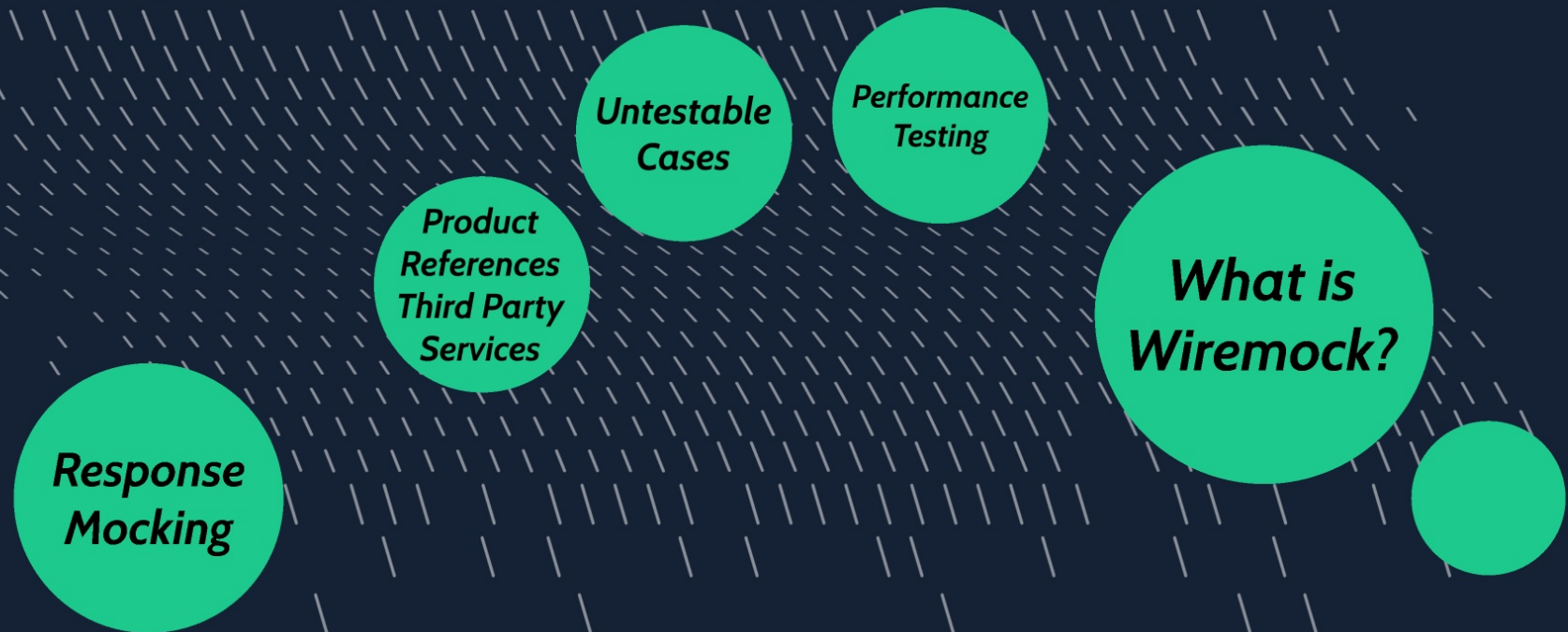


Wiremock in API and Performance Testing



Mocking in Testing

***TO MOCK
OR NOT
TO MOCK***



***What is a
Mock?***

***Why are
Mocks
Useful in
Testing?***

What is a Mock?

Dictionary term: to imitate
(someone or something) closely.

Mocks are objects that simulate
behavior of the real objects.

