

---

# **mbtest Documentation**

***Release 2.5.0***

**Simon Brunning**

**May 12, 2020**



CONTENTS:

<b>1</b>	<b>API Reference</b>	<b>3</b>
1.1	The <i>mbtest.server</i> module . . . . .	3
1.2	The <i>mbtest.imposters.imposters</i> module . . . . .	6
1.3	The <i>mbtest.imposters.stubs</i> module . . . . .	7
1.4	The <i>mbtest.imposters.predicates</i> module . . . . .	7
1.5	The <i>mbtest.imposters.responses</i> module . . . . .	10
1.6	The <i>mbtest.imposters.behaviors.copy</i> module . . . . .	12
1.7	The <i>mbtest.imposters.behaviors.lookup</i> module . . . . .	12
1.8	The <i>mbtest.imposters.behaviors.using</i> module . . . . .	13
1.9	The <i>mbtest.matchers</i> module . . . . .	15
1.10	The <i>mbtest.imposters.base</i> module . . . . .	17
<b>2</b>	<b>Indices and tables</b>	<b>19</b>
<b>3</b>	<b>Installation</b>	<b>21</b>
<b>4</b>	<b>Usage</b>	<b>23</b>
<b>5</b>	<b>Indices and tables</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>29</b>



Opinionated Python wrapper & utils for the [Mountebank](#) over the wire test double tool.  
Includes [pytest](#) fixture and [PyHamcrest](#) matchers.



## API REFERENCE

### 1.1 The *mbtest.server* module

`mbtest.server.mock_server(request, executable='node_modules/.bin/mb', port=2525, timeout=5, debug=True, allow_injection=True, local_only=True, data_dir='mbdb')`

Pytest fixture, making available a mock server, running one or more impostors, one for each domain being mocked.

Use in a pytest confest.py fixture as follows:

```
@pytest.fixture(scope="session")
def mock_server(request):
    return server.mock_server(request)
```

Test will look like:

```
def test_an_imposter(mock_server):
    imposter = Imposter(Stub(Predicate(path='/test'),
                             Response(body='sausages')),
                        record_requests=True)

    with mock_server(imposter) as s:
        r = requests.get('{0}/test'.format(imposter.url))

        assert_that(r, is_response().with_status_code(200).and_body("sausages"))
        assert_that(s, had_request(path='/test', method="GET"))
```

#### Parameters

- **request** (`FixtureRequest`) – Request for a fixture from a test or fixture function.
- **executable** (`Union[str, Path]`) – Alternate location for the Mountebank executable.
- **port** (`int`) – Server port.
- **timeout** (`int`) – specifies how long to wait for the Mountebank server to start.
- **debug** (`bool`) – Start the server in debug mode, which records all requests. This needs to be *True* for the `mbtest.matchers.had_request()` matcher to work.
- **allow\_injection** (`bool`) – Allow JavaScript injection. If *True*, *local\_only* should also be *True*, as per Mountebank security.
- **local\_only** (`bool`) – Accept request only from localhost.
- **data\_dir** (`Optional[str]`) – Persist all operations to disk, in this directory.

**Return type** *ExecutingMountebankServer*

**Returns** Mock server.

**class** `mbtest.server.MountebankServer` (*port*, *scheme='http'*, *host='localhost'*, *im-*  
*posters\_path='imposters'*)

Allow addition of imposters to an already running Mountebank mock server.

Test will look like:

```
def test_an_imposter(mock_server):
    mb = MountebankServer(1234)
    imposter = Imposter(Stub(Predicate(path='/test'),
                             Response(body='sausages')),
                        record_requests=True)

    with mb(imposter) as s:
        r = requests.get('{0}/test'.format(imposter.url))

        assert_that(r, is_response().with_status_code(200).and_body("sausages"))
        assert_that(s, had_request(path='/test', method="GET"))
```

Impostors will be torn down when the *with* block is exited.

#### Parameters

- **port** (*int*) – Server port.
- **scheme** (*str*) – Server scheme, if not *http*.
- **host** (*str*) – Server host, if not *localhost*.
- **imposters\_path** (*str*) – Impostors path, if not *imposters*.

**add\_imposters** (*definition*)

Add imposters to Mountebank server.

**Parameters** *definition* (*Imposter* or *list* (*Imposter*)) – One or more Imposters.

**Return type** *None*

**delete\_imposters** ()

**Return type** *None*

**get\_actual\_requests** ()

**Return type** *Mapping*[*int*, *Any*]

**property** *server\_url*

**Return type** *furl*

**imposter\_url** (*imposter\_port*)

**Return type** *furl*

**query\_all\_impostors** ()

**class** `mbtest.server.ExecutingMountebankServer` (*executable='node\_modules/.bin/mb'*,  
*port=2525*, *timeout=5*, *debug=True*,  
*allow\_injection=True*, *local\_only=True*,  
*data\_dir='.mbdb'*)

A Mountebank mock server, running one or more impostors, one for each domain being mocked.

