# AccessPlan.prototype (/js/AccessPlan).validateDoc

```
AccessPlan.prototype.validateDoc(
   validateDocCol as String,
   schemaDef as Object
) as ModifyPlan
```

## Summary

Validate the document based on a supplied schema. This schema needs to be stored in the schema database. Check appserver error log for validate errors.

| Parameters | |
|---|---|
| validateDocCol | This is the column which contains the document to validate. This can be a string of the column name or an op.col (./op.col). Use op.viewCol (./op.viewCol) or op.schemaCol (./op.schemaCol) if you need to identify columns in the two views that have the same column name. |
| schemaDef | This is an object. The required 'kind' property of the schema object must be 'jsonSchema', 'schematron', or 'xmlSchema'. When 'kind' is 'jsonSchema' or 'schemtron' then a property 'schemaUri' is required. Property 'mode' takes 'strict', 'lax' or 'type' (refer to xdmp.validate). |

## See Also

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

## Example

```
const op = require('/MarkLogic/optic');
                op.fromDocDescriptors([{uri:'/optic/validateDoc/toValidate1.xml'}])
                   .joinDocCols(null, op.col('uri'))
                   .validateDoc('doc', {kind:'xmlSchema', mode:'strict'})
                   .result();
```

# Example

```
const op = require('/MarkLogic/optic');
op.fromDocDescriptors([{uri:'/optic/validateDoc/toValidate1.xml'},
                       {uri:"/optic/validateDoc/toValidate2.xml"},
                       {uri:"/optic/validateDoc/toValidate3.xml"}])
   .joinDocCols(null, op.col('uri'))
   .orderBy('uri')
   .validateDoc('doc', {kind:'xmlSchema', mode:'type', type:fn.QName("http://marklogic.com/xdmp/group"
   .result();
```

# Example

```
  const op = require('/MarkLogic/optic');
  op.fromDocDescriptors([{uri:"/optic/validateDoc/validate4.json"}])
     .joinDocCols(null, op.col('uri'))
     .validateDoc('doc', {kind:'jsonSchema', schemaUri:'/validateDoc-test.json', mode:"full"})
     .result();
```

# Example

```
const op = require('/MarkLogic/optic');
op.fromDocDescriptors([{uri:'/optic/validateDoc/toValidateSchematron1.xml'},
                       {uri:"/optic/validateDoc/toValidateSchematron2.xml"},
                       {uri:"/optic/validateDoc/toValidateSchematron3.xml"}], 'view')
   .joinDocCols(op.docCols('view'), op.viewCol('view','uri'))
   .orderBy(op.viewCol('view','uri'))
   .validateDoc(op.viewCol('view','doc'), {kind:'schematron', schemaUri:"/validateDoc-test.sch"})
   .result();
```