READY LAYER 2:

BITCOIN BUILDER COMPETITION

Bitcoin x AI Working Group

Where silicon intelligence meets internet money.

Bitcoin is the category winner

- AI is growing at an exponential rate
- What if we give AI agents a Bitcoin wallet?
- What is the ideal Bitcoin tool set for AI agents?

Building the Bitcoin agent tools

ai-agent-crew Public

Langchain + CrewAI powered AI agents with Bitcoin wallets.

Python 8 2

smart-contracts Public

Stacks smart contracts, test suite, and deployment plans.

TypeScript 1

agent-tools-ts Public

TS scripts for interacting with the Stacks blockchain, powered by Bun and Stacks.js.

TypeScript 1 2

protocol-docs Public

Documentation for using CrewAI AI agents with a Bitcoin wallet.

1

gated-402-api Public

Example API implementation with HTTP 402 responses for unpaid resources.

TypeScript 1

landing-page Public

Main landing page for https://aibtc.dev

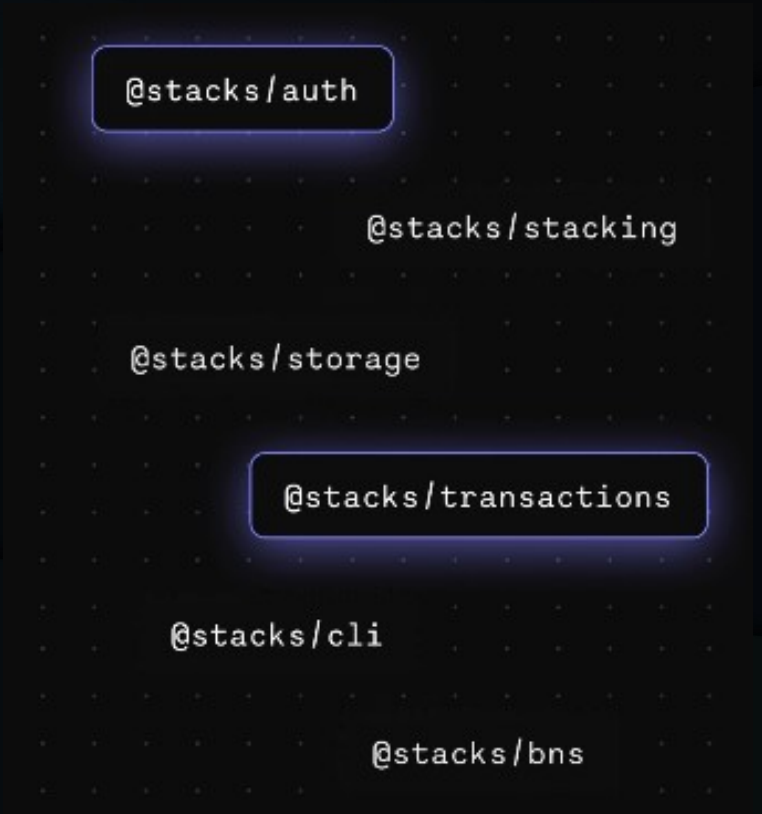
TypeScript

CrewAI powered AI agents

- AI agent framework based on Langchain
- Supports any model through OpenAI-compatible API
- Simplifies task creation, agent automation, and more

Stacks.js tooling for wallet interactions

- Create, access, and manage Bitcoin + Stacks wallets
- Send, receive, and manage on-chain assets
- Create, broadcast, and query transactions



Available tooling as of today

- Load wallet and account statuses
- Get transaction info or status by TXID
- Sign structured data per SIP-018 standard
- Get aiBTC from testnet faucet
- Get resource, invoice, or user data
- Get recent payments
- Pay resource invoice

Smart contracts for on-chain actions

- Clarity is a simple, safe, and secure language
- Clarinet SDK provides testing and integration
- First primitive: service provider with resources

```
;; tracks resource indexes by resource name
(define-map ResourceIndexes
  (string-utf8 50) ;; resource name
  uint             ;; resource index
)

;; tracks resources added by deployer keyed by resource index
;; can iterate over full map with resourceCount data-var
(define-map ResourceData
  uint ;; resource index
  {
    createdAt: uint,
    enabled: bool,
    name: (string-utf8 50),
    description: (string-utf8 255),
    price: uint,
    totalSpent: uint,
    totalUsed: uint,
  }
)

;; tracks invoices paid by users requesting access to a resource
(define-map InvoiceData
  uint ;; invoice count
  {
    amount: uint,
    createdAt: uint,
    userIndex: uint,
    resourceName: (string-utf8 50),
    resourceIndex: uint,
  }
)
```

clarity

Real-world integration example

- Gated API – you only get access if you've paid for the resource
- Demonstrates blend of on-chain resources and applications


```
{"paid":false,"status":"No payment data found for
ST35K818S3K2GSNEBC3M35GA3W8Q7X72KF4RVM3QA.","paymentInfo":
{"contractName":"stacks-m2m-
v2","contractAddress":"ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TDGBZQ3EKR",
"functionName":"pay-invoice-by-resource-name","functionArgs":
["bitcoin-face","string-utf8
50","ST35K818S3K2GSNEBC3M35GA3W8Q7X72KF4RVM3QA","principal"]}}
```