Test will look like:



```

def test_an_imposter(mock_server):
    mb = ExecutingMountebankServer()
    imposter = Imposter(Stub(Predicate(path='/test'),
                               Response(body='sausages')),
                        record_requests=True)

    with mb(imposter) as s:
        r = requests.get('{0}/test'.format(imposter.url))

        assert_that(r, is_response().with_status_code(200).and_body("sausages"))
        assert_that(s, had_request(path='/test', method="GET"))

    mb.close()

```

The mountebank server will be started when this class is instantiated, and needs to be closed if it's not to be left running. Consider using the `mock_server()` pytest fixture, which will take care of this for you.

#### Parameters

- **executable** (`Union[str, Path]`) – Optional, alternate location for the Mountebank executable.
- **port** (`int`) – Server port.
- **timeout** (`int`) – How long to wait for the Mountebank server to start.
- **debug** (`bool`) – Start the server in debug mode, which records all requests. This needs to be `True` for the `mbtest.matchers.had_request()` matcher to work.
- **allow\_injection** (`bool`) – Allow JavaScript injection. If `True`, `local_only` should also be `True`, as per [Mountebank security](#).
- **local\_only** (`bool`) – Accept request only from localhost.
- **data\_dir** (`Optional[str]`) – Persist all operations to disk, in this directory.

`running = {}`

`start_lock = <unlocked _thread.lock object>`

`close()`

**Return type** `None`

**exception** `mbtest.server.MountebankException`

Exception using Mountebank server.

**exception** `mbtest.server.MountebankPortInUseException`

Mountebank server failed to start - port already in use.

**exception** `mbtest.server.MountebankTimeoutError`

Mountebank server failed to start in time.

## 1.2 The *mbtest.imposters.imposters* module

**class** `mbtest.imposters.imposters.Imposter` (*stubs*, *port=None*, *protocol=<Protocol.HTTP: 'http'>*, *name=None*, *record\_requests=True*)

Represents a [Mountebank imposter](#). Think of an imposter as a mock website, running a protocol, on a specific port. Required behaviors are specified using stubs.

**Parameters**

- **stubs** (`Union[Stub, Iterable[Stub]]`) – One or more Stubs.
- **port** (`Optional[int]`) – Port.
- **protocol** (`Protocol`) – Protocol to run on.
- **name** (`Optional[str]`) – Impostor name - useful for interactive exploration of impostors on <http://localhost:2525/impostors>
- **record\_requests** (`bool`) – Record requests made against this impostor, so they can be asserted against later.

**class** `Protocol`

Imposter `Protocol`.

`HTTP = 'http'`

`HTTPS = 'https'`

`SMTP = 'smtp'`

`TCP = 'tcp'`

**property** `url`

**Return type** `furl`

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** `Any`

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (`Any`) – JSON structure to be converted.

**Return type** `Imposter`

**Returns** Converted object.

`mbtest.imposters.imposters.smtp_imposter` (*name='smtp'*, *record\_requests=True*)

Canned SMTP server impostor.

**Return type** `Imposter`

## 1.3 The *mbtest.imposters.stubs* module

**class** `mbtest.imposters.stubs.Stub` (*predicates=None, responses=None*)

Represents a [Mountebank stub](#). Think of a stub as a behavior, triggered by a matching predicate.

### Parameters

- **predicates** (`Union[BasePredicate, Iterable[BasePredicate], None]`) – Trigger this stub if one of these predicates matches the request
- **responses** (`Union[BaseResponse, Iterable[BaseResponse], None]`) – Use these response behaviors (in order)

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** `Any`

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (`Any`) – JSON structure to be converted.

**Return type** `Stub`

**Returns** Converted object.

## 1.4 The *mbtest.imposters.predicates* module

**class** `mbtest.imposters.predicates.BasePredicate`

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (`Any`) – JSON structure to be converted.

**Return type** `BasePredicate`

**Returns** Converted object.

**class** `mbtest.imposters.predicates.LogicallyCombinablePredicate`

**class** `mbtest.imposters.predicates.Predicate` (*path=None, method=None, query=None, body=None, headers=None, xpath=None, operator=<Operator.EQUALS: 'equals'>, case\_sensitive=True*)

Represents a [Mountebank predicate](#). A predicate can be thought of as a trigger, which may or may not match a request.