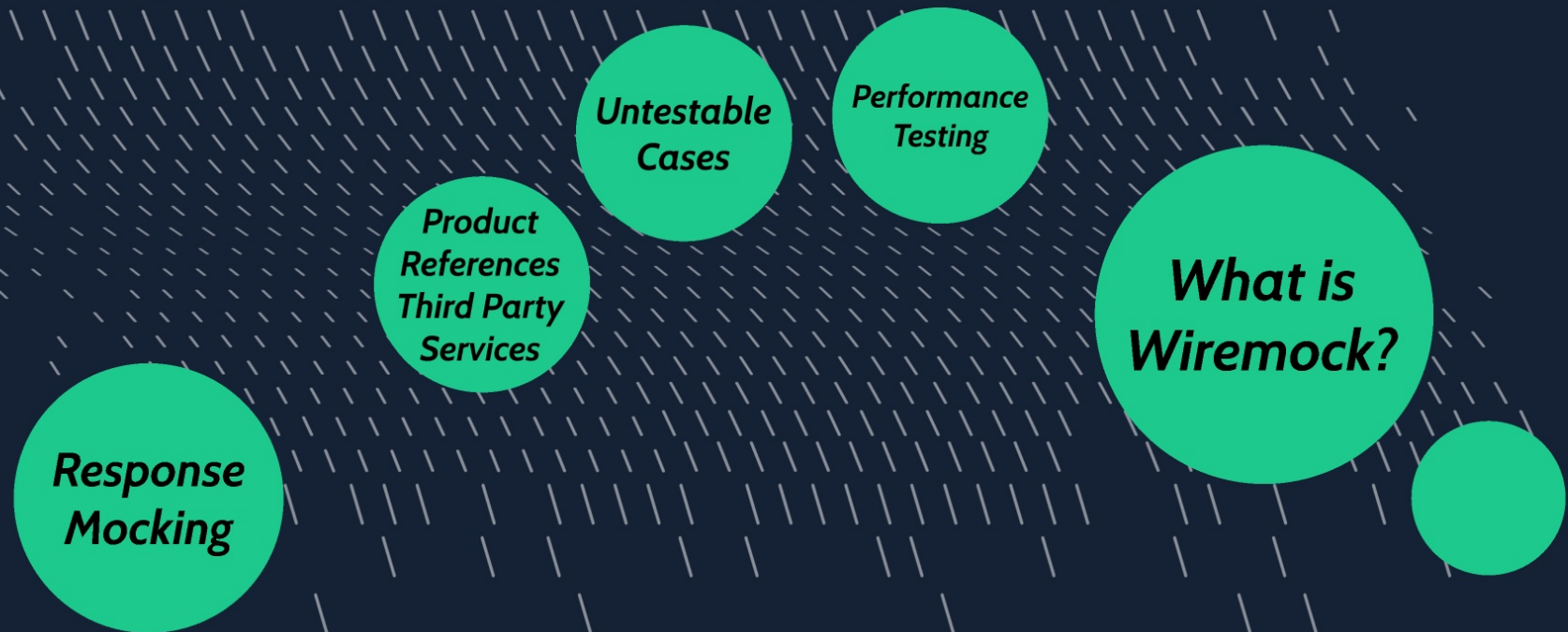
Mocking is a way to isolate code or
components to ensure that unit tests run
against the testable unit of code only
without actually utilizing other
components or dependencies of an
application.

Why are Mocks Useful in Testing?

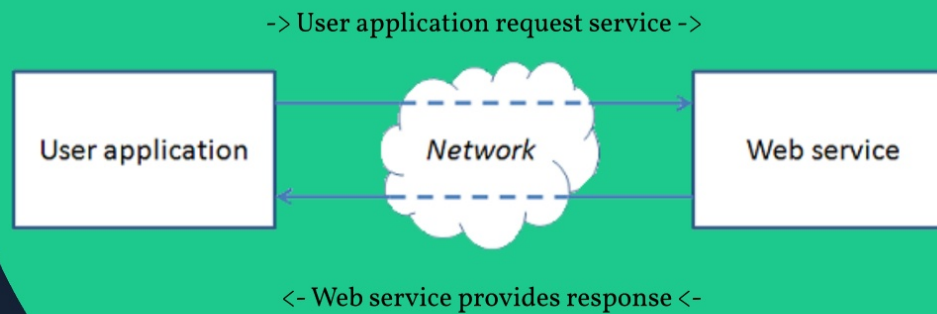
Mocking is primarily used in unit testing.

Mock objects provide you with the ability to test what you write without having to address dependency concerns.

Wiremock in API and Performance Testing



Web Application References Third Party Services



***New Planned
Application
Functionality***

***Service
Development
Delayed***



New Application Functionality on a Way

We want to start testing the
Application as soon as possible.

