

---

# **FaunaDB Documentation**

***Release 2.11.0***

**Fauna, Inc.**

**Jan 21, 2020**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Compatibility</b>	<b>5</b>
<b>3</b>	<b>Documentation</b>	<b>7</b>
<b>4</b>	<b>Basic Usage</b>	<b>9</b>
<b>5</b>	<b>Building it yourself</b>	<b>11</b>
5.1	Setup . . . . .	11
5.2	Testing . . . . .	11
5.3	Coverage . . . . .	11
5.4	Documenting . . . . .	11
<b>6</b>	<b>Contribute</b>	<b>13</b>
<b>7</b>	<b>License</b>	<b>15</b>
7.1	FaunaClient . . . . .	15
7.2	Query . . . . .	16
7.3	Objects . . . . .	24
7.4	Errors . . . . .	25
7.5	Client Logger . . . . .	26
7.6	RequestResult . . . . .	26
7.7	Page . . . . .	27
	<b>Python Module Index</b>	<b>29</b>
	<b>Index</b>	<b>31</b>



Python driver for [FaunaDB](#).



# CHAPTER 1

---

## Installation

---

```
$ pip install faunadb
```





## CHAPTER 2

---

### Compatibility

---

The following versions of Python are supported:

- Python 2.7
- Python 3.3
- Python 3.4
- Python 3.5
- Python 3.6



## CHAPTER 3

---

### Documentation

---

Driver documentation is available at <https://faunadb.readthedocs.org/en/latest/>.

See the [FaunaDB Documentation](#) for a complete API reference, or look in [tests](#) for more examples.



## CHAPTER 4

---

### Basic Usage

---

```
from faunadb import query as q
from faunadb.objects import Ref
from faunadb.client import FaunaClient

client = FaunaClient(secret="your-secret-here")

indexes = client.query(q.paginate(q.indexes()))

print(indexes)
```



### 5.1 Setup

```
$ virtualenv venv
$ source venv/bin/activate
$ pip install .
```

### 5.2 Testing

To run the tests you must have a FaunaDB database available. Then set the environment variable `FAUNA_ROOT_KEY` to your database's root key. If you use FaunaDB cloud, this is the password you log in with.

Then run `make test`. To test a single test, use e.g. `python -m unittest tests.test_client.ClientTest.test_ping`.

Tests can also be run via a Docker container with `FAUNA_ROOT_KEY="your-cloud-secret"` `make docker-test` (an alternate Alpine-based Python image can be provided via `RUNTIME_IMAGE`).

### 5.3 Coverage

To run the tests with coverage, install the coverage dependencies with `pip install .[coverage]`, and then run `make coverage`. A summary will be displayed to the terminal, and a detailed coverage report will be available at `htmlcov/index.html`.

### 5.4 Documenting

Run `pip install .[doc]` to install the needed packages to generate the docs. Then run `make doc`, then open `docs/_build/html/index.html` in a web browser.





## CHAPTER 6

---

### Contribute

---

GitHub pull requests are very welcome.



Copyright 2019 Fauna, Inc.

Licensed under the Mozilla Public License, Version 2.0 (the “License”); you may not use this software except in compliance with the License. You may obtain a copy of the License at

<http://mozilla.org/MPL/2.0/>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## 7.1 FaunaClient

```
class faunadb.client.FaunaClient (secret, domain='db.fauna.com', scheme='https', port=None,
                                timeout=60, observer=None, pool_connections=10,
                                pool_maxsize=10, **kwargs)
```

Directly communicates with FaunaDB via JSON.

For data sent to the server, the `to_fauna_json` method will be called on any values. It is encouraged to pass e.g. `Ref` objects instead of raw JSON data.

All methods return a converted JSON response. This is a dict containing lists, ints, floats, strings, and other dicts. Any `Ref`, `SetRef`, `FaunaTime`, or `datetime.date` values in it will also be parsed. (So instead of `{"@ref": {"id": "frogs", "class": {"@ref": {"id": "classes"}}`}, you will get `Ref("frogs", Native.CLASSES)`.)

```
__init__ (secret, domain='db.fauna.com', scheme='https', port=None, timeout=60, observer=None,
          pool_connections=10, pool_maxsize=10, **kwargs)
```

### Parameters

- **secret** – Auth token for the FaunaDB server.
- **domain** – Base URL for the FaunaDB server.
- **scheme** – "http" or "https".

