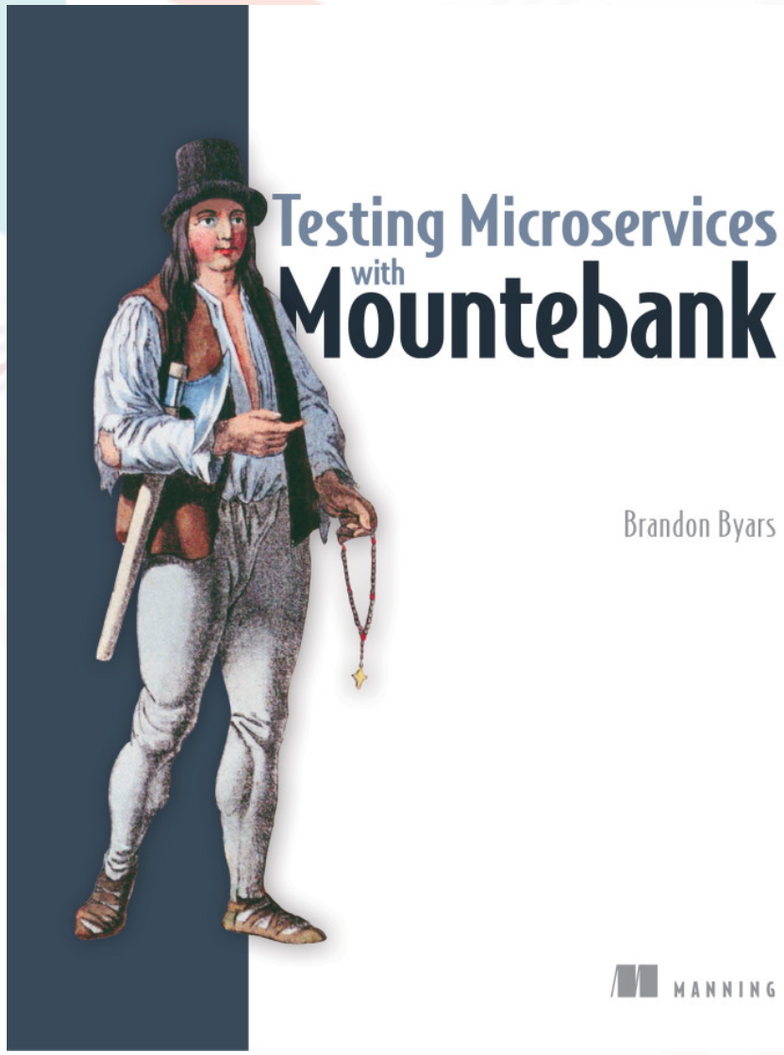
จำลองระบบเพื่อใช้ในการทดสอบ ด้วย

Mountebank 101



- Hello, World! Mountebank
- Predicates
- Record/Replay
- Behavior and Programming mountebank
- Adding Behaviors

Install Mountebank

Setup Node.js(Windows)

Downloads

Latest LTS Version: **12.16.1** (includes npm 6.13.4)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users		Current Latest Features
Windows Installer	macOS Installer	Source Code
node-v12.16.1-x86.msi	node-v12.16.1.pkg	node-v12.16.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	
Linux Binaries (ARM)	ARMv7	ARMv8
Source Code	node-v12.16.1.tar.gz	

- Goto Node.js downloads and download Node.js for Windows
  - LTS
  - Current

Run the Node.js Installer(Windows)

- Welcome to the Node.js setup wizard
  - Select **Next**
- End-User License Agreement (EULA)
  - Check **I accept the terms in the License Agreement**
  - Select **Next**
- Destination Folder
  - Select **Next**
- Custom Setup
  - Select **Next**
- Ready to install Node.js
  - Select **Install**
  - *Note*: This step requires Administrator privileges.
  - If prompted, authenticate as an Administrator.
- Installing Node.js
  - Let the installer run to completion.
- Completed the Node.js Setup Wizard
  - Click **Finish**

Setup Node.js(Mac)

Install Homebrew

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Install Node.js

```
brew install node
```

