json.mset

- Examples

Set JSON values for multiple keys. The operation is atomic. Either all values are set or none is set.

- If the path calls for an object member:
  - If the parent element does not exist, the command will return NONEXISTENT error.
  - If the parent element exists but is not an object, the command will return ERROR.
  - If the parent element exists and is an object:
    - If the member does not exist, a new member will be appended to the parent object if and only if the parent object is the last child in the path. Otherwise, the command will return NONEXISTENT error.
    - If the member exists, its value will be replaced by the JSON value.
- If the path calls for an array index:
  - If the parent element does not exist, the command will return a NONEXISTENT error.
  - If the parent element exists but is not an array, the command will return ERROR.
  - If the parent element exists but the index is out of bounds, the command will return OUTOFBOUNDARIES error.
  - If the parent element exists and the index is valid, the element will be replaced by the new JSON value.
- If the path calls for an object or array, the value (object or array) will be replaced by the new JSON value.

Examples

Enhanced path syntax:

```
127.0.0.1:6379> JSON.MSET k1 . '[1,2,3,4,5]' k2 . '{"a":{"a":1, "b":2, "c":3}}' k3 . '{"a":
[1,2,3,4,5]}'
OK
127.0.0.1:6379> JSON.GET k1
"[1,2,3,4,5]"
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":1,\"b\":2,\"c\":3}}"
127.0.0.1:6379> JSON.MSET k2 $.a.* '0' k3 $.a[*] '0'
```

```
OK
127.0.0.1:6379> JSON.GET k2
"{\"a\":{\"a\":0,\"b\":0,\"c\":0}}"
127.0.0.1:6379> JSON.GET k3
"{\"a\":[0,0,0,0,0]}"
```

Restricted path syntax:

```
127.0.0.1:6379> JSON.MSET k1 . '{"name": "John","address": {"street": "123 Main St","city":
"Springfield"},"phones": ["555-1234","555-5678"]}'
OK
127.0.0.1:6379> JSON.MSET k1 .address.street '"21 2nd Street"' k1 .address.city '"New York"'
OK
127.0.0.1:6379> JSON.GET k1 .address.street
"\"21 2nd Street\""
127.0.0.1:6379> JSON.GET k1 .address.city
"\"New York\""
```

□□□ #2
□ □□□ □□□ 2026-10-31 15:16:15 UTC
□ □□□ □□□ 2026-10-31 15:16:16 UTC