- **port** – Port of the FaunaDB server.
- **timeout** – Read timeout in seconds.
- **observer** – Callback that will be passed a *RequestResult* after every completed request.
- **pool\_connections** – The number of connection pools to cache.
- **pool\_maxsize** – The maximum number of connections to save in the pool.

**get\_last\_txn\_time** ()

Get the freshest timestamp reported to this client. :return:

**new\_session\_client** (*secret, observer=None*)

Create a new client from the existing config with a given secret. The returned client share its parent underlying resources.

#### Parameters

- **secret** – Credentials to use when sending requests.
- **observer** – Callback that will be passed a *RequestResult* after every completed request.

#### Returns

**ping** (*scope=None, timeout=None*)

Ping FaunaDB.

**query** (*expression*)

Use the FaunaDB query API.

**Parameters** **expression** – A query. See *Query* for information on queries.

**Returns** Converted JSON response.

**sync\_last\_txn\_time** (*new\_txn\_time*)

Sync the freshest timestamp seen by this client.

This has no effect if staler than currently stored timestamp. WARNING: This should be used only when coordinating timestamps across

multiple clients. Moving the timestamp arbitrarily forward into the future will cause transactions to stall.

**Parameters** **new\_txn\_time** – the new seen transaction time.

## 7.2 Query

This module contains functions that generate query objects, suitable to be passed into *FaunaClient.query*.

In addition to these functions, queries may contain:

- None
- Numbers (1, 2.5)
- Strings ("foo")
- Lists ([1, "foo"])
- Dicts ({"foo": "bar"})

- A *Ref*, *SetRef*, *FaunaTime*, or `datetime.date`.

For example: `select ("a", {"a": Ref("123", Ref("widgets", Native.CLASSES))})`.

`faunadb.query.abort` (*msg*)

See the docs.

`faunadb.query.add` (*\*numbers*)

See the docs.

`faunadb.query.all` (*collection*)

See the docs.

`faunadb.query.and_` (*\*booleans*)

See the docs.

`faunadb.query.and_expr` (*\*args*, *\*\*kwargs*)

See the docs.

`faunadb.query.any` (*collection*)

See the docs.

`faunadb.query.append` (*elements*, *collection*)

See the docs.

`faunadb.query.at` (*timestamp*, *expr*)

See the docs.

`faunadb.query.call` (*ref\_*, *\*arguments*)

See the docs.

`faunadb.query.casefold` (*string*, *normalizer=None*)

See the docs.

`faunadb.query.class_` (*\*args*, *\*\*kwargs*)

See the docs.

`faunadb.query.class_expr` (*\*args*, *\*\*kwargs*)

See the docs.

`faunadb.query.classes` (*\*args*, *\*\*kwargs*)

See the docs.

`faunadb.query.collection` (*collection\_name*, *scope=None*)

See the docs.

`faunadb.query.collections` (*scope=None*)

See the docs.

`faunadb.query.concat` (*strings*, *separator=None*)

See the docs.

`faunadb.query.contains` (*path*, *in\_*)

See the docs.

`faunadb.query.contains_str` (*value*, *search*)

See the docs.

`faunadb.query.contains_str_regex` (*value*, *pattern*)

See the docs.

`faunadb.query.count` (*collection*)

See the docs.

`faunadb.query.create` (*collection\_ref*, *params*)  
See the docs.

`faunadb.query.create_class` (*\*args*, *\*\*kwargs*)  
See the docs.

`faunadb.query.create_collection` (*collection\_params*)  
See the docs.

`faunadb.query.create_database` (*db\_params*)  
See the docs.

`faunadb.query.create_function` (*func\_params*)  
See the docs.

`faunadb.query.create_index` (*index\_params*)  
See the docs.

`faunadb.query.create_key` (*key\_params*)  
See the docs.

`faunadb.query.create_role` (*func\_params*)  
See the docs.

`faunadb.query.credentials` (*scope=None*)  
See the docs.

`faunadb.query.database` (*db\_name*, *scope=None*)  
See the docs.

`faunadb.query.databases` (*scope=None*)  
See the docs.

`faunadb.query.date` (*string*)  
See the docs.

`faunadb.query.day_of_month` (*expr*)  
See the docs.

`faunadb.query.day_of_week` (*expr*)  
See the docs.

`faunadb.query.day_of_year` (*expr*)  
See the docs.

`faunadb.query.delete` (*ref\_*)  
See the docs.

`faunadb.query.difference` (*\*sets*)  
See the docs.

`faunadb.query.distinct` (*set*)  
See the docs.

`faunadb.query.divide` (*\*numbers*)  
See the docs.