### Parameters

- **path** (`Union[str, furl, None]`) – URL path.
- **method** (`Optional[Method]`) – HTTP method.
- **query** (`Optional[Mapping[str, Union[str, int, bool]]]`) – Query arguments, keys and values.

- **body** (`Optional[str]`) – Body text. Can be a string, or a JSON serialisable data structure.
- **headers** (`Optional[Mapping[str, str]]`) – Headers, keys and values.
- **xpath** (`Optional[str]`) – xpath query
- **operator** (`Operator`) –
- **case\_sensitive** (`bool`) –

**exception InvalidPredicateOperator**

**class Method**

Predicate HTTP method.

**DELETE** = 'DELETE'

**GET** = 'GET'

**HEAD** = 'HEAD'

**POST** = 'POST'

**PUT** = 'PUT'

**class Operator**

Predicate operator.

**EQUALS** = 'equals'

**DEEP\_EQUALS** = 'deepEquals'

**CONTAINS** = 'contains'

**STARTS\_WITH** = 'startsWith'

**ENDS\_WITH** = 'endsWith'

**MATCHES** = 'matches'

**EXISTS** = 'exists'

**classmethod has\_value** (*name*)

Return type `bool`

**as\_structure** ()

Converted to a JSON serializable structure.

Return type `Any`

Returns Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

Parameters **structure** (`Any`) – JSON structure to be converted.

Return type `Predicate`

Returns Converted object.

**class** `mbtest.imposters.predicates.AndPredicate` (*left, right*)

**as\_structure** ()

Converted to a JSON serializable structure.

Return type `Any`

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *AndPredicate*

**Returns** Converted object.

**class** `mbtest.imposters.predicates.OrPredicate` (*left, right*)

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *OrPredicate*

**Returns** Converted object.

**class** `mbtest.imposters.predicates.TcpPredicate` (*data*)

Represents a [Mountebank TCP predicate](#). A predicate can be thought of as a trigger, which may or may not match a request.

**Parameters** **data** (*str*) – Data to match the request.

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *TcpPredicate*

**Returns** Converted object.

**class** `mbtest.imposters.predicates.InjectionPredicate` (*inject*)

Represents a [Mountebank injection predicate](#). A predicate can be thought of as a trigger, which may or may not match a request.

Injection requires Mountebank version 2.0 or higher.

**Parameters** **inject** (*str*) – JavaScript function to inject.

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *InjectionPredicate*

**Returns** Converted object.

## 1.5 The *mbtest.imposters.responses* module

**class** *mbtest.imposters.responses.BaseResponse*

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *BaseResponse*

**Returns** Converted object.

**class** *mbtest.imposters.responses.Response* (*body=""*, *status\_code=200*, *wait=None*, *repeat=None*, *headers=None*, *mode=None*, *copy=None*, *decorate=None*, *lookup=None*, *shell\_transform=None*)

Represents a [Mountebank](#) 'is' response behavior.

### Parameters

- **body** (*str*) – Body text for response. Can be a string, or a JSON serialisable data structure.
- **status\_code** (*Union[int, str]*) – HTTP status code
- **wait** (*Union[int, str, None]*) – Add latency, in ms.
- **repeat** (*Optional[int]*) – Repeat this many times before moving on to next response.
- **headers** (*Optional[Mapping[str, str]]*) – Response HTTP headers
- **mode** (*Optional[Mode]*) – Mode - text or binary
- **copy** (*Optional[Copy]*) – Copy behavior
- **decorate** (*Optional[str]*) – Decorate behavior.
- **lookup** (*Optional[Lookup]*) – Lookup behavior
- **shell\_transform** (*Union[str, Iterable[str], None]*) – shellTransform behavior

**class** *Mode*

An enumeration.

**TEXT** = 'text'

**BINARY** = 'binary'

**property** *body*

**Return type** *str*

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** *structure* (*Any*) – JSON structure to be converted.

**Return type** *Response*

**Returns** Converted object.

**class** `mbtest.imposters.responses.TcpResponse` (*data*)

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** *structure* (*Any*) – JSON structure to be converted.

**Return type** *TcpResponse*

**Returns** Converted object.

**class** `mbtest.imposters.responses.Proxy` (*to*, *wait=None*, *inject\_headers=None*,  
*mode=<Mode.ALWAYS: 'proxyAlways'>*)

Represents a [Mountebank proxy](#).

**Parameters** *to* (*Union[furl, str]*) – The origin server, to which the request should proxy.

**class** `Mode`

Defines the replay behavior of the proxy.

**ONCE** = 'proxyOnce'

**ALWAYS** = 'proxyAlways'

**TRANSPARENT** = 'proxyTransparent'

**as\_structure** ()

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** *structure* (*Any*) – JSON structure to be converted.