Verify That Node.js was Properly Installed and Install Mountebank

Run command

```
node -v
```

Update npm

```
npm install npm -g
```

Install mountebank

```
npm install mountebank -g
```

Day1: Hello, World! Mountebank

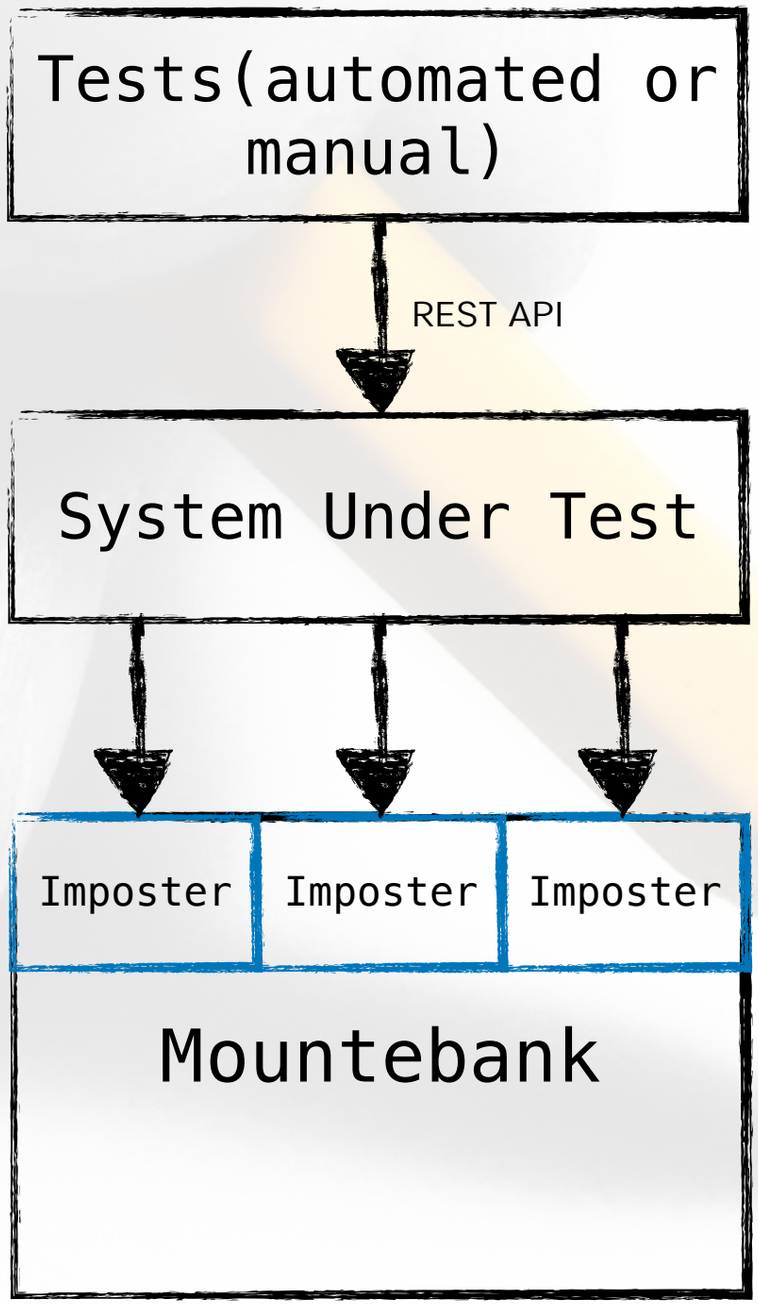
The Problem with End-to-End Testing

- Time
- Cost
- Slow feedback loop

3A pattern: Arrange, Act, Assert, Cleanup

- Setup data test
- Execute the system under test
- Assert the expected response
- Clean the state

Imposters: virtual services, a lightweight operation



A simple Mountebank imposter

- Mountebank provides configure virtual services, which are called imposters.
- Each imposter represents a socket that acts as the virtual service and accepts connections from the real service you are testing.
- Spinning up and shutting down imposters is a lightweight operation.

from: Testing Microservice with Mountebank page 17

How mountebank views an HTTP request

```
POST /products?page=2&itemsPerPage=2 HTTP/1.1
Host: api.petstore.com
Content-Type: application/json

{
  "key": "asdul7890"
}
```

HTTP request's view

```
{
  "method": "POST",
  "path": "/products",
  "query": {
    "page": 2,
    "itemsPrePage": 2
  },
  "headers": {
    "Host": "api.petstore.com",
    "Content-Type": "application/json"
  },
  "body": "{\n  \"key\": \"asdul7890\"\n}"
}
```

Mountebank's view

from: Testing Microservice with Mountebank page 26

How mountebank views an HTTP response

```
HTTP/1.1 200 OK
Date: Sun, 05 Apr 2020 10:10:10 GMT
Content-Type: application/json

{
  "key": "asdul7890"
}
```

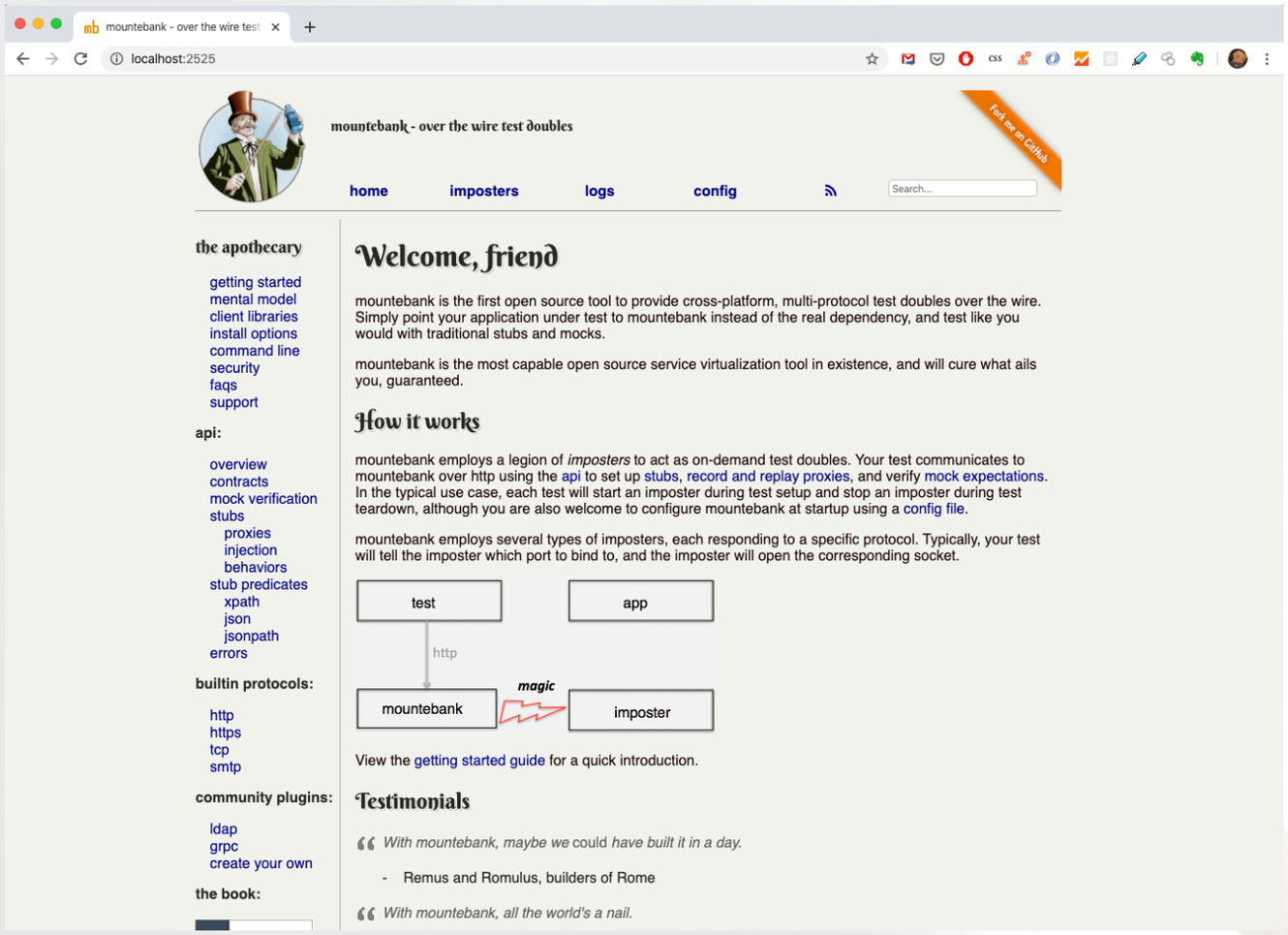
HTTP response's view

```
{
  "statusCode": 200,
  "headers": {
    "Date": "Sun, 05 Apr 2020 10:10:10 GMT",
    "Content-Type": "application/json"
  },
  "body": "{\n  \"key\": \"asdul7890\"\n}"
}
```