`faunadb.query.do` (*\*expressions*)  
See the docs.

`faunadb.query.documents` (*collections*)  
See the docs.

`faunadb.query.drop` (*number, collection*)  
See the docs.

`faunadb.query.ends_with` (*value, search*)  
See the docs.

`faunadb.query.epoch` (*number, unit*)  
See the docs.

`faunadb.query.equals` (*\*values*)  
See the docs.

`faunadb.query.events` (*ref\_set*)  
See the docs.

`faunadb.query.exists` (*ref\_, ts=None*)  
See the docs.

`faunadb.query.filter_` (*expr, collection*)  
See the docs.

`faunadb.query.filter_expr` (*\*args, \*\*kwargs*)  
See the docs.

`faunadb.query.find_str` (*value, find, start=None*)  
See the docs.

`faunadb.query.find_str_regex` (*value, pattern, start=None, numResults=None*)  
See the docs.

`faunadb.query.foreach` (*expr, collection*)  
See the docs.

`faunadb.query.format` (*string, \*values*)  
See the docs.

`faunadb.query.function` (*fn\_name, scope=None*)  
See the docs.

`faunadb.query.functions` (*scope=None*)  
See the docs.

`faunadb.query.get` (*ref\_, ts=None*)  
See the docs.

`faunadb.query.gt` (*\*values*)  
See the docs.

`faunadb.query.gte` (*\*values*)  
See the docs.

`faunadb.query.has_identity` ()  
See the docs.

`faunadb.query.hour` (*expr*)  
See the docs.

`faunadb.query.identify` (*ref\_, password*)  
See the docs.

`faunadb.query.identity` ()  
See the docs.

`faunadb.query.if_(condition, then, else_)`  
 See the docs.

`faunadb.query.if_expr(*args, **kwargs)`  
 See the docs.

`faunadb.query.index(index_name, scope=None)`  
 See the docs.

`faunadb.query.indexes(scope=None)`  
 See the docs.

`faunadb.query.insert(ref_, ts, action, params)`  
 See the docs.

`faunadb.query.intersection(*sets)`  
 See the docs.

`faunadb.query.is_empty(collection)`  
 See the docs.

`faunadb.query.is_nonempty(collection)`  
 See the docs.

`faunadb.query.join(source, target)`  
 See the docs.

`faunadb.query.key_from_secret(secret)`  
 See the docs.

`faunadb.query.keys(scope=None)`  
 See the docs.

`faunadb.query.lambda_(var_name_or_pattern, expr)`  
 See the docs.

`faunadb.query.lambda_expr(*args, **kwargs)`  
 See the docs.

`faunadb.query.lambda_query(func)`  
 See the docs. This form generates *var* objects for you, and is called like:

```
query.lambda_query(lambda a: query.add(a, a))
# Produces: {
#   "lambda": "a",
#   "expr": {"add": [{"var": "a"}, {"var": "a"}]}
# }
```

You usually don't need to call this directly as lambdas in queries will be converted for you. For example:  
`query.map_(lambda a: query.add(a, 1), collection).`

You can also use `lambda_` directly.

**Parameters func** – Takes one or more *var* expressions and uses them to construct an expression. If this has more than one argument, the lambda destructures an array argument. (To destructure single-element arrays use `lambda_.`)

`faunadb.query.length(value)`  
 See the docs.

`faunadb.query.let(*args, **kwargs)`  
 See the docs.



---

`faunadb.query.login` (*ref\_*, *params*)  
See the docs.

`faunadb.query.logout` (*delete\_tokens*)  
See the docs.

`faunadb.query.lowercase` (*value*)  
See the docs.

`faunadb.query.lt` (*\*values*)  
See the docs.

`faunadb.query.lte` (*\*values*)  
See the docs.

`faunadb.query.ltrim` (*value*)  
See the docs.

`faunadb.query.map_` (*expr*, *collection*)  
See the docs.

`faunadb.query.map_expr` (*\*args*, *\*\*kwargs*)  
See the docs.

`faunadb.query.match` (*index*, *\*terms*)  
See the docs.

`faunadb.query.mean` (*collection*)  
See the docs.

`faunadb.query.merge` (*merge*, *with\_*, *lambda\_=None*)  
See the docs.

`faunadb.query.minute` (*expr*)  
See the docs.

`faunadb.query.modulo` (*\*numbers*)  
See the docs.

`faunadb.query.month` (*expr*)  
See the docs.

