
Meshblu

Meshblu is a cross-protocol IoT machine-to-machine instant messaging system. It is the core messaging system for Citrix's Octoblu IoT platform.

Supported Protocols: HTTP, Socket.io, Websocket, MQTT, CoAP, AMQP, and XMPP.

Version 2.0

We have completely re-written Meshblu into many small components or micro-services. This Meshblu 1.0 repository is being preserved for historical reference.

All of the new Meshblu components are prefixed with `meshblu-core`. See a list [here](#).

Meshblu is dependent on `node.js`, `redis`, `mongodb`, and either `npm` or `yarn`.

Production

In order to run a barebones `meshblu-core` cluster, you'll need the following repositories.

1. `meshblu-core-dispatcher`
2. `meshblu-core-worker-webhook`
3. `meshblu-core-protocol-adapter-http`

All `meshblu-core` services and workers have a `Dockerfile`.

A production Meshblu cluster will consist of many services and workers. We currently don't have documentation for running a complex cluster but we are working on it.

Development

For development use, you can run the bundled barebones cluster:

Installation

```
1 git clone https://github.com/octoblu/meshblu
2 cd meshblu
3 npm install
```

See Usage

```
1 node command.js --help
```

Basic Example w/ `env`

```
1 #!/bin/bash
2
3 # For development usage only
4
5 env \
6   PRIVATE_KEY_BASE64="..." \
7   PUBLIC_KEY_BASE64="..." \
8   PEPPER='some-random-string' \
9   MESHBLU_HTTP_PORT='3000' \
10  node command.js
```

See `./test-start.sh`

Basic Example w/ `args`

```
1 #!/bin/bash
2
3 # For development usage only
4
5 node command.js \
6   --private-key-base64 '...' \
7   --public-key-base64 '...' \
8   --pepper 'some-other-random-string' \
9   --meshblu-http-port 3000
```

Debug Mode It is normal not see any debug output by default. If you want to see debug output, use the environment `DEBUG=*`, or something more specific, like `DEBUG=meshblu*`.

It's Alive! To verify that Meshblu 2.0 is alive and well, open `http://localhost:3000/status` in a web browser or open a new terminal tab and run:

```
1 curl http://localhost:3000/status
```

You should see Meshblu 2.0 respond with:

```
1 {"meshblu":"online"}
```

You can register a new IoT device by running:

```
1 curl -X POST http://localhost:3000/devices
```

You should see Meshblu 2.0 respond with an authentication UUID and Token as well as the device's security whitelist settings like this:

```
1 { online: false,
2   discoverWhitelist: [ '*' ],
```

```
3  configureWhitelist: [ '*' ],
4  sendWhitelist: [ '*' ],
5  receiveWhitelist: [ '*' ],
6  uuid: 'b112c941-7973-4e2b-8dbe-b7bba27ae199',
7  meshblu:
8    { createdAt: '2016-11-15T16:07:07.801Z',
9      hash: 'Dy5NlIlmygrnrhp0Cln+zb77nHLYdobb+HwbRVzWdPs=' },
10 _id: '582b32ab67899618f48c2e1b',
11 token: 'd5bcf1a57f4ccefa0ecd672c7090e7949cc8244' }
```

Meshblu API Documentation Checkout our developer docs for more information on our HTTP REST API as well as documentation for all other protocol APIs, security whitelists and blacklists, connectors, data forwarders, and overall Meshblu architecture.

Introducing the Meshblu CLI We have a convenient command line interface for simplifying the interaction with the Meshblu API.

```
1 # Install the meshblu cli utility
2 npm install --global meshblu-util
3 # Register a device
4 meshblu-util register -U http://localhost:3000 > meshblu.json
5 # Fetch the device
6 meshblu-util get
7 # Update the device
8 meshblu-util update -d '{"type": "some-device"}'
9 # Fetch the updated device
10 meshblu-util get
```

List of meshblu-core components

Workers

1. meshblu-core-dispatcher
2. meshblu-core-worker-webhook

Protocol Adapters

1. meshblu-core-protocol-adapter-socket.io
2. meshblu-core-protocol-adapter-http
3. meshblu-core-protocol-adapter-xmpp
4. meshblu-core-protocol-adapter-coap

-
5. meshblu-core-protocol-adapter-mqtt
 6. meshblu-core-protocol-adapter-http-streaming

Firehoses

1. meshblu-core-worker-firehose-amqp
2. meshblu-core-firehose-socket.io

Balancers

1. meshblu-haproxy
2. meshblu-balancer-http-streaming
3. meshblu-balancer-firehose-socket.io
4. meshblu-balancer-xmpp
5. meshblu-balancer-websocket
6. meshblu-balancer-mqtt
7. meshblu-balancer-coap
8. meshblu-balancer-socket.io

Core Datastores

1. meshblu-core-datastore
2. meshblu-core-cache

Core Managers

1. meshblu-core-manager-token
2. meshblu-core-manager-device
3. meshblu-core-manager-hydrant
4. meshblu-core-manager-whitelist
5. meshblu-core-manager-webhook
6. meshblu-core-manager-subscription
7. meshblu-core-manager-root-token
8. meshblu-core-manager-messenger