Mountebank's view

from: Testing Microservice with Mountebank page 27

Hello, World Mountebank



Create workspace call mb-workspace **unix windows**

```
mkdir mb-workspace
```

Change directory to mb-workspace **unix windows**

```
cd mb-workspace
```

Start Mountebank **unix windows**

```
mb
```

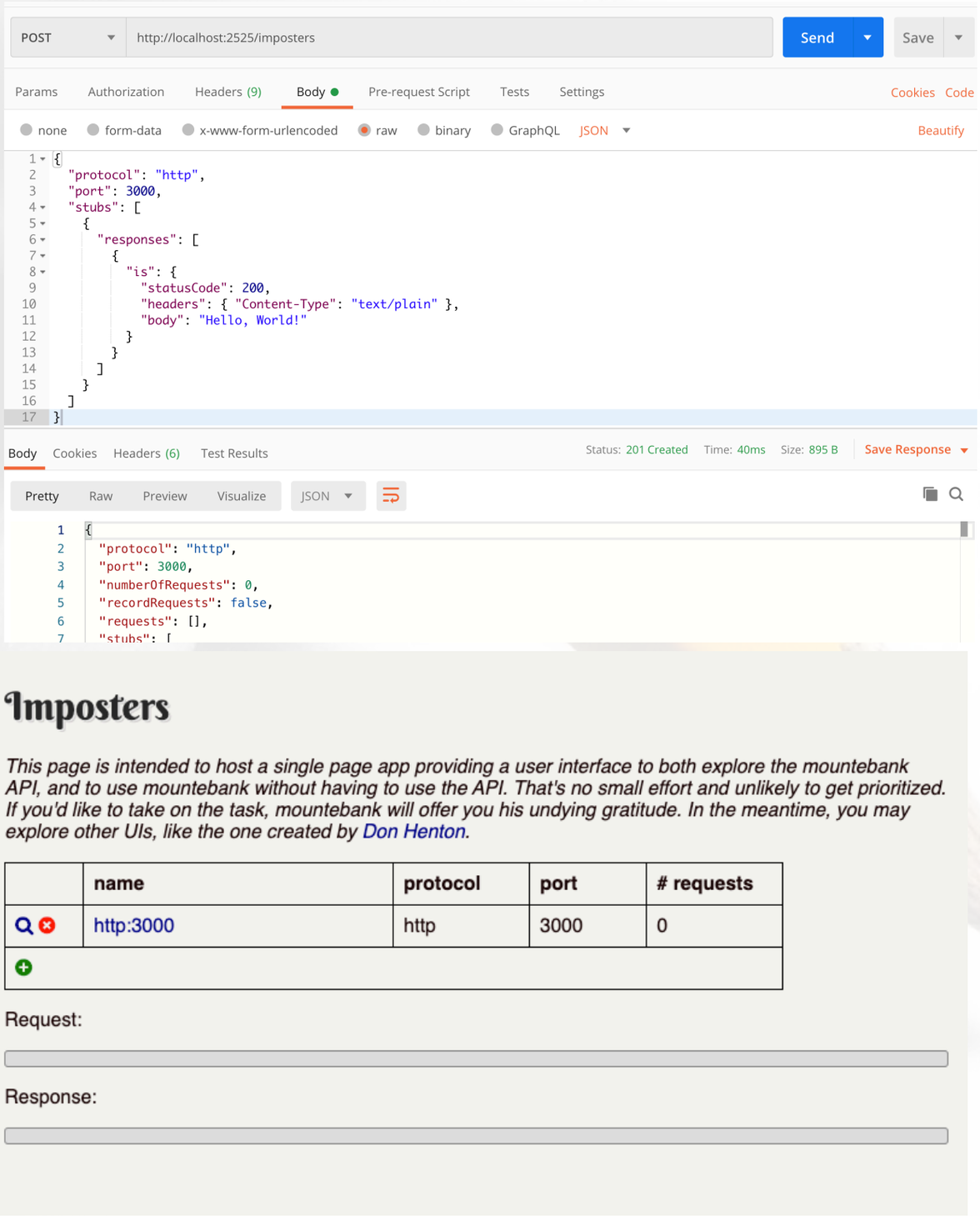
Open web browser then go to http://localhost:2525