`faunadb.query.move_database` (*from\_*, *to*)  
See the docs.

`faunadb.query.multiply` (*\*numbers*)  
See the docs.

`faunadb.query.new_id` ()  
See the docs.

`faunadb.query.next_id` (*\*args*, *\*\*kwargs*)  
See the docs.

`faunadb.query.ngram` (*terms*, *min=None*, *max=None*)  
See the docs.

`faunadb.query.not_` (*boolean*)  
See the docs.

`faunadb.query.not_expr` (*\*args*, *\*\*kwargs*)  
See the docs.

`faunadb.query.now()`

See the docs.

`faunadb.query.or_(*booleans)`

See the docs.

`faunadb.query.or_expr(*args, **kwargs)`

See the docs.

`faunadb.query.paginate(set, size=None, ts=None, after=None, before=None, events=None, sources=None)`

See the docs. You may want to convert the result of this to a *Page*.

`faunadb.query.prepend(elements, collection)`

See the docs.

`faunadb.query.query(_lambda)`

See the docs.

`faunadb.query.range(set, from_, to)`

See the docs.

`faunadb.query.reduce(lambda_, initial, collection)`

See the docs.

`faunadb.query.ref(collection_ref, id=None)`

See the docs.

`faunadb.query.regex_escape(value)`

See the docs.

`faunadb.query.remove(ref_, ts, action)`

See the docs.

`faunadb.query.repeat(value, number=None)`

See the docs.

`faunadb.query.replace(ref_, params)`

See the docs.

`faunadb.query.replace_str(value, find, replace)`

See the docs.

`faunadb.query.replace_str_regex(value, pattern, replace, first=None)`

See the docs.

`faunadb.query.role(role_name, scope=None)`

See the docs.

`faunadb.query.roles(scope=None)`

See the docs.

`faunadb.query.rtrim(value)`

See the docs.

`faunadb.query.second(expr)`

See the docs.

`faunadb.query.select(path, from_)`

See the docs. See also `select_with_default()`.

`faunadb.query.select_all(path, from_)`

See the docs.

`faunadb.query.select_with_default` (*path, from\_, default*)  
See the docs.

`faunadb.query.singleton` (*ref\_*)  
See the docs.

`faunadb.query.space` (*count*)  
See the docs.

`faunadb.query.starts_with` (*value, search*)  
See the docs.

`faunadb.query.substring` (*value, start, length=None*)  
See the docs.

`faunadb.query.subtract` (*\*numbers*)  
See the docs.

`faunadb.query.sum` (*collection*)  
See the docs.

`faunadb.query.take` (*number, collection*)  
See the docs.

`faunadb.query.time` (*string*)  
See the docs.

`faunadb.query.time_add` (*base, offset, unit*)  
See the docs.

`faunadb.query.time_diff` (*start, finish, unit*)  
See the docs.

`faunadb.query.time_subtract` (*base, offset, unit*)  
See the docs.

`faunadb.query.titlecase` (*value*)  
See the docs.

`faunadb.query.to_date` (*expr*)  
See the docs.

`faunadb.query.to_micros` (*expr*)  
See the docs.

`faunadb.query.to_millis` (*expr*)  
See the docs.

`faunadb.query.to_number` (*expr*)  
See the docs.

`faunadb.query.to_seconds` (*expr*)  
See the docs.

`faunadb.query.to_time` (*expr*)  
See the docs.

`faunadb.query.tokens` (*scope=None*)  
See the docs.

`faunadb.query.trim` (*value*)  
See the docs.

`faunadb.query.union` (\*sets)

See the [docs](#).

`faunadb.query.update` (ref\_, params)

See the [docs](#).

`faunadb.query.uppercase` (value)

See the [docs](#).

`faunadb.query.var` (var\_name)

See the [docs](#).

`faunadb.query.year` (expr)

See the [docs](#).

## 7.3 Objects

Types used in queries and responses. See the [docs](#).

**class** `faunadb.objects.FaunaTime` (value)

FaunaDB time. See the [docs](#).

For dates, regular `datetime.date` objects are used.

`__init__` (value)

**Parameters value** – If a `datetime.datetime` is passed, it is converted to a string. Must include an offset.

`to_datetime` ()

Convert to an offset-aware datetime object. This is lossy as datetimes have microsecond rather than nanosecond precision.

**class** `faunadb.objects.Query` (value)

Represents a `@query` type in FaunaDB. See the [docs](#).

**class** `faunadb.objects.Ref` (id, cls=None, db=None)

FaunaDB ref. See the [docs](#).