**Return type** *Proxy*

**Returns** Converted object.

**class** `mbtest.imposters.responses.InjectionResponse` (*inject*)

Represents a [Mountebank injection response](#).

Injection requires Mountebank version 2.0 or higher.

**Parameters** *inject* (*str*) – JavaScript function to inject .

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *InjectionResponse*

**Returns** Converted object.

## 1.6 The *mbtest.imposters.behaviors.copy* module

**class** *mbtest.imposters.behaviors.copy.Copy* (*from\_, into, using*)

Represents a *copy* behavior.

**Parameters**

- **from** – The name of the request field to copy from, or, if the request field is an object, then an object specifying the path to the request field.
- **into** (*str*) – The token to replace in the response with the selected request value.
- **using** (*Using*) – The configuration needed to select values from the response.

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *Copy*

**Returns** Converted object.

## 1.7 The *mbtest.imposters.behaviors.lookup* module

**class** *mbtest.imposters.behaviors.lookup.Lookup* (*key, datasource\_path, data-source\_key\_column, into*)

Represents a *lookup* behavior.

**Parameters**

- **key** (*Key*) – How to select the key from the request.
- **datasource\_path** (*Union[str, Path]*) – The path to the data source.
- **datasource\_key\_column** (*str*) – The header of the column to match against the key.
- **into** (*str*) – The token to replace in the response with the selected request value.



**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *Lookup*

**Returns** Converted object.

**class** `mbtest.imposters.behaviors.lookup.Key` (*from\_, using, index=0*)

The information on how to select the key from the request.

**Parameters**

- **from** – The name of the request field to copy from, or, if the request field is an object, then an object specifying the path to the request field.
- **using** (*Using*) – The configuration needed to select values from the response
- **index** (*int*) – Index of the item from the result array to be selected.

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *Key*

**Returns** Converted object.

## 1.8 The *mbtest.imposters.behaviors.using* module

**class** `mbtest.imposters.behaviors.using.Using` (*method, selector*)

How to select values from the response.

**Parameters**

- **method** (*Method*) – The method used to select the value(s) from the request.
- **selector** (*str*) – The selector used to select the value(s) from the request.

**class** `Method`

An enumeration.

**REGEX** = 'regex'

**XPATH** = 'xpath'

**JSONPATH** = 'jsonpath'

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *Using*

**Returns** Converted object.

**class** `mbtest.imposters.behaviors.using.UsingRegex` (*selector, ignore\_case=False, multiline=False*)

Select values from the response using a regular expression.

**Parameters**

- **selector** (*str*) – The selector used to select the value(s) from the request.
- **ignore\_case** (*bool*) – Uses a case-insensitive regular expression
- **multiline** (*bool*) – Uses a multiline regular expression

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *UsingRegex*

**Returns** Converted object.

**class** `mbtest.imposters.behaviors.using.UsingXpath` (*selector, ns=None*)

Select values from the response using an xpath expression.

**Parameters**

- **selector** (*str*) – The selector used to select the value(s) from the request.
- **ns** (*Optional[Mapping[str, str]]*) – The ns object maps namespace aliases to URLs

**as\_structure()**

Converted to a JSON serializable structure.

**Return type** *Any*

**Returns** Structure suitable for JSON serialisation.

**static from\_structure(structure)**

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *UsingXpath*

**Returns** Converted object.

**class** `mbtest.imposters.behaviors.using.UsingJsonpath(selector)`

Select values from the response using a jsonpath expression.

**Parameters** `selector` (`str`) – The selector used to select the value(s) from the request.

**static** `from_structure(structure)`

Converted from a JSON serializable structure.

**Parameters** `structure` – JSON structure to be converted.

**Return type** `UsingJsonpath`

**Returns** Converted object.

## 1.9 The *mbtest.matchers* module

```
mbtest.matchers.had_request (method=<hamcrest.core.core.isanything.IsAnything object>,
                             path=<hamcrest.core.core.isanything.IsAnything object>,
                             query=<hamcrest.core.core.isanything.IsAnything object>,
                             headers=<hamcrest.core.core.isanything.IsAnything object>,
                             body=<hamcrest.core.core.isanything.IsAnything object>,
                             times=<hamcrest.core.core.isanything.IsAnything object>)
```

Mountebank server has recorded call matching.

Build criteria with *with\_* and *and\_* methods:

```
assert_that(server, had_request().with_path("/test").and_method("GET"))
```

Available attributes as per parameters.