Stop Mountebank **unix windows**

```
command + c
```

```
control + c
```

Create First Imposter



HTTP Method

POST

URL

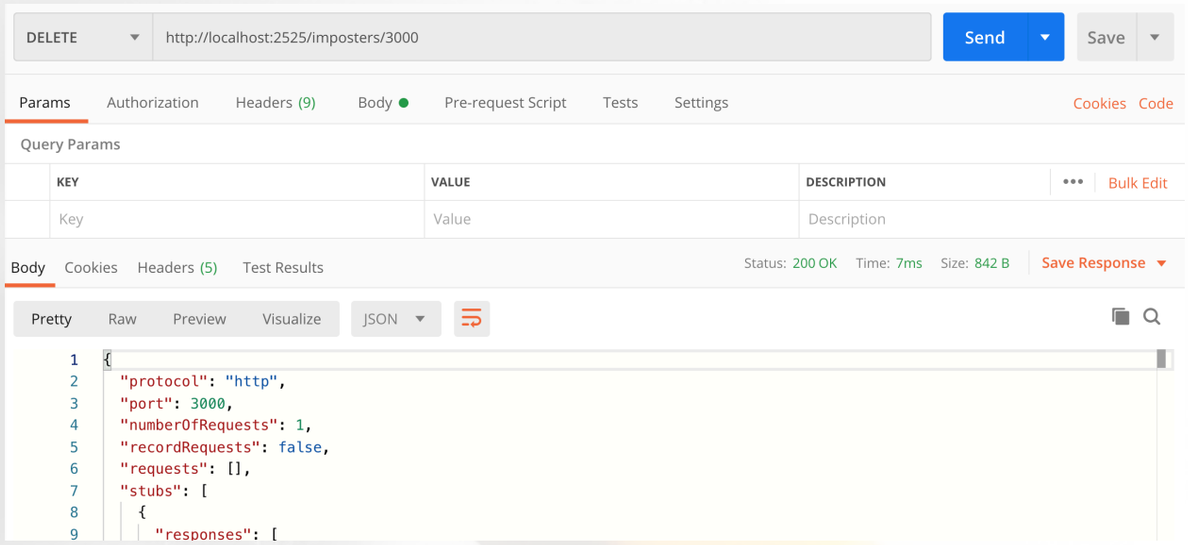
http://localhost:2525/imposters

Body

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": { "Content-Type": "text/plain" },
        "body": "Hello, World!"
      }
    }]
  }]
}
```

Open web browser then go to http://localhost:2525/impostes

Call The First Imposter

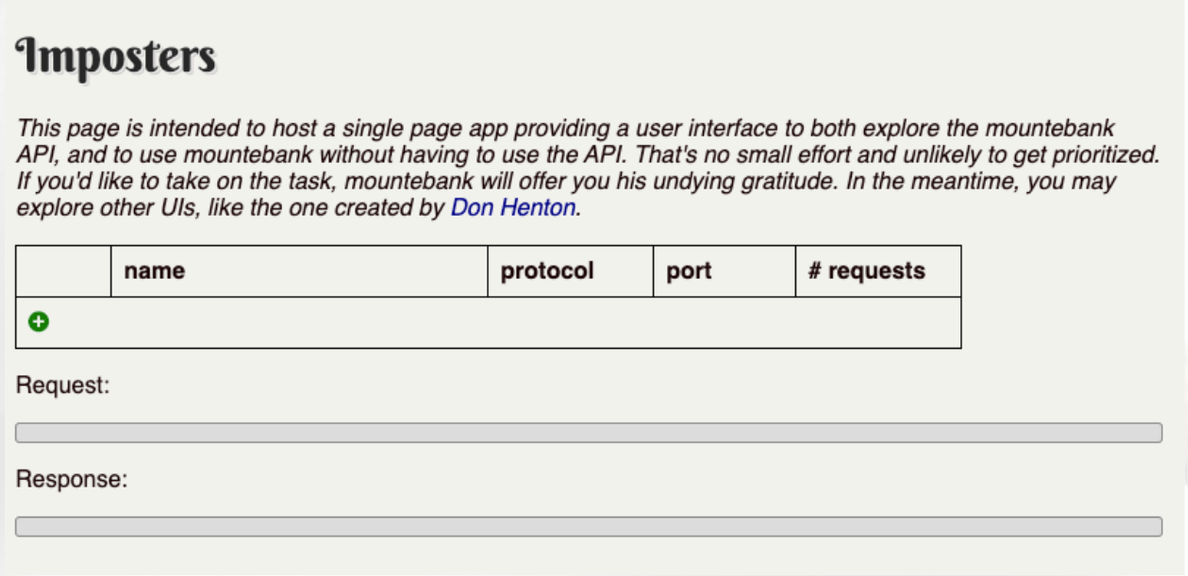


HTTP Method

DELETE

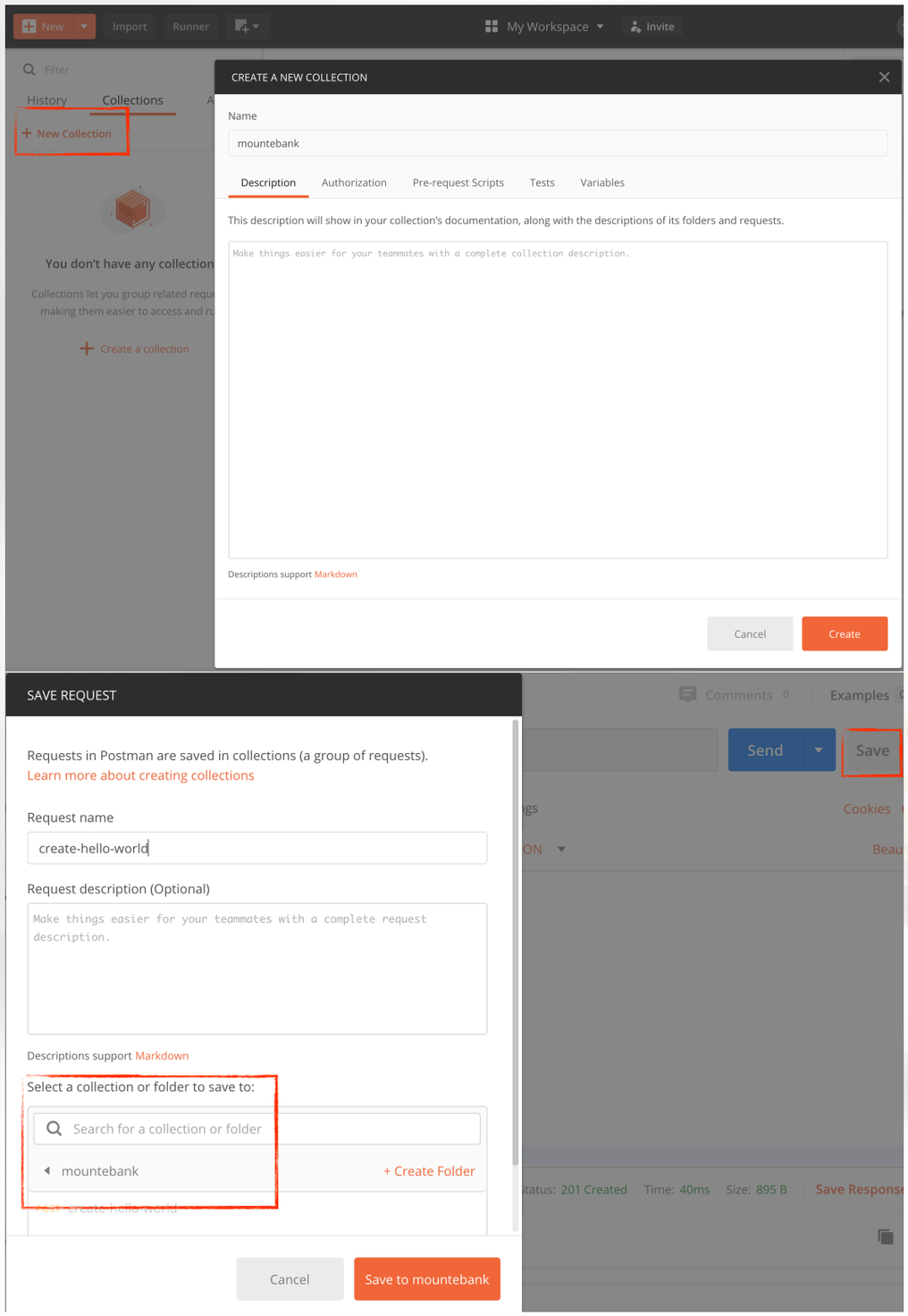
URL

http://localhost:2525/imposters/3000



Go to http://localhost:2525/impostes

Save Script to Collection



Click

+ New Collection