A simple wrapper around a string which can be extracted using `ref.value`. Queries that require a Ref will not work if you just pass in a string.

`__init__` (id, cls=None, db=None)

`x.__init__(...)` initializes x; see `help(type(x))` for signature

`class_` (\*\*kwargs)

Gets the class part out of the Ref.

`collection` ()

Gets the collection part out of the Ref.

`database` ()

Gets the database part out of the Ref.

`id` ()

Gets the id part out of the Ref.

**class** `faunadb.objects.SetRef` (set\_ref)

FaunaDB Set. This represents a set returned as part of a response. For query sets see [Query](#).

`__init__` (set\_ref)

`x.__init__(...)` initializes x; see `help(type(x))` for signature

## 7.4 Errors

Error types that methods in the FaunaDB client throw.

**exception** `faunadb.errors.BadRequest` (*request\_result*)  
HTTP 400 error.

**class** `faunadb.errors.ErrorData` (*code, description, position, failures*)  
Data for one error returned by the server.

`__init__` (*code, description, position, failures*)  
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

**code** = `None`  
Error code. See all error codes [here](#).

**description** = `None`  
Error description.

**failures** = `None`  
List of all *Failure* objects returned by the server. `None` unless `code == "validation failed"`.

**position** = `None`  
Position of the error in a query. May be `None`.

**class** `faunadb.errors.Failure` (*code, description, field*)  
Part of the failures of an *ErrorData*. See the Invalid Data section of the [docs](#).

`__init__` (*code, description, field*)  
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

**code** = `None`  
Failure code.

**description** = `None`  
Failure description.

**field** = `None`  
Field of the failure in the instance.

**exception** `faunadb.errors.FaunaError` (*description, request\_result*)  
Error returned by the FaunaDB server. For documentation of error types, see the [docs](#).

`__init__` (*description, request\_result*)  
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

**request\_result** = `None`  
*RequestResult* for the request that caused this error.

**exception** `faunadb.errors.HttpError` (*request\_result*)

`__init__` (*request\_result*)  
`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

**errors** = `None`  
List of all *ErrorData* objects sent by the server.

**exception** `faunadb.errors.InternalError` (*request\_result*)  
HTTP 500 error.

**exception** `faunadb.errors.NotFound` (*request\_result*)  
HTTP 404 error.

**exception** `faunadb.errors.PermissionDenied(request_result)`  
HTTP 403 error.

**exception** `faunadb.errors.Unauthorized(request_result)`  
HTTP 401 error.

**exception** `faunadb.errors.UnavailableError(request_result)`  
HTTP 503 error.

**exception** `faunadb.errors.UnexpectedError(description, request_result)`  
Error for when the server returns an unexpected kind of response.

## 7.5 Client Logger

`faunadb.client_logger.logger(logger_func)`

Function that can be the observer for a `FaunaClient`. Will call `logger_func` on a string representation of each `RequestResult`.

Use it like:

```
def log(logged):
    print logged
client = FaunaClient(observer=logger(log), ...)
client.ping() # Calls `log`
```

**Parameters** `logger_func` – Callback taking a string to be logged.

`faunadb.client_logger.show_request_result(request_result)`  
Translates a `RequestResult` to a string suitable for logging.

## 7.6 RequestResult

**class** `faunadb.request_result.RequestResult(method, path, query, request_content, response_raw, response_content, status_code, response_headers, start_time, end_time)`

Stores information about a single request and response.

`__init__(method, path, query, request_content, response_raw, response_content, status_code, response_headers, start_time, end_time)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

**end\_time = None**

Time the response was received.

**method = None**

“GET” or “POST”.

**path = None**

Path that was queried. Relative to client’s domain.

**query = None**

URL query. None unless `method == GET`. *Not* related to `FaunaClient.query`.

**request\_content = None**

Request data.

**response\_content = None**

Parsed value returned by the server. Includes “resource” wrapper dict, or may be an “errors” dict instead. In the case of a JSON parse error, this will be None.

**response\_headers = None**

A dictionary of headers with case-insensitive keys.

**response\_raw = None**

String value returned by the server.

**start\_time = None**

Time the request started.

**status\_code = None**

HTTP status code.

**time\_taken**

`end_time - start_time`

## 7.7 Page

**class** `faunadb.page.Page` (*data*, *before=None*, *after=None*)

Represents a single pagination result. See `paginate` in the docs. You must convert to Page yourself using `from_raw()`.