### Parameters

- **method** (`Union[str, Matcher[str]]`) – Request’s method matched...
- **path** (`Union[furl, str, Matcher[Union[furl, str]]]`) – Request’s path matched...
- **query** (`Union[Mapping[str, str], Matcher[Mapping[str, str]]]`) – Request’s query matched...
- **headers** (`Union[Mapping[str, str], Matcher[Mapping[str, str]]]`) – Request’s headers matched...
- **body** (`Union[str, Matcher[str]]`) – Request’s body matched...
- **times** (`Union[int, Matcher[int]]`) – Request’s number of times called matched...

**Return type** `Matcher[MountebankServer]`

```
class mbtest.matchers.HadRequest (method=<hamcrest.core.core.isanything.IsAnything object>,
                                  path=<hamcrest.core.core.isanything.IsAnything object>,
                                  query=<hamcrest.core.core.isanything.IsAnything object>,
                                  headers=<hamcrest.core.core.isanything.IsAnything object>,
                                  body=<hamcrest.core.core.isanything.IsAnything object>,
                                  times=<hamcrest.core.core.isanything.IsAnything object>)
```

Mountebank server has recorded call matching

### Parameters

- **method** (`Union[str, Matcher[str]]`) – Request’s method matched...

- **path** (`Union[furl, str, Matcher[Union[furl, str]]]`) – Request’s path matched...
- **query** (`Union[Mapping[str, str], Matcher[Mapping[str, str]]]`) – Request’s query matched...
- **headers** (`Union[Mapping[str, str], Matcher[Mapping[str, str]]]`) – Request’s headers matched...
- **body** (`Union[str, Matcher[str]]`) – Request’s body matched...
- **times** (`Union[int, Matcher[int]]`) – Request’s number of times called matched...

**describe\_to** (*description*)

Generates a description of the object.

The description may be part of a description of a larger object of which this is just a component, so it should be worded appropriately.

**Parameters** **description** (`Description`) – The description to be built or appended to.

**Return type** `None`

**static append\_matcher\_description** (*field\_matcher, field\_name, description*)

**Return type** `None`

**describe\_mismatch** (*server, description*)

Generates a description of why the matcher has not accepted the item.

The description will be part of a larger description of why a matching failed, so it should be concise.

This method assumes that `matches(item)` is `False`, but will not check this.

**Parameters**

- **item** – The item that the `Matcher` has rejected.
- **mismatch\_description** – The description to be built or appended to.

**Return type** `None`

**with\_method** (*method*)

**and\_method** (*method*)

**with\_path** (*path*)

**and\_path** (*path*)

**with\_query** (*query*)

**and\_query** (*query*)

**with\_headers** (*headers*)

**and\_headers** (*headers*)

**with\_body** (*body*)

**and\_body** (*body*)

**with\_times** (*times*)

**and\_times** (*times*)

```
mbtest.matchers.email_sent (to=<hamcrest.core.core.isanything.IsAnything object>, sub-
                             ject=<hamcrest.core.core.isanything.IsAnything object>,
                             body_text=<hamcrest.core.core.isanything.IsAnything object>)
```

Mountebank SMTP server was asked to sent email matching:

#### Parameters

- **to** (`Union[str, Matcher[str]]`) – Email’s to field matched...
- **subject** (`Union[str, Matcher[str]]`) – Email’s subject field matched...
- **body\_text** (`Union[str, Matcher[str]]`) – Email’s body matched...

**Return type** `Matcher[MountebankServer]`

```
class mbtest.matchers.EmailSent (to=<hamcrest.core.core.isanything.IsAnything object>, sub-
                                ject=<hamcrest.core.core.isanything.IsAnything object>,
                                body_text=<hamcrest.core.core.isanything.IsAnything ob-
                                ject>)
```

Mountebank SMTP server was asked to sent email matching:

#### Parameters

- **to** (`Union[str, Matcher[str]]`) – Email’s to field matched...
- **subject** (`Union[str, Matcher[str]]`) – Email’s subject field matched...
- **body\_text** (`Union[str, Matcher[str]]`) – Email’s body matched...

**describe\_to** (*description*)

Generates a description of the object.

The description may be part of a description of a larger object of which this is just a component, so it should be worded appropriately.

**Parameters** **description** (`Description`) – The description to be built or appended to.

**Return type** `None`

**describe\_mismatch** (*server*, *description*)

Generates a description of why the matcher has not accepted the item.

The description will be part of a larger description of why a matching failed, so it should be concise.

This method assumes that `matches(item)` is `False`, but will not check this.

#### Parameters

- **item** – The item that the `Matcher` has rejected.
- **mismatch\_description** – The description to be built or appended to.

**Return type** `None`

## 1.10 The *mbtest.imposters.base* module

```
class mbtest.imposters.base.JsonSerializable
```

Object capable of being converted to a JSON serializable structure (using `as_structure()`) or from such a structure ((using `from_structure()`)).