Message signing as identity

- Digital signatures for off-chain verification
- Know WHAT address is requesting
- Replaces traditional authentication

```
===== ACCOUNT INFO =====
Network: testnet
Chain ID: 2147483648
Tx version: 128
Account index: 0
Account address: ST2HQ5J6RP8HSQE9KKGWCHW9P
===== SIGNATURE INFO =====
Message: ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TD
Signed message type: 10
Signed message data: 424b7c7d4faf0dc3edb5e
fdd00434dd15a1254a9ce3a60d3ed5def54e9d588a
f897c497aeb5ff4566cec18afc0052500
===== VALIDATION INFO =====
Public key from private:   024938a72851a9e
67afb215f7bf82a02232c7ed79a0f7523f
Public key from signature: 024938a72851a9e
67afb215f7bf82a02232c7ed79a0f7523f
Signature verified: true
```

Running an AI agent crew

```python
def engage_crew_with_tasks():
    # define agent
    wallet_agent = BitcoinCrew.wallet_agent()
    # define the tasks
    task_1 = Task(
        description="What are the wallet addresses you have access to?",
        agent=wallet_agent,
    )
    task_2 = Task(
        description="What is the fourth wallet address you have access to?",
        agent=wallet_agent,
    )
    task_3 = Task(
        description="Get information about the configured wallet.",
        agent=wallet_agent,
    )
    # create a crew
    wallet_crew = Crew(
        agents=[wallet_agent],
        process=Process.sequential,
        tasks=[task_1, task_2, task_3],
        verbose=True,
    )
    # run the crew
    wallet_result = wallet_crew.kickoff()
```

```
> Entering new CrewAgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Get Wallet Addresses
Action Input: None{'output': '0: ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TDGBZQ3EKR\n1:
n4: ST388ZPC7RVQ23548NWK543AYYP2A2AB558D3CD7S\n5: ST2G192AS4P9TW1BXFKSQX18R4RH
3BSSDDMZTD959SY\n9: ST3NRRJS9BNEDJN9WG7CNVQ247N9DHFVSQSWF9K14\n10: ST3F37HCN7G
Final Answer: Here are the addresses of the configured wallet:

1. ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TDGBZQ3EKR
2. ST3GEF4KYM4V41FHC9NX0F7K0GW1VC6A4WPXNYQKS
3. ST1TZE9ZY61FYR7YM9BR0543XKX9YG5TR9017R4WJ
4. ST35K818S3K2GSNEBC3M35GA3W8Q7X72KF4RVM3QA
5. ST388ZPC7RVQ23548NWK543AYYP2A2AB558D3CD7S
6. ST2G192AS4P9TW1BXFKSQX18R4RH1FHYRP2PWHW24
7. ST6CWNRQWF468S6A56Q995WVY7F6X43GFV7H16N2
8. ST1TP7192SZ2SRAFT0JY0V91EYJBBH8C6KB7F3QPR
9. ST129ABK9RK7Y2MCMWR48KPP2S3BSSDDMZTD959SY
10. ST3NRRJS9BNEDJN9WG7CNVQ247N9DHFVSQSWF9K14
11. ST3F37HCN7G2CXFVZRT36M57SWKG77P2BSE9ET80W
```

```
> Entering new CrewAgentExecutor chain...
Thought: Do I need to use a tool? Yes
Action: Get Wallet Status
Action Input: None{'output': 'Account index: 0\nAccount address: ST2HQ5J6RP8HSQE9KKGW
ing the address ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TDGBZQ3EKR with a nonce of 10. The acc
Do I need to use a tool? No
Final Answer: The configured wallet is using the address ST2HQ5J6RP8HSQE9KKGWCHW9PT9

> Finished chain.
[DEBUG]: [Wallet] Task output: The configured wallet is using the address ST2HQ5J6RP8


------------------------------------------------
Wallet Crew Result:
The configured wallet is using the address ST2HQ5J6RP8HSQE9KKGWCHW9PT9SVE4TDGBZQ3EKR.
------------------------------------------------
```

What we’d love to see

- A marketplace for AI agent-related services
- Expanded AI agent tooling for new abilities
- AI agents that complete complex on-chain tasks

Let’s build together!

- Main Website

  https://aibtc.dev

- GitHub Organization

  https://github.com/aibtcdev