**\_\_init\_\_** (*data*, *before=None*, *after=None*)

`x.__init__(...)` initializes x; see `help(type(x))` for signature

**after = None**

Optional *Ref* for an instance that comes after this page.

**before = None**

Optional *Ref* for an instance that comes before this page.

**data = None**

List of elements returned by the query.

**static from\_raw** (*raw*)

Convert a raw response dict to a Page.

**map\_data** (*func*)

Return a new Page whose data has had `func` applied to each element.

**static set\_iterator** (*client*, *set\_query*, *map\_lambda=None*, *mapper=None*, *page\_size=None*)

Iterator that keeps getting new pages of a set.

**Parameters**

- **map\_lambda** – If present, a *lambda\_* for mapping set elements.
- **mapper** – Mapping Python function used on each page element.
- **page\_size** – Number of instances to be fetched at a time.

**Returns** Iterator through all elements in the set.





**f**

`faunadb.client_logger`, 26  
`faunadb.errors`, 25  
`faunadb.objects`, 24  
`faunadb.page`, 27  
`faunadb.query`, 16  
`faunadb.request_result`, 26



## Symbols

\_\_init\_\_() (*faunadb.client.FaunaClient method*), 15  
 \_\_init\_\_() (*faunadb.errors.ErrorData method*), 25  
 \_\_init\_\_() (*faunadb.errors.Failure method*), 25  
 \_\_init\_\_() (*faunadb.errors.FaunaError method*), 25  
 \_\_init\_\_() (*faunadb.errors.HttpError method*), 25  
 \_\_init\_\_() (*faunadb.objects.FaunaTime method*), 24  
 \_\_init\_\_() (*faunadb.objects.Ref method*), 24  
 \_\_init\_\_() (*faunadb.objects.SetRef method*), 24  
 \_\_init\_\_() (*faunadb.page.Page method*), 27  
 \_\_init\_\_() (*faunadb.request\_result.RequestResult method*), 26

## A

abort() (*in module faunadb.query*), 17  
 add() (*in module faunadb.query*), 17  
 after (*faunadb.page.Page attribute*), 27  
 all() (*in module faunadb.query*), 17  
 and\_() (*in module faunadb.query*), 17  
 and\_expr() (*in module faunadb.query*), 17  
 any() (*in module faunadb.query*), 17  
 append() (*in module faunadb.query*), 17  
 at() (*in module faunadb.query*), 17

## B

BadRequest, 25  
 before (*faunadb.page.Page attribute*), 27

## C

call() (*in module faunadb.query*), 17  
 casefold() (*in module faunadb.query*), 17  
 class\_() (*faunadb.objects.Ref method*), 24  
 class\_() (*in module faunadb.query*), 17  
 class\_expr() (*in module faunadb.query*), 17  
 classes() (*in module faunadb.query*), 17  
 code (*faunadb.errors.ErrorData attribute*), 25  
 code (*faunadb.errors.Failure attribute*), 25  
 collection() (*faunadb.objects.Ref method*), 24  
 collection() (*in module faunadb.query*), 17

collections() (*in module faunadb.query*), 17  
 concat() (*in module faunadb.query*), 17  
 contains() (*in module faunadb.query*), 17  
 contains\_str() (*in module faunadb.query*), 17  
 contains\_str\_regex() (*in module faunadb.query*), 17  
 count() (*in module faunadb.query*), 17  
 create() (*in module faunadb.query*), 17  
 create\_class() (*in module faunadb.query*), 18  
 create\_collection() (*in module faunadb.query*), 18  
 create\_database() (*in module faunadb.query*), 18  
 create\_function() (*in module faunadb.query*), 18  
 create\_index() (*in module faunadb.query*), 18  
 create\_key() (*in module faunadb.query*), 18  
 create\_role() (*in module faunadb.query*), 18  
 credentials() (*in module faunadb.query*), 18

## D

data (*faunadb.page.Page attribute*), 27  
 database() (*faunadb.objects.Ref method*), 24  
 database() (*in module faunadb.query*), 18  
 databases() (*in module faunadb.query*), 18  
 date() (*in module faunadb.query*), 18  
 day\_of\_month() (*in module faunadb.query*), 18  
 day\_of\_week() (*in module faunadb.query*), 18  
 day\_of\_year() (*in module faunadb.query*), 18  
 delete() (*in module faunadb.query*), 18  
 description (*faunadb.errors.ErrorData attribute*), 25  
 description (*faunadb.errors.Failure attribute*), 25  
 difference() (*in module faunadb.query*), 18  
 distinct() (*in module faunadb.query*), 18  
 divide() (*in module faunadb.query*), 18  
 do() (*in module faunadb.query*), 18  
 documents() (*in module faunadb.query*), 18  
 drop() (*in module faunadb.query*), 18

