NetApp:
Swagger UI Alternatives

Group Members: Sophie Sfeir, Anthony McGaw, Jarod Miller, Madison Haugh

Sophie Sfeir

About me:

❑ University of Pittsburgh Student, BS in Computer Science and Spanish, Graduating in Fall 2021

Why I chose this project:

❑ I am interested in learning new ideas/concepts and wanted to learn more about APIs and how they work. I also wanted to gain more professional experience in software development.

Favorite part:

❑ Learning how to build off a pre-existing codebase with new coding languages, and having the experience to work collaboratively in a professional setting.

Anthony McGaw

About me:

❏ University of Pittsburgh Student, BS in Computer Science, Graduating in May 2021

Why I chose this project:

❏ I have recently been interested in UX and this felt like a interesting project to get experience working on a project which is important to end-users. Also valuable experience with Agile, React, and API’s.

Favorite Part:

❏ Working through the Agile process with a reliable team, and understanding the lifecycle of a large project

Jarod Miller

About me:

- ❏ University of Pittsburgh Student, BS in Computer Science, Graduating in May 2021

Why I chose this project:

- ❏ I'm interested in gaining experience in full-stack website development and learning new technologies in the tech industry.

Favorite Part:

- ❏ Learning how to code in Typescript with React and managing a large piece of software with a team.

Madison Haugh

About me:

- University of Pittsburgh Student, BS in Computer Science, Minor in Applied Statistics, Graduating in May 2021

Why I chose this project:

- I’ve had previous experience with full-stack web development and was interested in developing and expanding my skill set

Favorite Part:

- Getting to work on a real-life problem in an industry setting

Outline

❑ Midterm Review
❑ Project Development Process
❑ Project Experience
❑ Second Half Goals
❑ Challenges
❑ Features/Functionalities
❑ Performance Enhancements
❑ Demo
❑ Future Suggestions

Midterm Review

❑ Research API Visualizers

❑ Decide on API Visualizer

❑ Learn Redoc Codebase

❑ Implement deep search functionality

❑ Import models

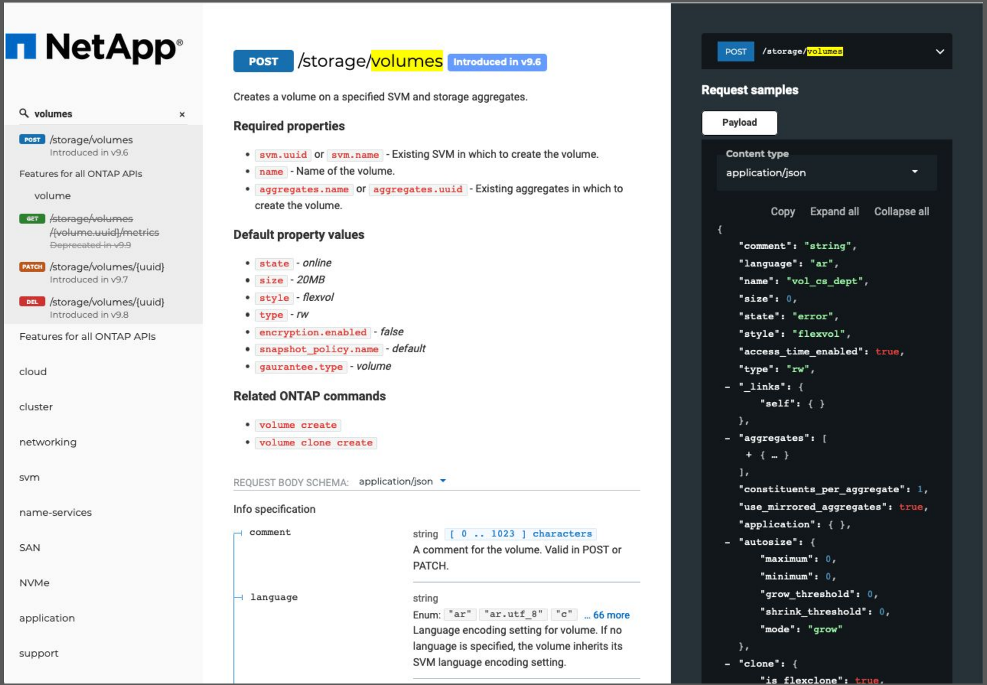
Decision Review: Redoc

Reasoning

- ❑ Built-in search
  - Main feature
- ❑ Three-panel display
  - Much more user friendly
- ❑ Language
  - React, Typescript
  - Best suited for our skillset

Shortcomings

- ❑ Extra work converting from Swagger UI



POST /storage/volumes Introduced in v9.6

Creates a volume on a specified SVM and storage aggregates.

