

Strategy: DOCUMENT_FROM_VIEW_BY_KEY

Use this strategy to select document from the view by key. The view will be searched by the first sorted column.

Allowed method

- GET
- POST

Parameters:

- `databaseName(String databaseName)`: name of the database to read data from, use "server!!path/database.nsf" format; when omitted, current database is used
- `viewName(String viewName)`: name of the view; mandatory
- `keyVariableName(String keyVariableName)`: name of the variable name in route to read value from; mandatory
- `formName(String formName)`: name of the Form used when creating new document via `@new`

Let's assume our database has view called (`ByTopic`), that contains first sorted column displaying `Topic` field.

Notice the `{keyValue}` part of the first line below. This is placeholder for where SmartNSF expects to read parameter value for `keyVariableName` from, see line 4.

```
router.GET('topics/key/{keyValue}') {
  strategy(DOCUMENT_FROM_VIEW_BY_KEY) {
    viewName('ByTopic')
    keyVariableName('keyValue')
  }
  mapJson 'id', json:'id', type:'STRING', isformula:true, formula:'@DocumentUniqueID'
  mapJson 'date_created', json:'date_created', type:'DATETIME', isformula:true, formula:'@Created'
  mapJson 'topic', json:'topic', type:'STRING'
  mapJson 'author', json:'author', type:'STRING', isformula:true, formula:'@Name([CN]; @Author)'
}
```

With route defined above, URL `http://server.name/path-to/db.nsf/xsp/.xrest/topics/key/test` will return first document from view (`ByTopic`) with field `Topic` "test".

For example, we'll get this JSON:

```
{
  "date_created": "2017-03-11T13:11+00:00",
  "author": "Martin Jinoch",
  "id": "DCEFCAE41C0C9815C12580E0004BD296",
  "topic": "test"
}
```

POST method

Creating new document

With route defined below, you can create new document by POSTing to URL `http://server.name/path-to/db.nsf/xsp/.xrest/topics/key/@new`

```
router.POST('topics/key/{keyValue}') {
  strategy(DOCUMENT_FROM_VIEW_BY_KEY) {
    viewName('ByTopic')
    keyVariableName('keyValue')
    formName('MainTopic')
  }
  mapJson 'id', json:'id', type:'STRING', isformula:true, formula:'@DocumentUniqueID', readonly:true
  mapJson 'date_created', json:'date_created', type:'DATETIME', isformula:true, formula:'@Created', readonly:
true
  mapJson 'topic', json:'topic', type:'STRING'
  mapJson "body", json:'content', type:'MIME'
  mapJson 'categories', json:'categories', type:'ARRAY_OF_STRING', writeonly:true
  mapJson "author", json:'author', type:'STRING', isformula:true, formula:'@Name([CN]; From)', readonly:true
}
```

Related articles

- [Strategy: CUSTOM](#)
- [Strategy: DOCUMENT_BY_UNID](#)
- [Strategy: DOCUMENT_FROM_VIEW_BY_KEY](#)
- [Strategy: DOCUMENTS_BY_FORMULA](#)
- [Strategy: DOCUMENTS_BY_FORMULA_PAGED](#)
- [Strategy: DOCUMENTS_BY_SEARCH_FT](#)
- [Strategy: DOCUMENTS_BY_SEARCH_FT_PAGED](#)
- [Strategy: DOCUMENTS_BY_VIEW](#)
- [Strategy: DOCUMENTS_BY_VIEW_PAGED](#)
- [Strategy: DOCUMENTS_FROM_VIEW_BY_KEY](#)
- [Strategy: DOCUMENTS_FROM_VIEW_BY_KEY_PAGED](#)
- [Strategy: VIEWENTRIES](#)
- [Strategy: VIEWENTRIES_BY_CATEGORY](#)
- [Strategy: VIEWENTRIES_BY_CATEGORY_PAGED](#)
- [Strategy: VIEWENTRIES_PAGED](#)