Name

mountebank

Click

Create

Click

Save

Request name

create-hello-world

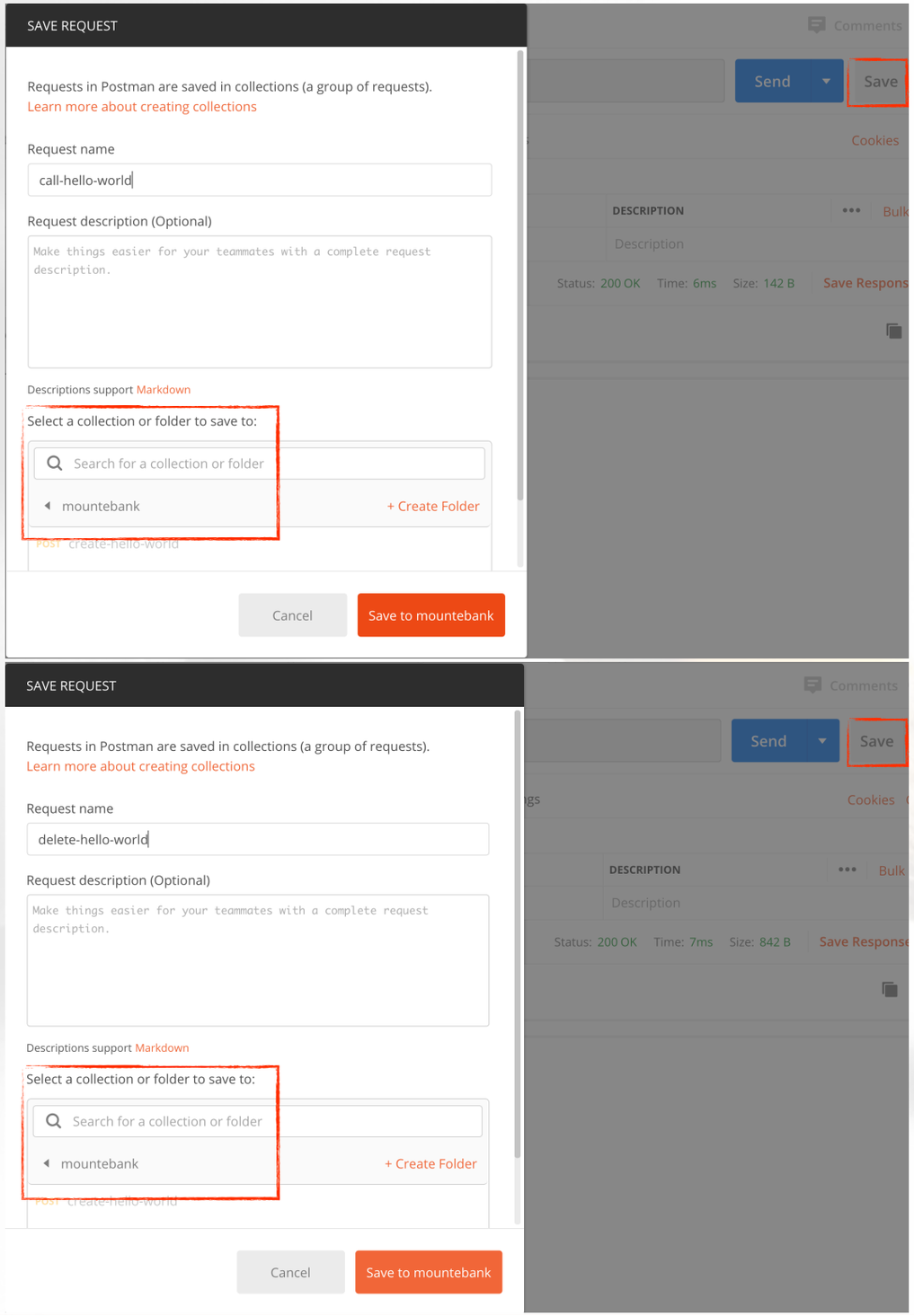
Select a collection

mountebank

Click

Save to mountebank

Save Script to Collection



Click

Save

Request name

call-hello-world

Select a collection

mountebank

Click

Save to mountebank

Click

Save

Request name

delete-hello-world

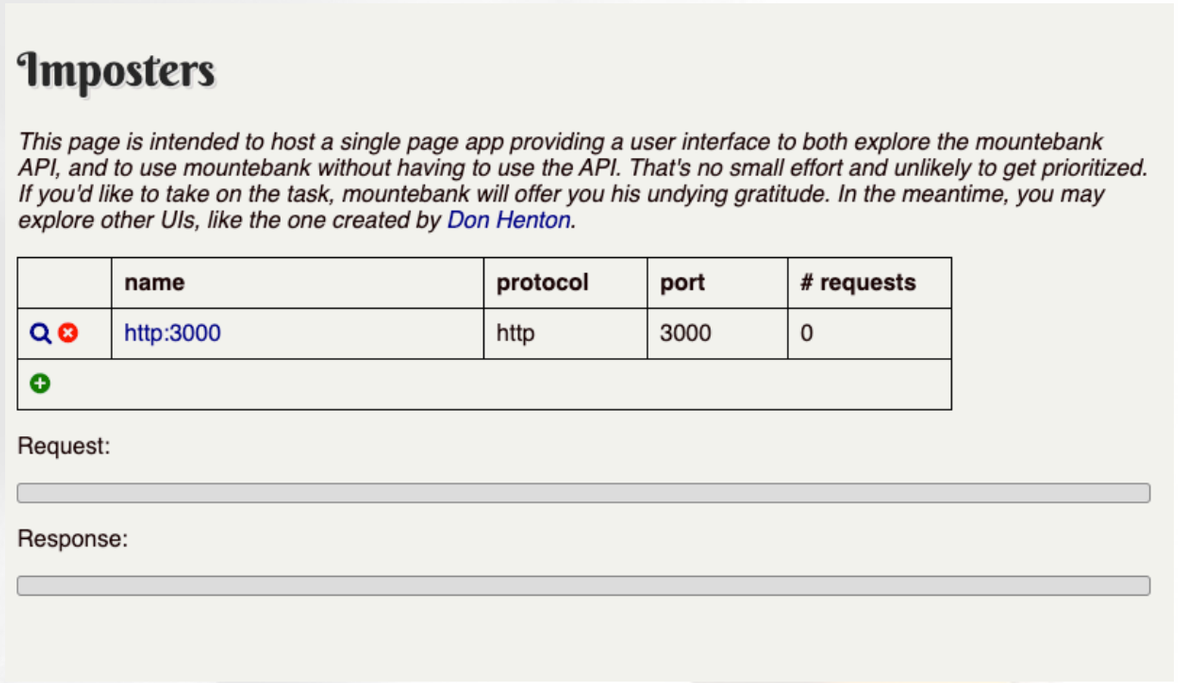
Select a collection

mountebank

Click

Save to mountebank

Create Imposters from Config File



Create file call hello-world.json

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": { "Content-Type": "text/plain" },
        "body": "Hello, World!"
      }
    }]
  }]
}
```