Required properties

- `svm.uuid` or `svm.name` - Existing SVM in which to create the volume.
- `name` - Name of the volume.
- `aggregates.name` or `aggregates.uuid` - Existing aggregates in which to create the volume.

Default property values

- `state` - *online*
- `size` - *20MB*
- `style` - *flexvol*
- `type` - *rw*
- `encryption.enabled` - *false*
- `snapshot_policy.name` - *default*
- `gaurantee.type` - *volume*

Related ONTAP commands

- `volume create`
- `volume clone create`

REQUEST BODY SCHEMA: application/json

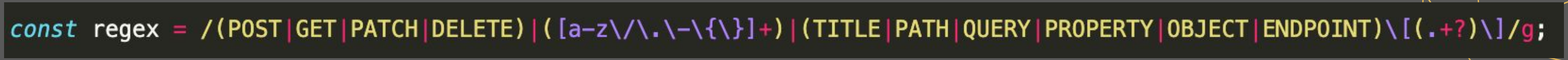
Info specification

comment — string `[ 0 .. 1023 ] characters`
A comment for the volume. Valid in POST or PATCH.

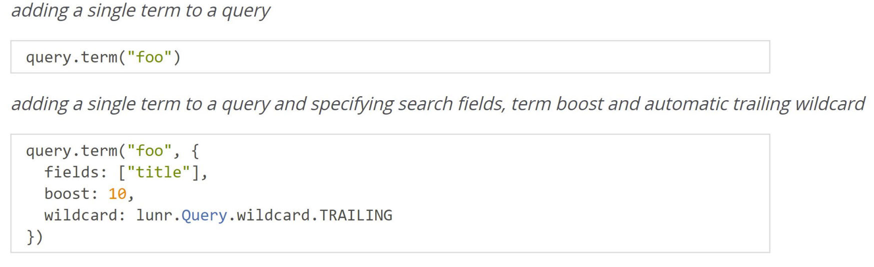
language — string
Enum: `"ar"` `"ar.utf_8"` `"c"` ... 66 more
Language encoding setting for volume. If no language is specified, the volume inherits its SVM language encoding setting.

```
{
    "comment": "string",
    "language": "ar",
    "name": "vol_cs_dept",
    "size": 0,
    "state": "error",
    "style": "flexvol",
    "access_time_enabled": true,
    "type": "rw",
  - "_links": {
        "self": { }
    },
  - "aggregates": [
     + { … }
    ],
    "constituents_per_aggregate": 1,
    "use_mirrored_aggregates": true,
    "application": { },
  - "autosize": {
        "maximum": 0,
        "minimum": 0,
        "grow_threshold": 0,
        "shrink_threshold": 0,
        "mode": "grow"
    },
  - "clone": {
        "is_flexclone": true,
```

Deep Search



```
const regex = /(POST|GET|PATCH|DELETE)|([a-z\/\.\-\{\}]+)|(TITLE|PATH|QUERY|PROPERTY|OBJECT|ENDPOINT)\[(.+?)\]/g;
```

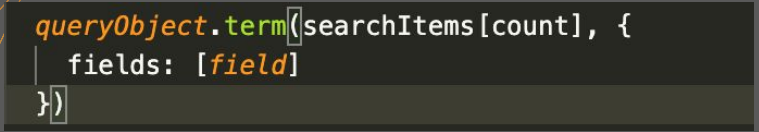


adding a single term to a query

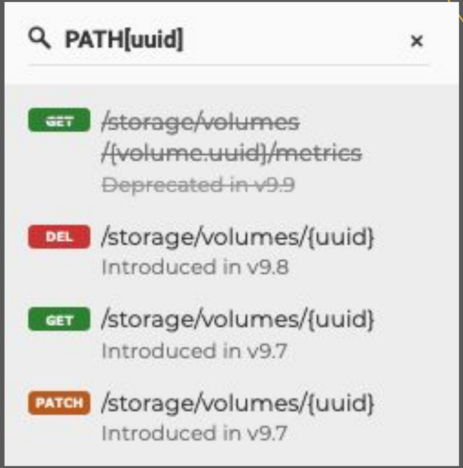
```
query.term("foo")
```

adding a single term to a query and specifying search fields, term boost and automatic trailing wildcard

```
query.term("foo", {
  fields: ["title"],
  boost: 10,
  wildcard: lunr.Query.wildcard.TRAILING
})
```



```
queryObject.term(searchItems[count], {
  fields: [field]
})
```