Core Tasks

1. meshblu-core-task-black-list-token
2. meshblu-core-task-check-broadcast-received-whitelist
3. meshblu-core-task-check-broadcast-sent-whitelist
4. meshblu-core-task-check-configure-as-whitelist
5. meshblu-core-task-check-configure-whitelist
6. meshblu-core-task-check-discover-as-whitelist
7. meshblu-core-task-check-discover-whitelist
8. meshblu-core-task-check-discoveras-whitelist
9. meshblu-core-task-check-forwarded-for
10. meshblu-core-task-check-receive-as-whitelist
11. meshblu-core-task-check-receive-whitelist
12. meshblu-core-task-check-root-token
13. meshblu-core-task-check-send-as-whitelist
14. meshblu-core-task-check-send-whitelist
15. meshblu-core-task-check-token
16. meshblu-core-task-check-token-black-list
17. meshblu-core-task-check-token-cache
18. meshblu-core-task-check-update-device-is-valid
19. meshblu-core-task-check-whitelist-broadcast-as
20. meshblu-core-task-check-whitelist-broadcast-received
21. meshblu-core-task-check-whitelist-broadcast-sent
22. meshblu-core-task-check-whitelist-configure-as
23. meshblu-core-task-check-whitelist-configure-received
24. meshblu-core-task-check-whitelist-configure-sent
25. meshblu-core-task-check-whitelist-configure-update
26. meshblu-core-task-check-whitelist-discover-as
27. meshblu-core-task-check-whitelist-discover-view
28. meshblu-core-task-check-whitelist-message-as
29. meshblu-core-task-check-whitelist-message-from
30. meshblu-core-task-check-whitelist-message-received
31. meshblu-core-task-check-whitelist-message-sent
32. meshblu-core-task-create-session-token
33. meshblu-core-task-create-subscription
34. meshblu-core-task-deliver-webhook
35. meshblu-core-task-enforce-message-rate-limit
36. meshblu-core-task-enqueue-deprecated-webhooks

-
37. meshblu-core-task-enqueue-jobs-for-forward-broadcast-received
 38. meshblu-core-task-enqueue-jobs-for-forward-configure-received
 39. meshblu-core-task-enqueue-jobs-for-forward-unregister-received
 40. meshblu-core-task-enqueue-jobs-for-subscriptions-broadcast-received
 41. meshblu-core-task-enqueue-jobs-for-subscriptions-broadcast-sent
 42. meshblu-core-task-enqueue-jobs-for-subscriptions-configure-received
 43. meshblu-core-task-enqueue-jobs-for-subscriptions-configure-sent
 44. meshblu-core-task-enqueue-jobs-for-subscriptions-message-received
 45. meshblu-core-task-enqueue-jobs-for-subscriptions-message-sent
 46. meshblu-core-task-enqueue-jobs-for-subscriptions-unregister-received
 47. meshblu-core-task-enqueue-jobs-for-subscriptions-unregister-sent
 48. meshblu-core-task-enqueue-jobs-for-webhooks-broadcast-received
 49. meshblu-core-task-enqueue-jobs-for-webhooks-broadcast-sent
 50. meshblu-core-task-enqueue-jobs-for-webhooks-configure-received
 51. meshblu-core-task-enqueue-jobs-for-webhooks-configure-sent
 52. meshblu-core-task-enqueue-jobs-for-webhooks-message-received
 53. meshblu-core-task-enqueue-jobs-for-webhooks-message-sent
 54. meshblu-core-task-enqueue-jobs-for-webhooks-unregister-received
 55. meshblu-core-task-enqueue-jobs-for-webhooks-unregister-sent
 56. meshblu-core-task-enqueue-webhooks
 57. meshblu-core-task-find-and-update-device
 58. meshblu-core-task-forbidden
 59. meshblu-core-task-get-authorized-subscription-types
 60. meshblu-core-task-get-broadcast-subscription-types
 61. meshblu-core-task-get-device
 62. meshblu-core-task-get-device-public-key
 63. meshblu-core-task-get-global-public-key
 64. meshblu-core-task-get-status
 65. meshblu-core-task-get-subscriptions
 66. meshblu-core-task-migrate-root-token
 67. meshblu-core-task-no-content
 68. meshblu-core-task-protect-your-as
 69. meshblu-core-task-publish-broadcast-received
 70. meshblu-core-task-publish-configure-received
 71. meshblu-core-task-publish-deprecated-subscriptions
 72. meshblu-core-task-publish-message
 73. meshblu-core-task-publish-message-received
 74. meshblu-core-task-publish-subscriptions

-
75. meshblu-core-task-publish-unregister-received
 76. meshblu-core-task-register-device
 77. meshblu-core-task-reject-your-as
 78. meshblu-core-task-remove-device-cache
 79. meshblu-core-task-remove-root-session-token
 80. meshblu-core-task-remove-subscription
 81. meshblu-core-task-remove-token-cache
 82. meshblu-core-task-reset-token
 83. meshblu-core-task-revoke-all-tokens
 84. meshblu-core-task-revoke-session-token
 85. meshblu-core-task-revoke-token-by-query
 86. meshblu-core-task-search-device
 87. meshblu-core-task-search-token
 88. meshblu-core-task-send-message
 89. meshblu-core-task-send-message-2
 90. meshblu-core-task-unregister-device
 91. meshblu-core-task-update-device
 92. meshblu-core-task-update-message-rate

Clients

1. node-meshblu-socket.io
2. node-meshblu-firehose-socket.io
3. node-meshblu-http
4. node-meshblu-websocket
5. node-meshblu-mqtt
6. node-meshblu-xmpp
7. node-meshblu-amqp
8. node-meshblu-coap
9. browser-meshblu-http
10. swift-meshblu-http

Utilities

1. meshblu-util

Legacy Meshblu 1.x

[View it here](#)