We don't want to wait till the
Service is fully ready and
deployed to be pointed to.

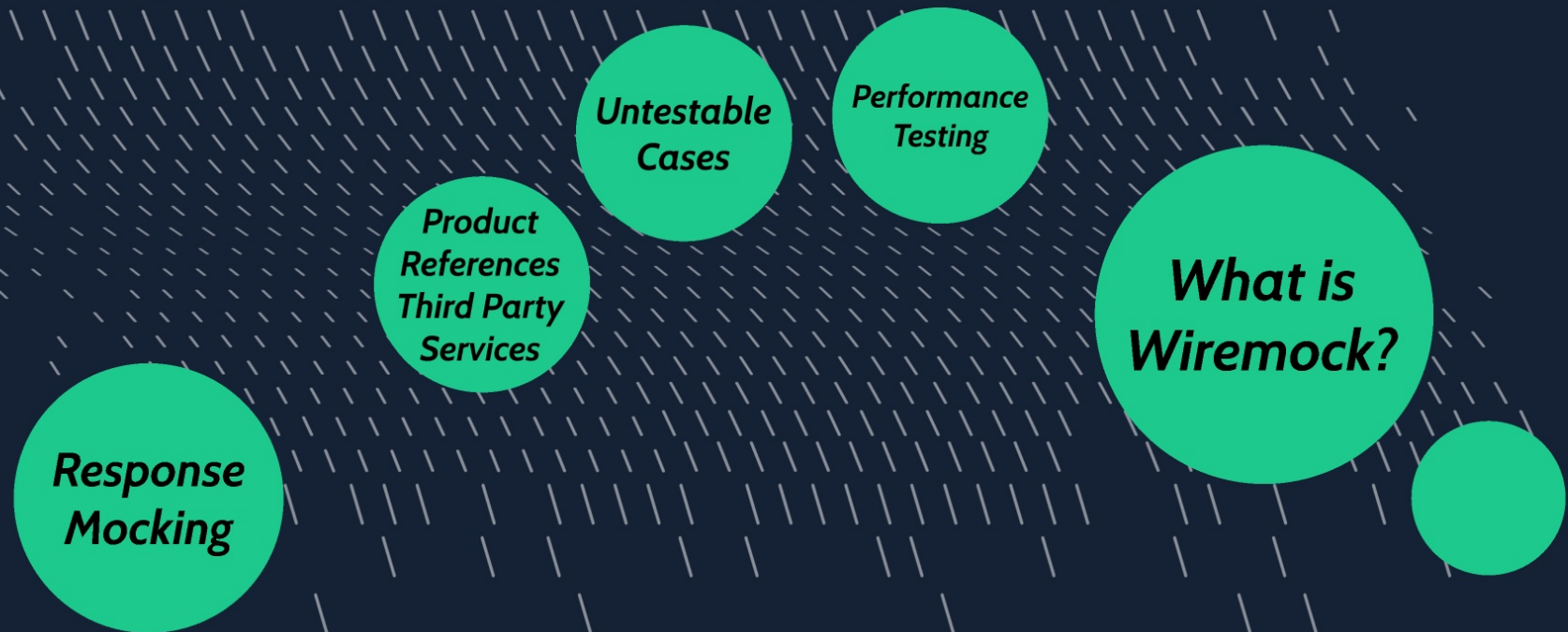


Service Deployment Delayed

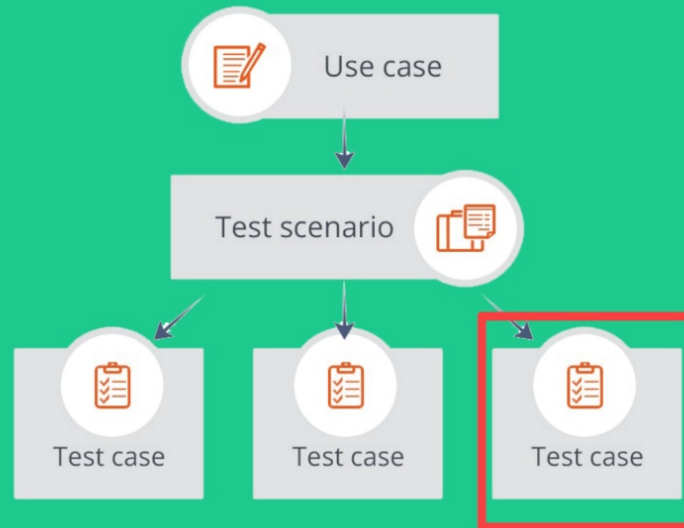
Services development team is having troubles with resources and cannot create the Service on time.

