

Strategy: VIEWENTRIES_BY_CATEGORY

Available since Beta 4

Use this strategy to get viewentries (columns) from the view. This is the fastest way to get data from the view.

Allowed method

- GET

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

Let's assume our database has view called `By Category`.

```
router.GET('topics/category/{key}') {
  strategy(VIEWENTRIES_BY_CATEGORY) {
    viewName('By Category')
    keyVariableName('key')
  }
  mapJson '@unid', json:'id', type:'STRING'
  mapJson '\$115', json:'topic', type:'STRING'
}
```

`mapJson` expects to get programmatic name of the column, `json` name and `type` (allowed types are `STRING`, `DATETIME` and `DOUBLE`, for multi-value columns use `ARRAY_OF_STRING`, `ARRAY_OF_DATETIME` and `ARRAY_OF_DOUBLE`).

If programmatic name contains `$`, you must escape it: `\$`

You can also use `@unid` to get universal id of the documents corresponding with the viewentries.

With route defined above, URL `http://server.name/path-to/db.nsf/xsp/.xrest/topics/category/test` will return all viewentries (columns) under category "test" from view `By Category`.

Returned JSON will contain an array of columns from the view, such as:

```
[
  {
    "id": "54502859C07299C7C12580D8006404F4",
    "topic": "updated from rest (Martin Jinoch)"
  },
  {
    "id": "FB0ABE3F113B5EE3C12580E100361BD1",
    "topic": "created from rest (Martin Jinoch [Anonymous])"
  },
  {
    "id": "CD4C360B5B275798C12580E100364A4B",
    "topic": "created from rest (Martin Jinoch [Anonymous])"
  },
  {
    "id": "EDA5D746AC3B64CEC12580E1003680AA",
    "topic": "created from rest (Martin Jinoch [Anonymous])"
  },
  {
    "id": "4B7878B42A1F7BD8C12580E10036C46E",
    "topic": "created from rest (Martin Jinoch [Anonymous])"
  },
  {
    "id": "52CED96E3A9556C3C12580E40062E73B",
    "topic": "created from rest (Anonymous)"
  },
  {
    "id": "09C68CACF670FE9FC12580E400645002",
    "topic": "created from rest ()"
  },
  {
    "id": "72E9DCB48D258651C12580E4006456CE",
    "topic": "created from rest ()"
  },
  {
    "id": "E844BB2879E055F0C12580F3004FDBF6",
    "topic": "updated from rest (Martin Jinoch)"
  },
  {
    "id": "41B590BBCFDE224EC12580F3004FDBF7",
    "topic": "created from rest (Martin Jinoch [Anonymous])"
  }
]
```

Related articles

Page:Strategy: CUSTOM

Page:Strategy: DOCUMENT_BY_UNID

Page:Strategy: DOCUMENT_FROM_VIEW_BY_KEY

Page:Strategy: DOCUMENTS_BY_FORMULA

Page:Strategy: DOCUMENTS_BY_FORMULA_PAGED
Page:Strategy: DOCUMENTS_BY_SEARCH_FT
Page:Strategy: DOCUMENTS_BY_SEARCH_FT_PAGED
Page:Strategy: DOCUMENTS_BY_VIEW
Page:Strategy: DOCUMENTS_BY_VIEW_PAGED
Page:Strategy: DOCUMENTS_FROM_VIEW_BY_KEY
Page:Strategy: DOCUMENTS_FROM_VIEW_BY_KEY_PAGED
Page:Strategy: VIEWENTRIES
Page:Strategy: VIEWENTRIES_BY_CATEGORY
Page:Strategy: VIEWENTRIES_BY_CATEGORY_PAGED
Page:Strategy: VIEWENTRIES_PAGED