

Bulk Operations

On this page

- [insert_one](#)
- [insert_many](#)
- [delete_one](#)
- [delete_many](#)
- [replace_one](#)
- [update_one](#)
- [update_many](#)

The bulk write API sends a list of write operations to the server in one method call. Execution order of the operations is guaranteed if you pass in the `ordered` option.

The `bulk_write` method takes three arguments:

- A list of operations.
- The `ordered` option with a boolean. Defaults to `true`.
- A write concern option. Defaults to the collection's write concern.

Valid bulk write operations are the following:

`insert_one`

```
{ :insert_one => { :x => 1 } }
```

`insert_many`

```
{ :insert_many => [ { :x => 1 }, { :x => 2 } ] }
```

delete_one

```
{ :delete_one => { :filter => { :x => 1 } } }
```

delete_many

```
{ :delete_many => { :filter => { :x => 1 } } }
```

replace_one

```
{ :replace_one => { :filter => { :x => 1 },  
                  :replacement => { :x => 2 },  
                  :upsert => true } # upsert is optional and defaults to false  
}
```

update_one

```
{ :update_one => { :filter => { :x => 1 },  
                  :update => { '$set' => { :x => 2 } },  
                  :upsert => true } # upsert is optional and defaults to false  
}
```

update_many

```

{ :update_many => { :filter => { :x => 1 },
  :update => { '$set' => { :x => 2 } },
  :upsert => true } # upsert is optional and defaults to false
}

```

The following example shows how to pass operations to the `bulk_write` method.

```

coll = client['documents']
coll.bulk_write([ { :insert_one => { :x => 1 }
  },
  { :update_one => { :filter => { :x => 1 },
    :update => { '$set' => { :x => 2 } }
  },
  { :replace_one => { :filter => { :x => 2 },
    :replacement => { :x => 3 }
  }
],
:ordered => true)

```