Service deployment delay affects the Application development process deadlines.

Wiremock in API and Performance Testing



Untestable Cases



***Errors and
Response
Codes***

***Other
Hardly
Testable
Cases***

Errors and Response Codes

It is difficult to invoke every possible Error Code or Error Message in real life situations.

Error handling test cases cannot be automated in a lot of circumstances.

Calling the service in real life it might give back a number of different HTTP response status codes: 200, 500...

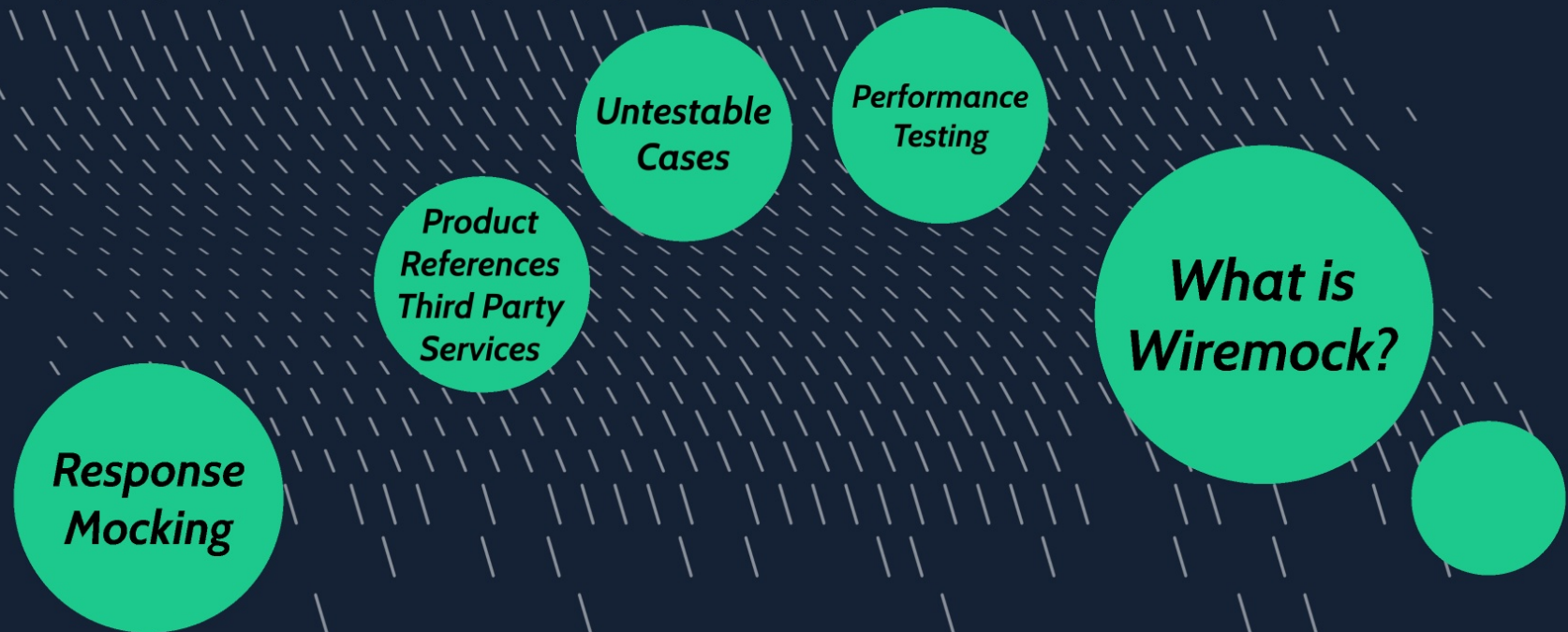
Hardly Testable Cases

Data does not exist in the database
but we need to test particular
values.