**abstract as\_structure** ()

Converted to a JSON serializable structure.

**Return type** `Any`

**Returns** Structure suitable for JSON serialisation.

**abstract static from\_structure** (*structure*)

Converted from a JSON serializable structure.

**Parameters** **structure** (*Any*) – JSON structure to be converted.

**Return type** *JsonSerializable*

**Returns** Converted object.

## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`





## INSTALLATION

Install from [Pypi](#) as usual, using `pip`, `tox`, or `setup.py`.

Also requires [Mountebank](#) to have been installed:

```
$ npm install mountebank@2.2 --production
```



## USAGE

A basic example:

```
import requests
from hamcrest import assert_that
from brunns.matchers.response import is_response
from mbtest.matchers import had_request
from mbtest.imposters import Imposter, Predicate, Response, Stub

def test_request_to_mock_server(mock_server):
    # Set up mock server with required behavior
    imposter = Imposter(Stub(Predicate(path="/test"),
                             Response(body="sausages")))

    with mock_server(imposter) as server:
        # Make request to mock server - exercise code under test here
        response = requests.get("{} /test".format(imposter.url))

        assert_that("We got the expected response",
                     response, is_response().with_status_code(200).and_body("sausages
↪"))
        assert_that("The mock server recorded the request",
                     server, had_request().with_path("/test").and_method("GET"))
```

Needs a pytest fixture, most easily defined in `conftest.py`:

```
import pytest
from mbtest import server

@pytest.fixture(scope="session")
def mock_server(request):
    return server.mock_server(request)
```



## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### m

- `mbtest.imposters.base`, [17](#)
- `mbtest.imposters.behaviors.copy`, [12](#)
- `mbtest.imposters.behaviors.lookup`, [12](#)
- `mbtest.imposters.behaviors.using`, [13](#)
- `mbtest.imposters.imposters`, [6](#)
- `mbtest.imposters.predicates`, [7](#)
- `mbtest.imposters.responses`, [10](#)
- `mbtest.imposters.stubs`, [7](#)
- `mbtest.matchers`, [15](#)
- `mbtest.server`, [3](#)





## INDEX

### A

`add_imposters()` (*mbtest.server.MountebankServer* method), 4  
`ALWAYS` (*mbtest.imposters.responses.Proxy.Mode* attribute), 11  
`and_body()` (*mbtest.matchers.HadRequest* method), 16  
`and_headers()` (*mbtest.matchers.HadRequest* method), 16  
`and_method()` (*mbtest.matchers.HadRequest* method), 16  
`and_path()` (*mbtest.matchers.HadRequest* method), 16  
`and_query()` (*mbtest.matchers.HadRequest* method), 16  
`and_times()` (*mbtest.matchers.HadRequest* method), 16  
`AndPredicate` (class in *mbtest.imposters.predicates*), 8  
`append_matcher_description()` (*mbtest.matchers.HadRequest* static method), 16  
`as_structure()` (*mbtest.imposters.base.JsonSerializable* method), 17  
`as_structure()` (*mbtest.imposters.behaviors.copy.Copy* method), 12  
`as_structure()` (*mbtest.imposters.behaviors.lookup.Key* method), 13  
`as_structure()` (*mbtest.imposters.behaviors.lookup.Lookup* method), 12  
`as_structure()` (*mbtest.imposters.behaviors.using.Using* method), 13  
`as_structure()` (*mbtest.imposters.behaviors.using.UsingRegex* method), 14  
`as_structure()` (*mbtest.imposters.behaviors.using.UsingXPath* method), 14  
`as_structure()` (*mbtest.imposters.imposters.Imposter* method), 6  
`as_structure()` (*mbtest.imposters.predicates.AndPredicate* method), 8  
`as_structure()` (*mbtest.imposters.predicates.InjectionPredicate* method), 9

`as_structure()` (*mbtest.imposters.predicates.OrPredicate* method), 9  
`as_structure()` (*mbtest.imposters.predicates.Predicate* method), 8  
`as_structure()` (*mbtest.imposters.predicates.TcpPredicate* method), 9  
`as_structure()` (*mbtest.imposters.responses.InjectionResponse* method), 12  
`as_structure()` (*mbtest.imposters.responses.Proxy* method), 11  
`as_structure()` (*mbtest.imposters.responses.Response* method), 10  
`as_structure()` (*mbtest.imposters.responses.TcpResponse* method), 11  
`as_structure()` (*mbtest.imposters.stubs.Stub* method), 7

