An incrementing variable to track mint progress ⇒

Setting the address of the artist ⇒

Set price for minting NFT ⇒

Ability to pause minting ⇒

Setting upper limit of mint quantity ⇒

Ability to burn a specific NFT ⇒

Check if a principal is the owner ⇒

```
(define-private (mint-many-iter (ignore bool) (next-id uint))
  (if (<= next-id (var-get mint-limit))
    (begin
      (unwrap! (nft-mint? bitcoin-toadz next-id tx-sender) next-id)
      (+ next-id u1)
    )
    next-id))

(define-public (set-artist-address (address principal))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set artist-address address))))

(define-public (set-price (price uint))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set total-price price))))

(define-public (toggle-pause)
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (ok (var-set mint-paused (not (var-get mint-paused))))))

(define-public (set-mint-limit (limit uint))
  (begin
    (asserts! (or (is-eq tx-sender (var-get artist-address)) (is-eq tx-sender DEPLOYER)) (err ERR-INVALID-USER))
    (asserts! (< limit (var-get mint-limit)) (err ERR-MINT-LIMIT))
    (ok (var-set mint-limit limit))))

(define-public (burn (token-id uint))
  (begin
    (asserts! (is-owner token-id tx-sender) (err ERR-NOT-AUTHORIZED))
    (nft-burn? bitcoin-toadz token-id tx-sender)))

(define-private (is-owner (token-id uint) (user principal))
  (is-eq user (unwrap! (nft-get-owner? bitcoin-toadz token-id) false)))
```

More Functions!

◆ SIP-009 compliant **transfer** function ⇒
Sends NFT from one principal to another

SIP-009 Functions!

◆ Check owner of the contract ⇒

◆ Get the ID of the last token minted ⇒

◆ Get the metadata URI of the NFT ⇒

Check if the mint has been paused ⇒

Check the price of minting an NFT ⇒

Check the # of mints of a specific principal ⇒

Check the quantity limit of the mint ⇒

```
;; Non-custodial SIP-009 transfer function
(define-public (transfer (id uint) (sender principal) (recipient principal))
  (begin
    (asserts! (is-eq tx-sender sender) (err ERR-NOT-AUTHORIZED))
    (asserts! (is-none (map-get? market id)) (err ERR-LISTING))
    (trnsfr id sender recipient)))

;; read-only functions
(define-read-only (get-owner (token-id uint))
  (ok (nft-get-owner? bitcoin-toadz token-id)))

(define-read-only (get-last-token-id)
  (ok (- (var-get last-id) u1)))

(define-read-only (get-token-uri (token-id uint))
  (ok (some (concat (concat (var-get ipfs-root) "{id}") ".json"))))

(define-read-only (get-paused)
  (ok (var-get mint-paused)))

(define-read-only (get-price)
  (ok (var-get total-price)))

(define-read-only (get-mints (caller principal))
  (default-to u0 (map-get? mints-per-user caller)))

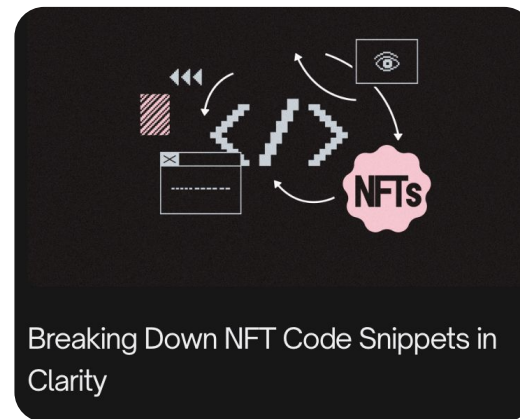
(define-read-only (get-mint-limit)
  (ok (var-get mint-limit)))
```

Learn
more
about
NFTs on
Stacks!



Bitcoin NFTs Explained: What They Are,
How They Work, and Why You Should
Build Them

<https://www.hiro.so/blog/bitcoin-nfts-explained-what-the-y-are-how-they-work-and-why-you-should-build-them>



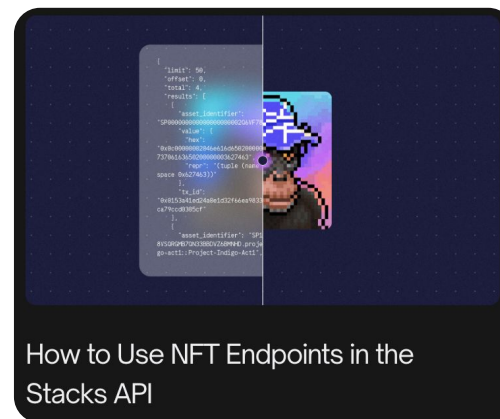
Breaking Down NFT Code Snippets in
Clarity

<https://www.hiro.so/blog/breaking-down-nft-code-snippets-in-clarity>



How to Use NFT Endpoints in
the Stacks API

<https://www.youtube.com/watch?v=RLdhlGwu7kA>



How to Use NFT Endpoints in the
Stacks API

<https://www.hiro.so/blog/how-to-use-nft-endpoints-in-the-stacks-api>

Community Topics



Issue: Community topic

Suggest a topic for Hiro Monthly Developer Meeting. If this doesn't look right, [choose a different type](#).



