

# MockResultSet DSL

Mocking interactions with a database can be frustrating; however, the [MockRunner JDBC](#) library can really simplify the mocking of `ResultSet` interaction, which is often enough to get basic unit testing done. Vanilla provides an added layer on top of the MockRunner `MockResultSet` class so that the `ResultSet` may be configured using a simple DSL, such as:

```
ResultSet rs = ResultSetBuilder.resultSet {
    columns 'first_name', 'last_name', 'phone_number', 'age'
    data 'Fred', 'Flintstone', '555-123-9876', 56
    object phoneRecord
    map firstName:'Barney', lastName:'Rubble', age:55, phoneNumber:'555-222-3456'
}
```

The resulting `ResultSet` is implemented by the `com.mockrunner.mock.jdbc.MockResultSet` class from MockRunner and should behave like a real `ResultSet` within the scope of the test mocking.

## ResultSet DSL

The `ResultSet` DSL consists of four statements:

### columns

Accepts a `String...` or `List<String>` argument to provide the names of the columns configured in the `ResultSet`.

### data

Accepts a `String...` or `List<String>` argument to provide the data for a single row, in the same order as the columns.

### object

Accepts an Object which will be used to populate the row. The column names will be converted to camel-case and used to find properties on the object.

### map

Accepts a Map which will be used to populate the row. The column names will be converted to camel-case and used to find properties in the map.