

Connection Name

Connections can be assigned a name which will appear in some of the server monitoring data. This name is not required, but is **highly recommended** as a friendly connection name will help in monitoring, error reporting, debugging, and testing.

Go

```
nc, err := nats.Connect("demo.nats.io", nats.Name("API Name Option Example"))
if err != nil {
    log.Fatal(err)
}
defer nc.Close()

// Do something with the connection
```

Java

```
Options options = new Options.Builder()
    .server("nats://demo.nats.io:4222")
    .connectionName("API Name Option Example") // Set Name
    .build();
Connection nc = Nats.connect(options);

// Do something with the connection

nc.close();
```

JavaScript

```
const nc = await connect({
  name: "my-connection",
  servers: ["demo.nats.io:4222"],
});
```

Python

```
nc = NATS()
await nc.connect(
    servers=["nats://demo.nats.io:4222"],
    name="API Name Option Example")

# Do something with the connection

await nc.close()
```

C#

```
// dotnet add package NATS.Net
using NATS.Net;

await using var client = new NatsClient(name: "API Name Option Example", url:
    "nats://demo.nats.io:4222");

// It's optional to call ConnectAsync()
// as it will be called when needed automatically
await client.ConnectAsync();
```

Ruby

```
require 'nats/client'

NATS.start(servers: ["nats://demo.nats.io:4222"], name: "API Name Option Example") do |nc|
    # Do something with the connection

    # Close the connection
    nc.close
end
```

C

```
natsConnection      *conn    = NULL;
natsOptions          *opts    = NULL;
natsStatus           s        = NATS_OK;

s = natsOptions_Create(&opts);
if (s == NATS_OK)
    s = natsOptions_SetName(opts, "API Name Option Example");
```

```
if (s == NATS_OK)
    s = natsConnection_Connect(&conn, opts);

(...)

// Destroy objects that were created
natsConnection_Destroy(conn);
natsOptions_Destroy(opts);
```

```
### #1
[] cocobook-sync [] 2026-10-28 11:11:06 UTC
[] cocobook-sync [] 2026-10-28 11:11:06 UTC
```