Write

Preview

H B I

****Topic:****

<!-- (remove this commented block after reading)

Example Topic: I am using Clarinet, and I want to know the best practices on how to use it in coordination with CI/CD pipelines

-->

****Why this topic?****

<!-- (remove this commented block after reading)

Example: I want to learn and follow the best practices to rely on the production hardened methods when it comes to CI/CD deployments.

-->

Attach files by dragging & dropping, selecting or pasting them.



Styling with Markdown is supported

Submit new issue

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).



Q1 2022 Product Releases



Q1 2022 Releases

API

- ✓ Faster APIs
- ✓ Faster Event Replays
- ✓ New NFT Endpoints
- ✓ Improved Docs & Tests

Stacks.js

- ✓ Reduced Bundle Size
- ✓ Cross-browser Compatibility
- ✓ Release Automations

Clarinet

- ✓ [New In-line Debugger!](#)
- ✓ Improved Check-checker
- ✓ Improved Error Messages

Blockchain

- ✓ Improved Fee Estimates
- ✓ [Introduced Hyperchains!](#)

Wallet

- ✓ Fiat Onramps (Moonpay & Transak)
- ✓ Improved Fee Estimates
- ✓ Improved Onboarding Experience

Explorer

- ✓ Improved Performance



Q2 2022 Product Focus



Q2 2022 Product Focus

API

- 99.99% Uptime
- Optimize Event Replays
- Optimize CPU Utilization
- Functional Testing

Stacks.js

- Stacks.js Starter Templates
- Improve Documentation

Clarinet

- VS Code: In-line Debugger
- Optimize `clarinet integrate`

Blockchain

- [Hyperchains MVP Launch!](#)

Wallet

- Improved fiat onramp UX
- Arbitrary message signing
- Ledger support for extension
- Improved BNS support
- Initial BTC support
- Hyperchains UX research
- Improved support UX and FAQs

Explorer

- Resolve performance issues and bugs

Are you looking for
the roadmap,
timelines, open
feature requests,
or report problems?



github.com/hirosystems

hirosystems / stacks-blockchain-api

< > Code Issues 67 Pull requests 22

Labels Milestones

Q2-2022

Due by May 31, 2022 0% complete

Q2-2022

5 Open 0 Closed

- ☐ ☒ Optimize event replay process Q2-2022
#1082 opened on Feb 25 by rafaelcr 4 of 8 tasks
- ☐ ☒ add arbitrary contract call support to ro
#1095 opened on Mar 8 by rafaelcr
- ☐ ☒ Introduce an API Gateway to access our
#1123 opened on Mar 24 by saralab
- ☐ ☒ Implement automated functional testing
#1002 opened on Jan 25 by saralab
- ☐ ☒ Add blockchain stats endpoints Q2-2022
#1130 opened on Mar 28 by rafaelcr 2 tasks

hirosystems / stacks.js

< > Code Issues 18 Pull requests

Labels Milestones

Q2-2022

Due by May 31, 2022 0% complete

Q2 2022

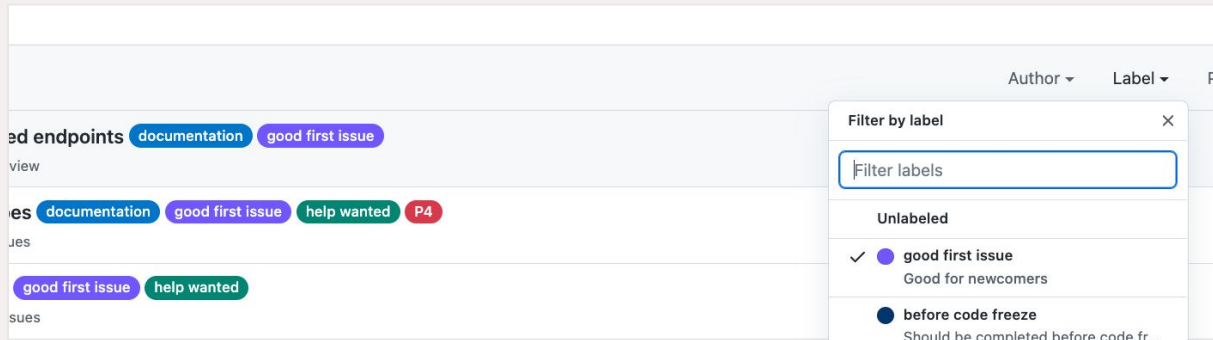
5 Open 0 Closed

- ☐ ☒ Offload bn.js ✓
#1150 opened on Dec 16, 2021 by ah
- ☐ ☒ Offload bip* dependency
#1198 opened on Feb 23 by janniks
- ☐ ☒ [wallet-sdk] Offload bip39
#1202 opened on Feb 28 by janniks
- ☐ ☒ [Bundle Size][Wallet-SDK]
#1237 opened 7 days ago by ahsan-
- ☐ ☒ Create stacks.js starter t
#1123 opened on Oct 20, 2021 by ky

New to Hiro, and are looking for ways to contribute?



github.com/hirosystems



Are you wondering
how to engage with
Hiro or ask
questions?