### B

`BasePredicate` (class in *mbtest.imposters.predicates*), 7  
`BaseResponse` (class in *mbtest.imposters.responses*), 10  
`BINARY` (*mbtest.imposters.responses.Response.Mode* attribute), 10  
`body()` (*mbtest.imposters.responses.Response* property), 10

### C

`close()` (*mbtest.server.ExecutingMountebankServer* method), 5  
`CONTAINS` (*mbtest.imposters.predicates.Predicate.Operator* attribute), 8  
`Copy` (class in *mbtest.imposters.behaviors.copy*), 12

### D

`DEEP_EQUALS` (*mbtest.imposters.predicates.Predicate.Operator* attribute), 8  
`DELETE` (*mbtest.imposters.predicates.Predicate.Method* attribute), 8  
`delete_imposters()` (*mbtest.server.MountebankServer* method), 4

[describe\\_mismatch\(\)](#) (*mbtest.matchers.EmailSent method*), 17  
[describe\\_mismatch\(\)](#) (*mbtest.matchers.HadRequest method*), 16  
[describe\\_to\(\)](#) (*mbtest.matchers.EmailSent method*), 17  
[describe\\_to\(\)](#) (*mbtest.matchers.HadRequest method*), 16

## E

[email\\_sent\(\)](#) (*in module mbtest.matchers*), 16  
[EmailSent](#) (*class in mbtest.matchers*), 17  
[ENDS\\_WITH](#) (*mbtest.imposters.predicates.Predicate.Operator attribute*), 8  
[EQUALS](#) (*mbtest.imposters.predicates.Predicate.Operator attribute*), 8  
[ExecutingMountebankServer](#) (*class in mbtest.server*), 4  
[EXISTS](#) (*mbtest.imposters.predicates.Predicate.Operator attribute*), 8

## F

[from\\_structure\(\)](#) (*mbtest.imposters.base.JsonSerializable static method*), 18  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.copy.Copy static method*), 12  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.lookup.Key static method*), 13  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.lookup.Lookup static method*), 13  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.using.Using static method*), 14  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.using.UsingJsonPath static method*), 15  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.using.UsingRegex static method*), 14  
[from\\_structure\(\)](#) (*mbtest.imposters.behaviors.using.UsingXPath static method*), 14  
[from\\_structure\(\)](#) (*mbtest.imposters.imposters.Imposter static method*), 6  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.AndPredicate static method*), 9  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.BasePredicate static method*), 7  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.InjectionPredicate static method*), 9  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.OrPredicate static method*), 9  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.Predicate static method*), 8  
[from\\_structure\(\)](#) (*mbtest.imposters.predicates.TcpPredicate static method*), 9  
[from\\_structure\(\)](#) (*mbtest.imposters.responses.BaseResponse static method*), 10

[from\\_structure\(\)](#) (*mbtest.imposters.responses.InjectionResponse static method*), 12  
[from\\_structure\(\)](#) (*mbtest.imposters.responses.Proxy static method*), 11  
[from\\_structure\(\)](#) (*mbtest.imposters.responses.Response static method*), 11  
[from\\_structure\(\)](#) (*mbtest.imposters.responses.TcpResponse static method*), 11  
[from\\_structure\(\)](#) (*mbtest.imposters.stubs.Stub static method*), 7

## G

[GET](#) (*mbtest.imposters.predicates.Predicate.Method attribute*), 8  
[get\\_actual\\_requests\(\)](#) (*mbtest.server.MountebankServer method*), 4

## H

[had\\_request\(\)](#) (*in module mbtest.matchers*), 15  
[HadRequest](#) (*class in mbtest.matchers*), 15  
[has\\_value\(\)](#) (*mbtest.imposters.predicates.Predicate.Operator static method*), 8  
[HEAD](#) (*mbtest.imposters.predicates.Predicate.Method attribute*), 8  
[HTTP](#) (*mbtest.imposters.imposters.Imposter.Protocol attribute*), 6  
[HTTPS](#) (*mbtest.imposters.imposters.Imposter.Protocol attribute*), 6

## I

[Imposter](#) (*class in mbtest.imposters.imposters*), 6  
[InjectionProtocol](#) (*class in mbtest.imposters.imposters*), 6  
[inject\\_url\(\)](#) (*mbtest.server.MountebankServer method*), 4  
[InjectionPredicate](#) (*class in mbtest.imposters.predicates*), 9  
[InjectionResponse](#) (*class in mbtest.imposters.responses*), 11

## J

[JSONPATH](#) (*mbtest.imposters.behaviors.using.Using.Method attribute*), 13  
[JsonSerializable](#) (*class in mbtest.imposters.base*), 17