Start Mountebank **unix** **windows**

```
mb start --configfile hello-world.json
```

Open web browser then go to http://localhost:2525/impostes

The Basic of Mountebank HTTP Response

The HTTP Response Structure in JSON

The *is* response type, which is the fundamental building block for a stub.

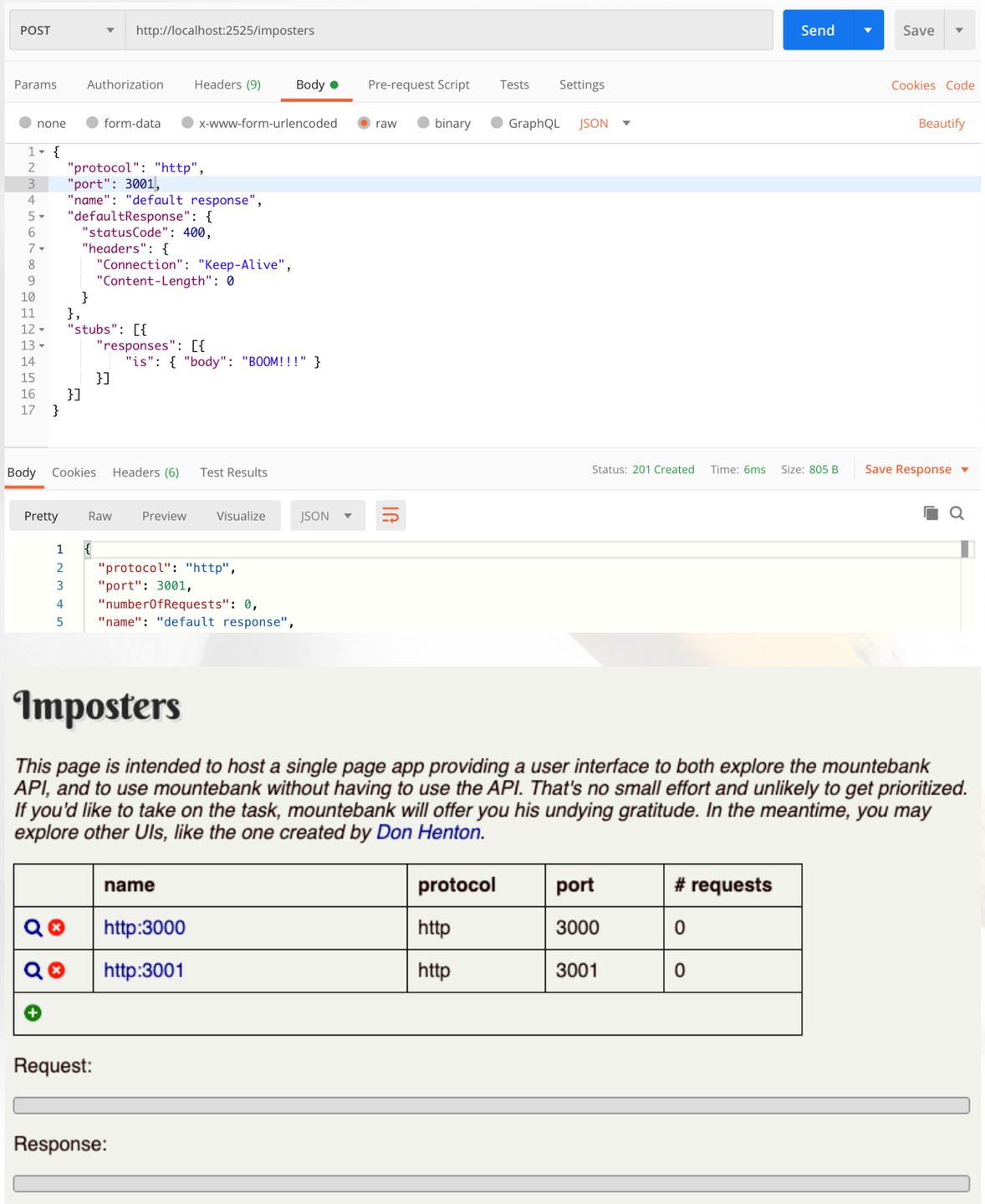
Imposter structure

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": { "Content-Type": "text/plain" },
        "body": "Hello, World!"
      }
    }]
  }]
}
```

The HTTP response structure in JSON