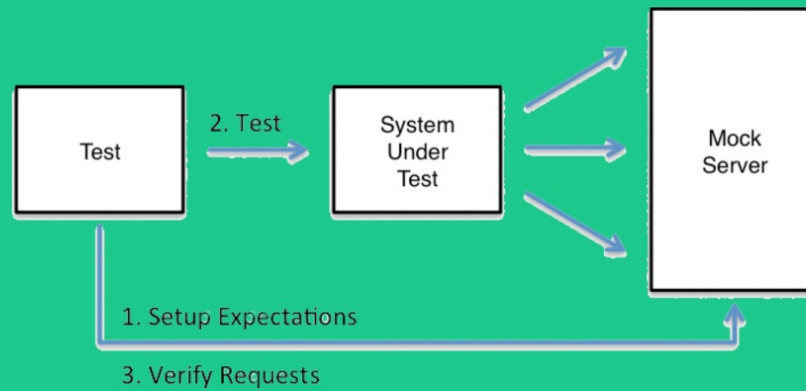
We have to test how application
handles different data formats.

Any other test cases that would
be difficult to get calling a real
Service endpoint.

Wiremock in API and Performance Testing



Performance Testing



**Actual
Service
Unavailable**

**Overloaded
Service**

**Unnecessary
Records
Entered in the
Database**

Actual Service Unavailable

Performance Tests may be dependent on actual service availability.

Scheduled Performance Tests run times have to be planned in advance to meet service availability timings.

Overloaded Service

We do not want to hit Service with too many requests when not necessary.

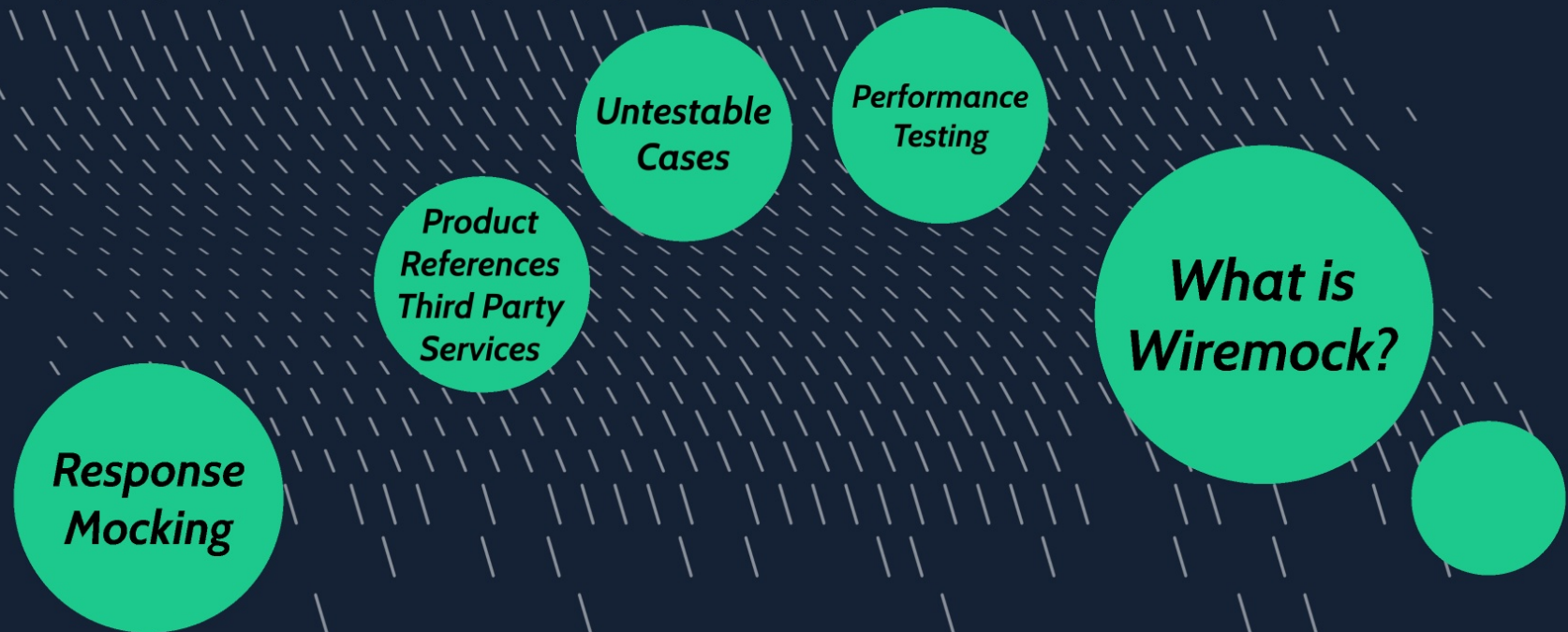
We do not want to affect the Service with unreal data sent to it.