## E

end\_time (*faunadb.request\_result.RequestResult attribute*), 26

`ends_with()` (in module `faunadb.query`), 19  
`epoch()` (in module `faunadb.query`), 19  
`equals()` (in module `faunadb.query`), 19  
`ErrorData` (class in `faunadb.errors`), 25  
`errors` (`faunadb.errors.HttpError` attribute), 25  
`events()` (in module `faunadb.query`), 19  
`exists()` (in module `faunadb.query`), 19

## F

`Failure` (class in `faunadb.errors`), 25  
`failures` (`faunadb.errors.ErrorData` attribute), 25  
`FaunaClient` (class in `faunadb.client`), 15  
`faunadb.client_logger` (module), 26  
`faunadb.errors` (module), 25  
`faunadb.objects` (module), 24  
`faunadb.page` (module), 27  
`faunadb.query` (module), 16  
`faunadb.request_result` (module), 26  
`FaunaError`, 25  
`FaunaTime` (class in `faunadb.objects`), 24  
`field` (`faunadb.errors.Failure` attribute), 25  
`filter_()` (in module `faunadb.query`), 19  
`filter_expr()` (in module `faunadb.query`), 19  
`find_str()` (in module `faunadb.query`), 19  
`find_str_regex()` (in module `faunadb.query`), 19  
`foreach()` (in module `faunadb.query`), 19  
`format()` (in module `faunadb.query`), 19  
`from_raw()` (`faunadb.page.Page` static method), 27  
`function()` (in module `faunadb.query`), 19  
`functions()` (in module `faunadb.query`), 19

## G

`get()` (in module `faunadb.query`), 19  
`get_last_txn_time()` (`faunadb.client.FaunaClient` method), 16  
`gt()` (in module `faunadb.query`), 19  
`gte()` (in module `faunadb.query`), 19

## H

`has_identity()` (in module `faunadb.query`), 19  
`hour()` (in module `faunadb.query`), 19  
`HttpError`, 25

## I

`id()` (`faunadb.objects.Ref` method), 24  
`identify()` (in module `faunadb.query`), 19  
`identity()` (in module `faunadb.query`), 19  
`if_()` (in module `faunadb.query`), 19  
`if_expr()` (in module `faunadb.query`), 20  
`index()` (in module `faunadb.query`), 20  
`indexes()` (in module `faunadb.query`), 20  
`insert()` (in module `faunadb.query`), 20  
`InternalServerError`, 25  
`intersection()` (in module `faunadb.query`), 20

`is_empty()` (in module `faunadb.query`), 20  
`is_nonempty()` (in module `faunadb.query`), 20

## J

`join()` (in module `faunadb.query`), 20

## K

`key_from_secret()` (in module `faunadb.query`), 20  
`keys()` (in module `faunadb.query`), 20

## L

`lambda_()` (in module `faunadb.query`), 20  
`lambda_expr()` (in module `faunadb.query`), 20  
`lambda_query()` (in module `faunadb.query`), 20  
`length()` (in module `faunadb.query`), 20  
`let()` (in module `faunadb.query`), 20  
`logger()` (in module `faunadb.client_logger`), 26  
`login()` (in module `faunadb.query`), 20  
`logout()` (in module `faunadb.query`), 21  
`lowercase()` (in module `faunadb.query`), 21  
`lt()` (in module `faunadb.query`), 21  
`lte()` (in module `faunadb.query`), 21  
`ltrim()` (in module `faunadb.query`), 21

## M

`map_()` (in module `faunadb.query`), 21  
`map_data()` (`faunadb.page.Page` method), 27  
`map_expr()` (in module `faunadb.query`), 21  
`match()` (in module `faunadb.query`), 21  
`mean()` (in module `faunadb.query`), 21  
`merge()` (in module `faunadb.query`), 21  
`method` (`faunadb.request_result.RequestResult` attribute), 26  
`minute()` (in module `faunadb.query`), 21  
`modulo()` (in module `faunadb.query`), 21  
`month()` (in module `faunadb.query`), 21  
`move_database()` (in module `faunadb.query`), 21  
`multiply()` (in module `faunadb.query`), 21

