Nock is an HTTP mocking and expectations library for Node.js

Nock can be used to test modules that perform HTTP requests in isolation.

For instance, if a module performs HTTP requests to a CouchDB server or makes HTTP requests to the Amazon API, you can test that module in isolation.

Nock works by overriding Node's http.request function. Also, it overrides http.ClientRequest too to cover for modules that use it directly.

**Install**

$ npm install nock

**Use**

On your test, you can setup your mocking object like this:

var nock = require('nock');

var couchdb = nock('http://myapp.iriscouch.com')

.get('/users/1')

.reply(200, {

\_id: '123ABC',

\_rev: '946B7D1C',

username: 'pgte',

email: 'pedro.teixeira@gmail.com'

});

This setup says that we will intercept every HTTP call to http://myapp.iriscouch.com.

It will intercept an HTTP GET request to '/users/1' and reply with a status 200, and the body will contain a user representation in JSON.

Then the test can call the module, and the module will do the HTTP requests.

**Specifying hostname**

The request hostname can be a string or a RegExp.

var scope = nock('http://www.example.com')

.get('/resource')

.reply(200, 'domain matched');

var scope = nock(/example**\.**com/)

.get('/resource')

.reply(200, 'domain regex matched');

(You can choose to include or not the protocol in the hostname matching)

**Specifying request body**

You can specify the request body to be matched as the second argument to the get, post, put or delete specifications. There are four types of second argument allowed:

**String**: nock will exact match the stringified request body with the provided string

nock('http://www.example.com')

.post('/login', 'username=pgte&password=123456')

.reply(200, { id: '123ABC' });