## K

[Key](#) (*class in mbtest.imposters.behaviors.lookup*), 13

## L

[LogicallyCombinablePredicate](#) (*class in mbtest.imposters.predicates*), 7

Lookup (class in *mbtest.imposters.behaviors.lookup*), 12

## M

MATCHES (*mbtest.imposters.predicates.Predicate.Operator* attribute), 8

*mbtest.imposters.base*  
module, 17

*mbtest.imposters.behaviors.copy*  
module, 12

*mbtest.imposters.behaviors.lookup*  
module, 12

*mbtest.imposters.behaviors.using*  
module, 13

*mbtest.imposters.imposters*  
module, 6

*mbtest.imposters.predicates*  
module, 7

*mbtest.imposters.responses*  
module, 10

*mbtest.imposters.stubs*  
module, 7

*mbtest.matchers*  
module, 15

*mbtest.server*  
module, 3

*mock\_server()* (in module *mbtest.server*), 3

module

*mbtest.imposters.base*, 17

*mbtest.imposters.behaviors.copy*, 12

*mbtest.imposters.behaviors.lookup*,  
12

*mbtest.imposters.behaviors.using*, 13

*mbtest.imposters.imposters*, 6

*mbtest.imposters.predicates*, 7

*mbtest.imposters.responses*, 10

*mbtest.imposters.stubs*, 7

*mbtest.matchers*, 15

*mbtest.server*, 3

*MountebankException*, 5

*MountebankPortInUseException*, 5

*MountebankServer* (class in *mbtest.server*), 4

*MountebankTimeoutError*, 5

## O

ONCE (*mbtest.imposters.responses.Proxy.Mode* attribute), 11

*OrPredicate* (class in *mbtest.imposters.predicates*), 9

## P

POST (*mbtest.imposters.predicates.Predicate.Method* attribute), 8

*Predicate* (class in *mbtest.imposters.predicates*), 7

*Predicate.InvalidPredicateOperator*, 8

*Predicate.Method* (class in *mbtest.imposters.predicates*), 8

*Predicate.Operator* (class in *mbtest.imposters.predicates*), 8

*Proxy* (class in *mbtest.imposters.responses*), 11

*Proxy.Mode* (class in *mbtest.imposters.responses*), 11

PUT (*mbtest.imposters.predicates.Predicate.Method* attribute), 8

## Q

*query\_all\_impostors()*  
(*mbtest.server.MountebankServer* method),  
4

## R

REGEX (*mbtest.imposters.behaviors.using.Using.Method* attribute), 13

*Response* (class in *mbtest.imposters.responses*), 10

*Response.Mode* (class in *mbtest.imposters.responses*),  
10

*running* (*mbtest.server.ExecutingMountebankServer* attribute), 5

## S

*server\_url()* (*mbtest.server.MountebankServer* property), 4

SMTP (*mbtest.imposters.imposters.Imposter.Protocol* attribute), 6

*smtp\_imposter()* (in module  
*mbtest.imposters.imposters*), 6

*start\_lock* (*mbtest.server.ExecutingMountebankServer* attribute), 5

STARTS\_WITH (*mbtest.imposters.predicates.Predicate.Operator* attribute), 8

*Stub* (class in *mbtest.imposters.stubs*), 7

## T

TCP (*mbtest.imposters.imposters.Imposter.Protocol* attribute), 6

*TcpPredicate* (class in *mbtest.imposters.predicates*),  
9

*TcpResponse* (class in *mbtest.imposters.responses*), 11

TEXT (*mbtest.imposters.responses.Response.Mode* attribute), 10

TRANSPARENT (*mbtest.imposters.responses.Proxy.Mode* attribute), 11

## U

*url()* (*mbtest.imposters.imposters.Imposter* property),  
6

*Using* (class in *mbtest.imposters.behaviors.using*), 13

*Using.Method* (class in  
*mbtest.imposters.behaviors.using*), 13

UsingJsonpath (class in  
    *mbtest.imposters.behaviors.using*), 14  
UsingRegex (class in  
    *mbtest.imposters.behaviors.using*), 14  
UsingXPath (class in  
    *mbtest.imposters.behaviors.using*), 14

## W

with\_body() (*mbtest.matchers.HadRequest* method),  
    16  
with\_headers() (*mbtest.matchers.HadRequest*  
    *method*), 16  
with\_method() (*mbtest.matchers.HadRequest*  
    *method*), 16  
with\_path() (*mbtest.matchers.HadRequest* method),  
    16  
with\_query() (*mbtest.matchers.HadRequest*  
    *method*), 16  
with\_times() (*mbtest.matchers.HadRequest*  
    *method*), 16

## X

XPATH (*mbtest.imposters.behaviors.using.Using.Method*  
    *attribute*), 13