## N

`new_id()` (in module `faunadb.query`), 21  
`new_session_client()` (`faunadb.client.FaunaClient` method), 16  
`next_id()` (in module `faunadb.query`), 21  
`ngram()` (in module `faunadb.query`), 21  
`not_()` (in module `faunadb.query`), 21  
`not_expr()` (in module `faunadb.query`), 21  
`NotFound`, 25  
`now()` (in module `faunadb.query`), 21

## O

`or_()` (in module `faunadb.query`), 22  
`or_expr()` (in module `faunadb.query`), 22

## P

Page (class in *faunadb.page*), 27  
 paginate() (in module *faunadb.query*), 22  
 path (*faunadb.request\_result.RequestResult* attribute), 26  
 PermissionDenied, 25  
 ping() (*faunadb.client.FaunaClient* method), 16  
 position (*faunadb.errors.ErrorData* attribute), 25  
 prepend() (in module *faunadb.query*), 22

## Q

Query (class in *faunadb.objects*), 24  
 query (*faunadb.request\_result.RequestResult* attribute), 26  
 query() (*faunadb.client.FaunaClient* method), 16  
 query() (in module *faunadb.query*), 22

## R

range() (in module *faunadb.query*), 22  
 reduce() (in module *faunadb.query*), 22  
 Ref (class in *faunadb.objects*), 24  
 ref() (in module *faunadb.query*), 22  
 regex\_escape() (in module *faunadb.query*), 22  
 remove() (in module *faunadb.query*), 22  
 repeat() (in module *faunadb.query*), 22  
 replace() (in module *faunadb.query*), 22  
 replace\_str() (in module *faunadb.query*), 22  
 replace\_str\_regex() (in module *faunadb.query*), 22  
 request\_content (*faunadb.request\_result.RequestResult* attribute), 26  
 request\_result (*faunadb.errors.FaunaError* attribute), 25  
 RequestResult (class in *faunadb.request\_result*), 26  
 response\_content (*faunadb.request\_result.RequestResult* attribute), 26  
 response\_headers (*faunadb.request\_result.RequestResult* attribute), 27  
 response\_raw (*faunadb.request\_result.RequestResult* attribute), 27  
 role() (in module *faunadb.query*), 22  
 roles() (in module *faunadb.query*), 22  
 rtrim() (in module *faunadb.query*), 22

## S

second() (in module *faunadb.query*), 22  
 select() (in module *faunadb.query*), 22  
 select\_all() (in module *faunadb.query*), 22  
 select\_with\_default() (in module *faunadb.query*), 22

set\_iterator() (*faunadb.page.Page* static method), 27  
 SetRef (class in *faunadb.objects*), 24  
 show\_request\_result() (in module *faunadb.client\_logger*), 26  
 singleton() (in module *faunadb.query*), 23  
 space() (in module *faunadb.query*), 23  
 start\_time (*faunadb.request\_result.RequestResult* attribute), 27  
 starts\_with() (in module *faunadb.query*), 23  
 status\_code (*faunadb.request\_result.RequestResult* attribute), 27  
 substring() (in module *faunadb.query*), 23  
 subtract() (in module *faunadb.query*), 23  
 sum() (in module *faunadb.query*), 23  
 sync\_last\_txn\_time() (*faunadb.client.FaunaClient* method), 16

## T

take() (in module *faunadb.query*), 23  
 time() (in module *faunadb.query*), 23  
 time\_add() (in module *faunadb.query*), 23  
 time\_diff() (in module *faunadb.query*), 23  
 time\_subtract() (in module *faunadb.query*), 23  
 time\_taken (*faunadb.request\_result.RequestResult* attribute), 27  
 titlecase() (in module *faunadb.query*), 23  
 to\_date() (in module *faunadb.query*), 23  
 to\_datetime() (*faunadb.objects.FaunaTime* method), 24  
 to\_micros() (in module *faunadb.query*), 23  
 to\_millis() (in module *faunadb.query*), 23  
 to\_number() (in module *faunadb.query*), 23  
 to\_seconds() (in module *faunadb.query*), 23  
 to\_time() (in module *faunadb.query*), 23  
 tokens() (in module *faunadb.query*), 23  
 trim() (in module *faunadb.query*), 23

## U

Unauthorized, 26  
 UnavailableError, 26  
 UnexpectedError, 26  
 union() (in module *faunadb.query*), 23  
 update() (in module *faunadb.query*), 24  
 uppercase() (in module *faunadb.query*), 24

## V

var() (in module *faunadb.query*), 24

## Y

year() (in module *faunadb.query*), 24