```
{
  "statusCode": 200,
  "headers": { "Content-Type": "text/plain" },
  "body": "Hello, World!"
}
```

Default Response



HTTP Method

POST

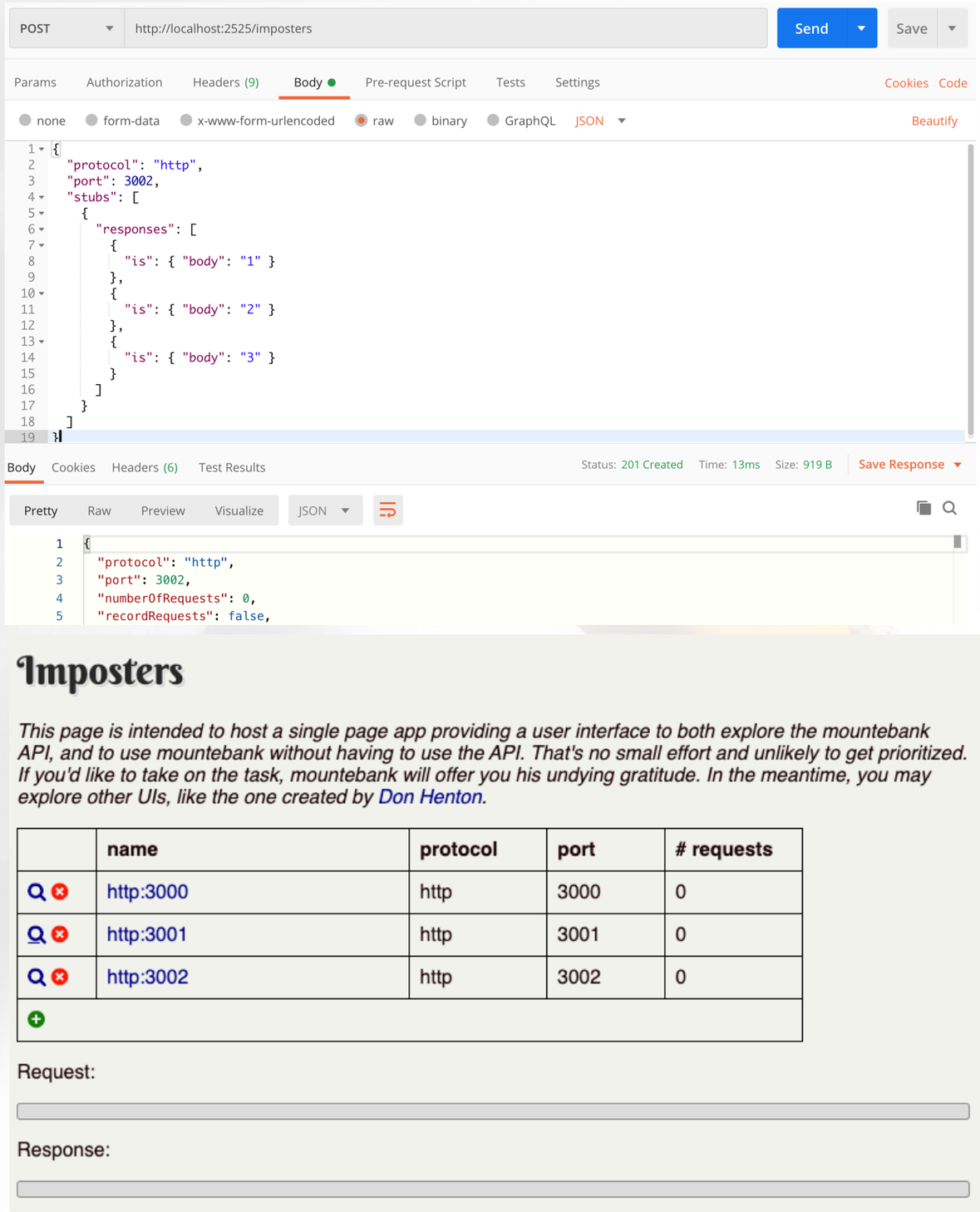
URL

http://localhost:2525/imposters

Body

```
{
  "protocol": "http",
  "port": 3001,
  "name": "default response",
  "defaultResponse": {
    "statusCode": 400,
    "headers": {
      "Connection": "Keep-Alive",
      "Content-Length": 0
    }
  },
  "stubs": [{
    "responses": [{
      "is": { "body": "BOOM!!!" }
    }]
  }]
}
```

Cycling Through Response



HTTP Method

POST

URL

http://localhost:2525/imposters

Body

```
{
  "protocol": "http",
  "port": 3002,
  "stubs": [
    {
      "responses": [
        {
          "is": { "body": "1" }
        },
        {
          "is": { "body": "2" }
        },
        {
          "is": { "body": "3" }
        }
      ]
    }
  ]
}
```

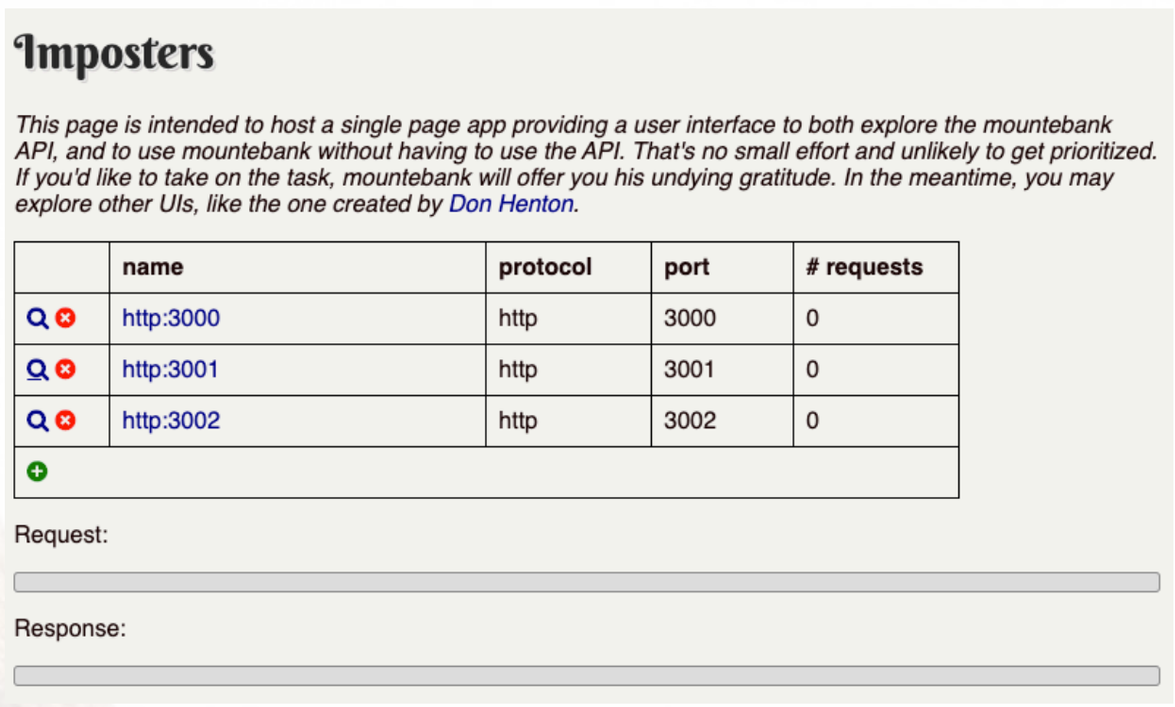
Create Imposters from Config File

Create file call imposters.json