Effect to a Database

We do not want to perform
unnecessary reads from the
database.

Wiremock in API and Performance Testing



What is *?*

WireMock is a simulator for
HTTP-based APIs.

It enables you to stay productive
when an API you depend on doesn't
exist or isn't complete.

Start

Functions

Demo



Where to Start

JAVA Runtime Environment

Wiremock standalone JAR

Run it

Run it



Run it

```
java -jar wiremock-standalone-2.25.1.jar  
--verbose --port 9001
```

Run it

```
java -jar wiremock-standalone-2.25.1.jar  
--verbose --port 9001
```

Main Wiremock Functions

Stubbing
Get All Mappings
Delete Mappings
Verifying
Matching
Record and Playback

Stubbing

*Get All
Mappings*

*Delete
Mappings*

Verifying

Matching

*Record and
Playback*

Stubbing

POST

http://localhost:9001/_admin/mappings

```
{
  "request": {
    "method": "GET",
    "url": "/bugsaloud"
  },
  "response": {
    "status": 200,
    "body": "Hello bugs'a'louders!",
    "headers": {
      "Content-Type": "text/plain"
    }
  }
}
```


Stubbing

POST



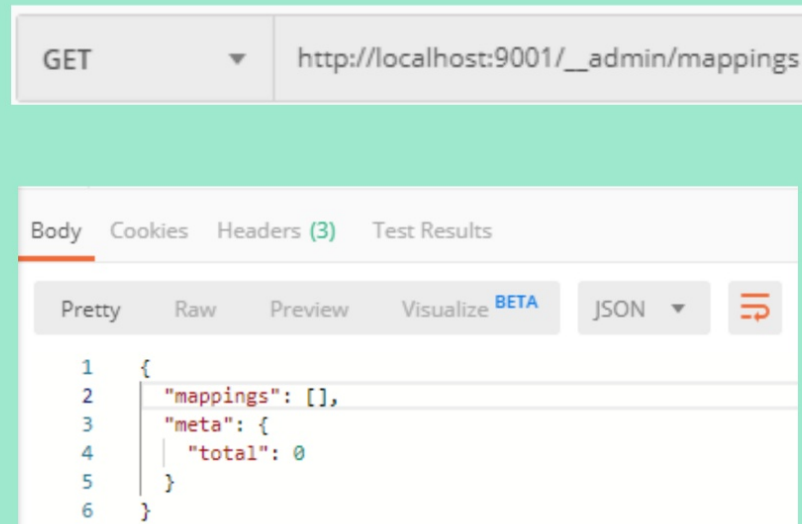
http://localhost:9001/__admin/mappings

```
{  
  "request": {  
    "method": "GET",  
    "url": "http://localhost:9001/__admin/mappings"
```

```
{  
  "request": {  
    "method": "GET",  
    "url": "/bugsaloud"  
  },  
}
```

```
,  
  "response": {  
    "status": 200,  
    "body": "Hello bugs'a'louders!",  
    "headers": {  
      "Content-Type": "text/plain"  
    }  
  }  
}
```

Get All Mappings



Get All Mappings

GET



http://localhost:9001/__admin/mappings

Body

Cookies

Headers (3)

Test Results

Body Cookies Headers (3) Test Results

Pretty

Raw

Preview

Visualize BETA

JSON ▼



```
1  {  
2    "mappings": [],  
3    "meta": {  
4      "total": 0  
5    }  
6  }
```

