op (/js/op).write

```
op.write(
   [cols as docColsIdentifier?]
) as ModifyPlan
```

Summary

Inserts or overwrites the documents identified by the uri column with the data supplied by the other document descriptor columns.

Parameters	
cols	An object to specify which document columns to use when writing. If this parameter is not passed in op.docCols() is used as default.

See Also

- Optic API for Multi-Model Data Access (/guide/app-dev/OpticAPI#id_93259) in the *Application Developer's Guide*

Example

```
   //write 3 new docs, returns the docs as well
declareUpdate();
const op = require('/MarkLogic/optic');
const docsDescriptor = [
  {uri:'/optic/update/write4.json', doc:{"desc":"write4"}},
  {uri:'/optic/update/write5.json', doc:{"desc":"write5"}, collections:['write']},
  {uri:'/optic/update/write6.json', doc:{"desc":"write6"}, collections:['write', 'write6'], metadata:{
];
op.fromDocDescriptors(docsDescriptor)
  .write(op.docCols())
  .result();
```

Example

```
   //write 3 new docs, returns nothing
declareUpdate();
const op = require('/MarkLogic/optic');
const docsDescriptor = [
  {uri:'/optic/update/write7.json', doc:{"desc":"write7"}, collections:['write', 'write7'], metadata:{
  {uri:'/optic/update/write8.json', doc:{"desc":"write8"}, collections:['write', 'write8'], metadata:[
  {uri:'/optic/update/write9.json', doc:{"desc":"write9"}, collections:['write', 'write9'], metadata:[
];
op.fromDocDescriptors(docsDescriptor)
  .orderBy('uri')
  .write({uri:'uri', doc:'doc', collections:'collections', metadata:'metadata', permissions:'permissio
  .execute();
```

Example

```
declareUpdate();
const op = require('/MarkLogic/optic');
const rows = [
    {id:1, myQua: 4,  o_uri:'/test/fromDocUris/1.json', myUri:'/optic/update/write10.json', myColl:['/
    {id:2, myQua: 5,  o_uri:'/test/fromDocUris/2.json', myUri:'/optic/update/write11.json', myColl:'/o
    {id:3, myQua: 6,  o_uri:'/test/fromDocUris/3.json', myUri:'/optic/update/write12.json', myColl:'/o
    ];
const outputCols = [
    {"column":"id",     "type":"integer", "nullable":false},
    {"column":"myQua",  "type":"integer", "nullable":true},
    {"column":"o_uri",  "type":"string",  "nullable":false},
    {"column":"myUri",  "type":"string",  "nullable":false},
    {"column":"myColl", "type":"none",    "nullable":true},
    {"column":"meta",   "type":"none",    "nullable":true}
    ];
op.fromParam('bindingParam', null, outputCols)
  .joinDoc(op.col('myDoc'), op.col('o_uri'))
  .bind(op.as('myMeta',op.jsonObject([op.prop('meta', op.jsonString(op.col('meta')))])))
  .bind(op.as('myPerm', op.jsonArray([
        op.jsonObject([op.prop('capability', 'read'), op.prop('roleId', xdmp.role('rest-reader'))]),
        op.jsonObject([op.prop('capability', 'update'), op.prop('roleId', xdmp.role('rest-writer'))])]
    )
  .orderBy('id')
  .write({uri:op.col('myUri'), doc:'myDoc', collections: op.col('myColl'), metadata: 'myMeta', quality
  .result('object', {"bindingParam":rows});
```

Example

```
//write a temporal document '/optic/update/temporal/temporal1.json' with temporal collection 'update'
declareUpdate();
const op = require('/MarkLogic/optic');
const docsDescriptor = [
  {uri:'/optic/update/temporal/temporal1.json',
   doc:{"desc":"temporal doc 1.1"},
   collections:['temporal1', 'write1', 'write1.1'],
   metadata:[{'validStart': "2014-04-03T11:00:00", 'validEnd': "9999-12-31T11:59:59Z", 'update': 1}],
   permissions:[xdmp.permission("rest-reader","read"),xdmp.permission("rest-writer","read"),xdmp.permi
   quality:11,
   temporalCollection:'update'
  }
];
op.fromDocDescriptors(docsDescriptor)
  .write(null)
  .result();
```

Example

```
// Overwrite document collections
declareUpdate();
const op = require('/MarkLogic/optic');
const docsDescriptor = [
  {uri:'/optic/update/write1.json', collections:['overwrite', 'overwrite10']},
  {uri:'/optic/update/write2.json'},
  {uri:'/optic/update/write3.json', collections:Sequence.from(['overwrite', 'overwrite12'])}
];
op.fromDocDescriptors(docsDescriptor)
  .orderBy('uri')
  .write(op.docCols(null, ['uri', 'doc', 'collections']))
  .result();
```