

# Working with Transactions

Using the Transaction API, client applications may create then commit or abort TM/MP (TMF) transactions. When a transaction is created, an opaque transaction ID is returned. The transaction ID is used in subsequent calls to commit or abort the transaction, and may be supplied as parameters to other API requests in order to enlist that request in the transaction.

The following example shows how to create a transaction, send a serverclass request enlisted in the transaction, and commit the transaction.

## Begin the Transaction

### Request

```
POST http://lightwave.example.com/lightwave/api/v1/transaction HTTP/1.1
Content-Type: application/json
lw-transaction-timeout: 30
```

### Response

```
HTTP/1.1 200 OK
Content-Type: application/json
lw-transaction-id: 2c00031c9bb70000380b806a992f7c6e
lw-transaction-state: suspended
```

## Send the Serverclass Request with the Transaction

### Request

```
POST http://lightwave.example.com/lightwave/api/v1/serverclass/=ne^pathmon/nesvr HTTP/1.1
Content-Type: application/json
lw-transaction-id: 2c00031c9bb70000380b806a992f7c6e

{
  "requestCode" : 1
}
```

### Response

```
HTTP/1.1 200 OK
Content-Type: application/json
lw-transaction-id: 2c00031c9bb70000380b806a992f7c6e
lw-transaction-state: suspended

{
  "resultCode": 0,
  "node": "\\ACT5",
  "systemInfo": {
    "nodeNumber": 43,
    "systemSubvol": "SYS03",
    "rvu": "G06"
  }
}
```

## Commit the Transaction

### Request

```
POST http://lightwave.example.com/lightwave/api/v1/transaction/2c00031c9bb70000380b806a992f7c6e HTTP/1.1
Content-Length: 0
```

**Response**

```
HTTP/1.1 200 OK
Content-Type: application/json
lw-transaction-id: 2c00031c9bb70000380b806a992f7c6e
lw-transaction-state: committed
```