Delete Mappings

Delete all mappings

DELETE



`http://localhost:9001/__admin/mappings/`

Delete specific mapping by ID

DELETE



`http://localhost:9001/__admin/mappings/{mapping-id}`

Delete Mappings

Delete all mappings

DELETE



http://localhost:9001/__admin/mappings/

Delete specific mapping by ID

Delete specific mapping by ID

DELETE ▼


http://localhost:9001/__admin/mappings/{mapping-id}

Verifying

GET ▼ http://localhost:9001/bugsaloud

Response

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize BETA Text ▼ 

```
1 Hello bugs'a'louder!
```


Verifying

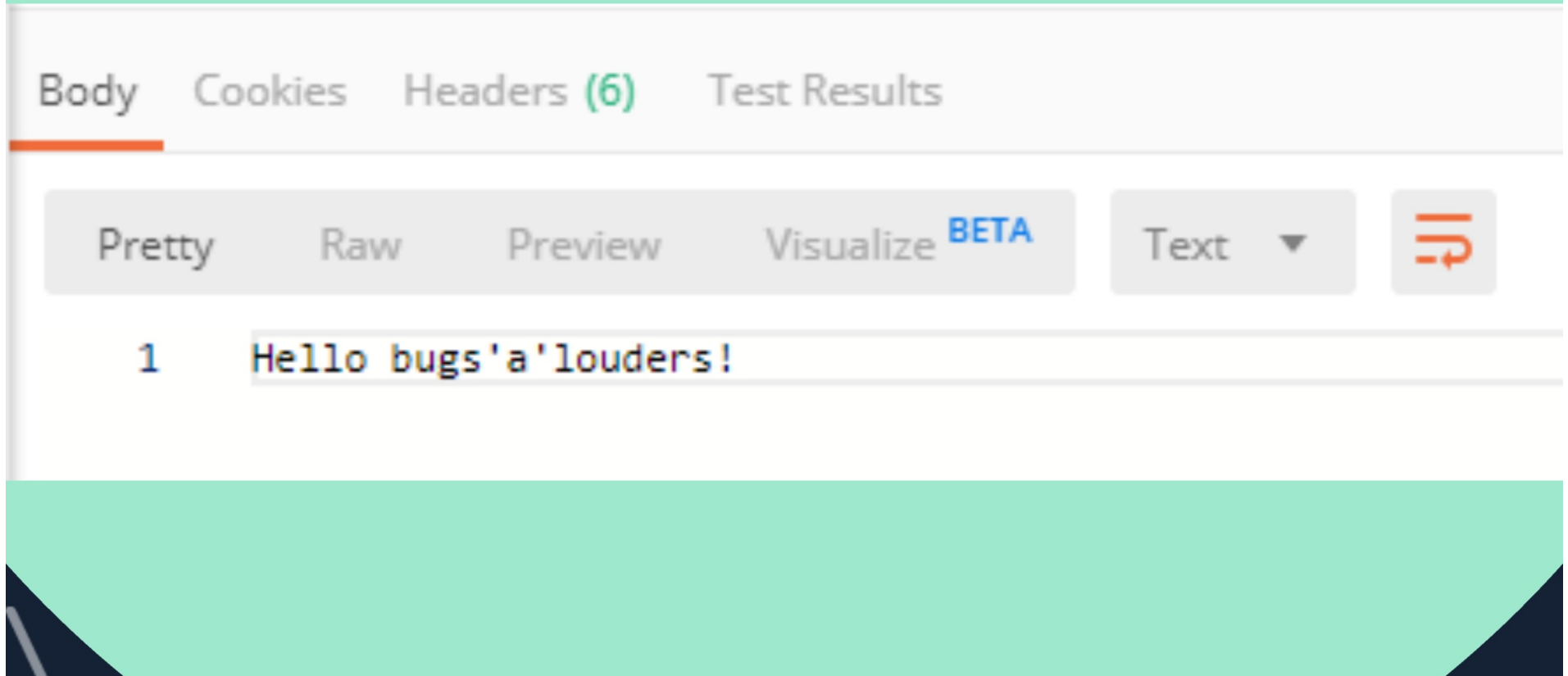
GET



`http://localhost:9001/bugsaloud`

Response

Response



Request Matching

URL
HTTP Method
Query parameters
Headers
Basic authentication (a special
case of header matching)
Cookies
Request body
Multipart/form-data