```
{
  "imposters": [
    {
      "protocol": "http",
      "port": 3000,
      "stubs": [{
         "responses": [{
            "is": {
              "statusCode": 200,
              "headers": { "Content-Type": "text/plain" },
              "body": "Hello, World!"
            }
         }]
      }]
    },
    {
      "protocol": "http",
      "port": 3001,
      "name": "Default Response",
      "defaultResponse": {
        "statusCode": 400,
        "headers": {
          "Connection": "Keep-Alive",
          "Content-Length": 0
        }
      },
      "stubs": [{
         "responses": [{
            "is": { "body": "BOOM!!!" }
         }]
      }]
    },
    {
      "protocol": "http",
      "port": 3002,
      "stubs": [{
         "responses": [
            { "is": { "body": "1" }},
            { "is": { "body": "2" }},
            { "is": { "body": "3" }}
         ]
      }]
    }
  ]
}
```

Start Mountebank **unix** **windows**

```
mb start --configfile imposters.json
```



Create Imposters from Config File with EJS

Create file call imposters.ejs

```
{
  "imposters": [
    <%- include hello-world.json %>
  ]
}
```

Start Mountebank **unix** **windows**

```
mb start --configfile imposters.ejs
```

file hello-world.json

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": { "Content-Type": "text/plain" },
        "body": "Hello, World!"
      }
    }]
  }]
}
```

file default-response.json

```
{
  "protocol": "http",
  "port": 3001,
  "name": "default response",
  "defaultResponse": {
    "statusCode": 400,
    "headers": {
      "Connection": "Keep-Alive",
      "Content-Length": 0
    }
  },
  "stubs": [{
    "responses": [{
      "is": { "body": "BOOM!!!" }
    }]
  }]
}
```

file cycling-through-response.json

```
{
  "protocol": "http",
  "port": 3002,
  "stubs": [
    {
      "responses": [
        {
          "is": { "body": "1" }
        },
        {
          "is": { "body": "2" }
        },
        {
          "is": { "body": "3" }
        }
      ]
    }
  ]
}
```

Create Imposters from Config File with EJS

Create file call imposters.ejs

```
{
  "imposters": [
    <%- include hello-world.json %>,
    <%- include default-response.json %>,
    <%- include cycling-through-response.json %>
  ]
}
```

Start Mountebank **unix** **windows**

```
mb start --configfile imposters.ejs
```

file hello-world.json

```
{
  "protocol": "http",
  "port": 3000,
  "stubs": [{
    "responses": [{
      "is": {
        "statusCode": 200,
        "headers": { "Content-Type": "text/plain" },
        "body": "Hello, World!"
      }
    }]
  }]
}
```

file default-response.json

```
{
  "protocol": "http",
  "port": 3001,
  "name": "default response",
  "defaultResponse": {
    "statusCode": 400,
    "headers": {
      "Connection": "Keep-Alive",
      "Content-Length": 0
    }
  },
  "stubs": [{
    "responses": [{
      "is": { "body": "BOOM!!!" }
    }]
  }]
}
```

file cycling-through-response.json

```
{
  "protocol": "http",
  "port": 3002,
  "stubs": [
    {
      "responses": [
        {
          "is": { "body": "1" }
        },
        {
          "is": { "body": "2" }
        },
        {
          "is": { "body": "3" }
        }
      ]
    }
  ]
}
```

Basic Command and Options

Start Mountebank **unix**

```
mb start --configfile imposters.ejs --pidfile &
```

windows

```
mb start --configfile imposters.ejs
```

Stop Mountebank **unix**

```
mb stop
```

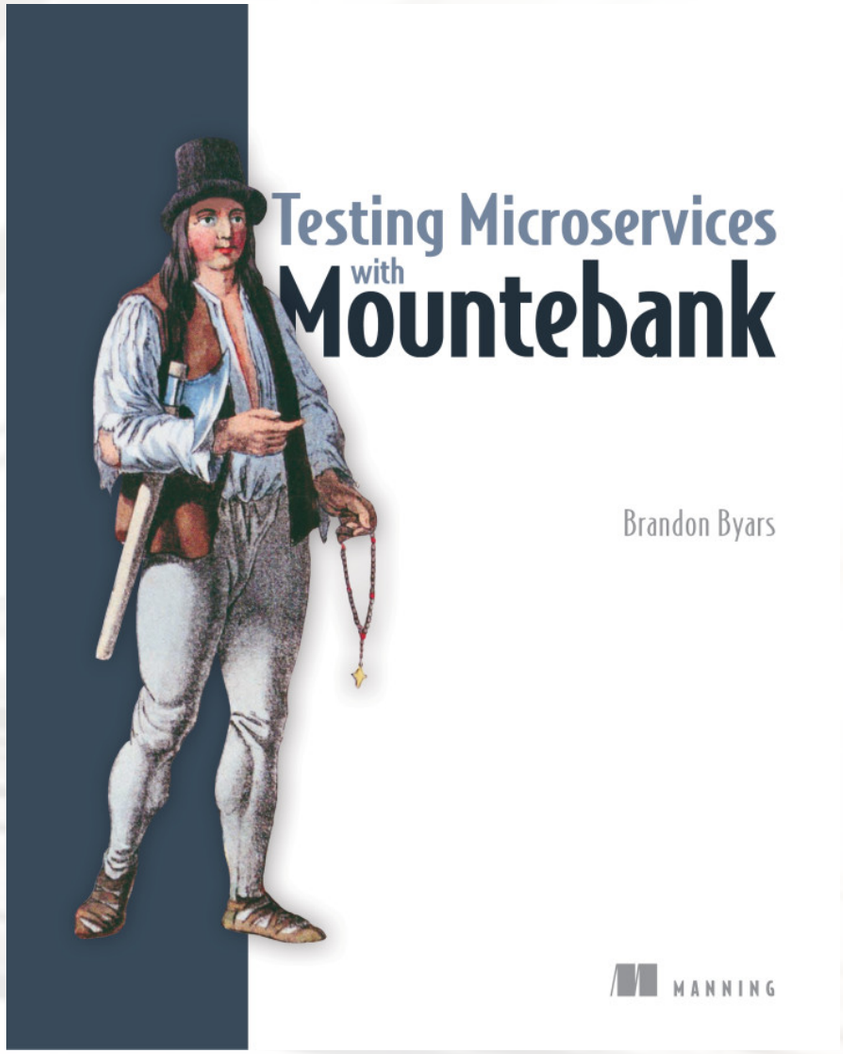
windows

```
control + c
```

Restart Mountebank **unix**

```
mb restart --configfile imposters.ejs --pidfile &
```

Books to Read and Practice



https://www.manning.com/books/testing-microservices-with-mountebank