***Request
Body***

Request Body Matching

```
{  
  "request": {  
    "method": "ANY",  
    "url": "/getdoginfo",  
    "bodyPatterns": [ {  
      "matchesJsonPath" : "$[?(@.DogBreed == 'bulldog')]"  
    } ]  
  },  
  ...  
}
```

```
{
  "request": {
    "method": "ANY",
    "url": "/getdoginfo",
    "bodyPatterns" : [ {
      "matchesJsonPath" : "$[?(@.DogBreed == 'bulldog')]"
    } ]
  },
  ...
}
```


Record and Playback

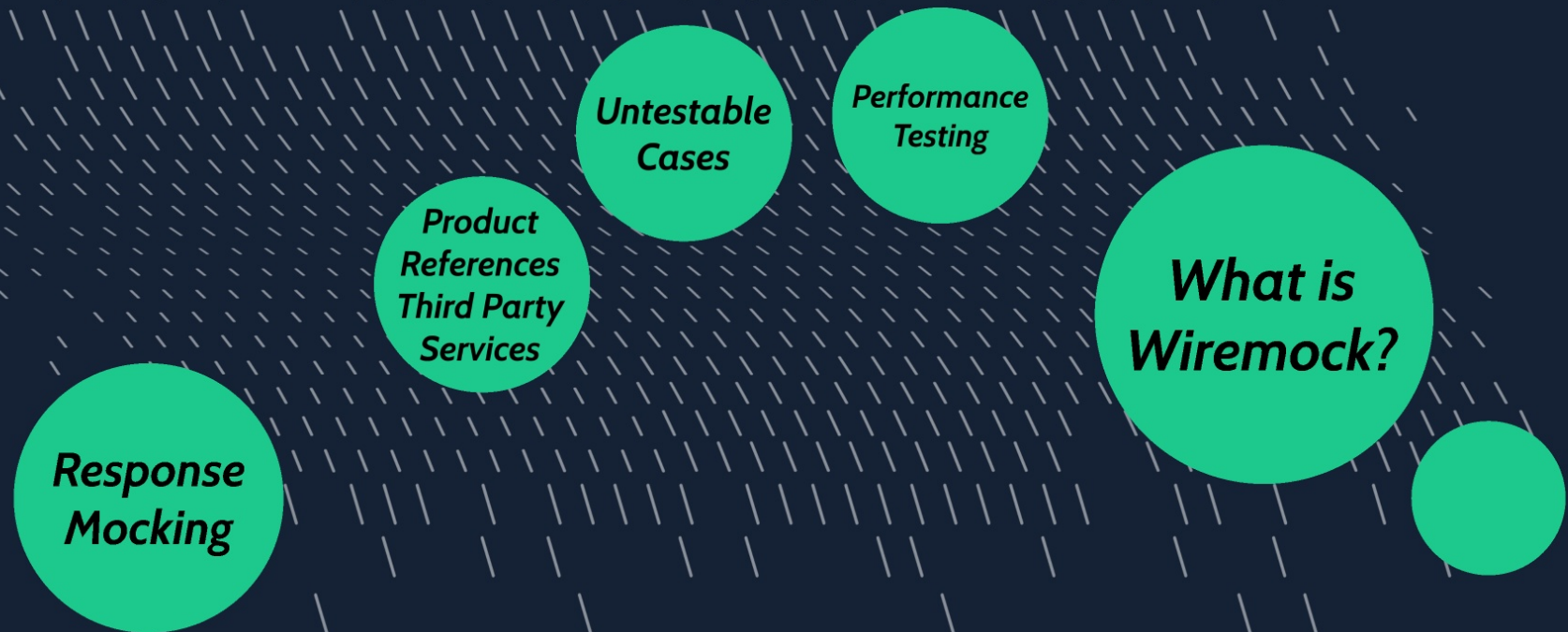


WireMock can create stub mappings from requests it has received.



Demo

Wiremock in API and Performance Testing





Thank You!

Questions?

Wiremock in API and Performance Testing