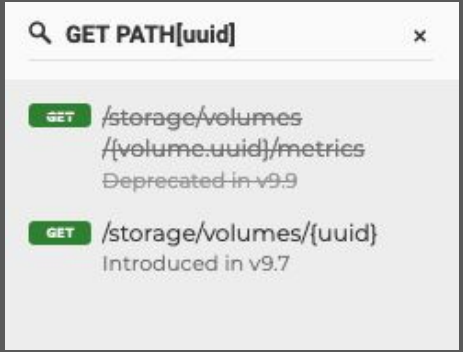
PATH[uuid] ×

GET ~~/storage/volumes/{volume.uuid}/metrics~~
~~Deprecated in v9.9~~

DEL /storage/volumes/{uuid}
Introduced in v9.8

GET /storage/volumes/{uuid}
Introduced in v9.7

PATCH /storage/volumes/{uuid}
Introduced in v9.7



GET PATH[uuid] ×

GET ~~/storage/volumes/{volume.uuid}/metrics~~
~~Deprecated in v9.9~~

GET /storage/volumes/{uuid}
Introduced in v9.7

Development Process

Student Team:

- ❏ Daily collaboration through our team Discord server
- ❏ Weekly meetings on Tuesday nights and Friday afternoons via Zoom
  - ❏ (which evolved into us adding Thursday nights as well)
- ❏ Split up certain feature development into sub-teams within our group

+ Netapp:

- ❏ Second collaborative Discord including our NetApp sponsors to ask questions and discuss challenges
- ❏ Weekly meetings Friday morning at 9am ☀️

Project Experience: Second Half

- In continuation with our midterm performance, we maintained a consistent meeting schedule
- Busy Schedules
  - Required more focus and better time management
- Continued support from our NetApp sponsors
- Assessing our goals against the time remaining

Second Half Goals

❏ Debug and improve performance of the deep search functionality

❏ Implement automated version control

❏ Rework the models section

❏ Design frontend of Try-It-Out feature

❏ Performance Testing

❏ Determine future recommendations

Challenges

1 Testing our implementation thoroughly with the full specification

2 Assessing performance issues and potential solutions

3 Maintaining a proper Agile development process alongside our individual class/finals schedules

Challenges cont…

4 Understanding and analyzing to the large codebase from which we would be developing

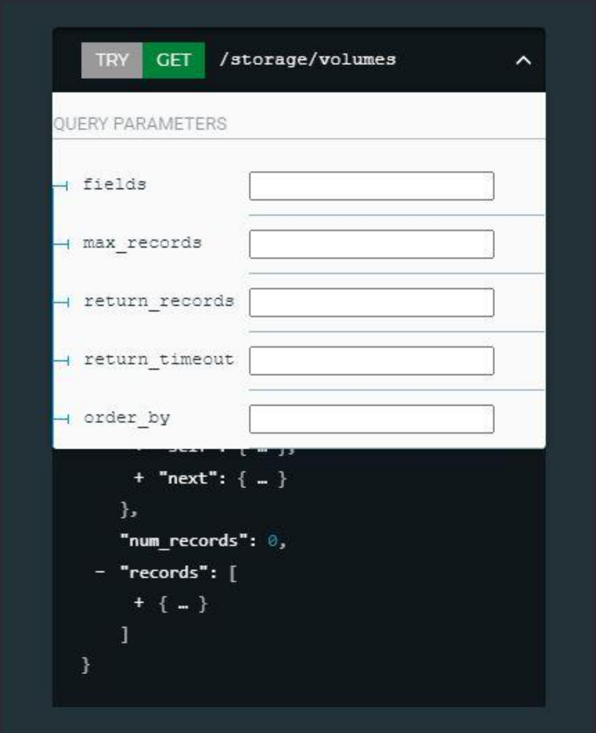
5 Assessing the state of our project and determining the best course for the future

6 Preserving features already implemented into SwaggerUI as we transition to Redoc

Features and Functionalities

Try-It-Out Feature: Front-End

Try-It-Out



TRY GET /storage/volumes

QUERY PARAMETERS

fields	
max_records	
return_records	
return_timeout	
order_by	

```
      + "next": { … }
    },
    "num_records": 0,
  - "records": [
      + { … }
    ]
}
```

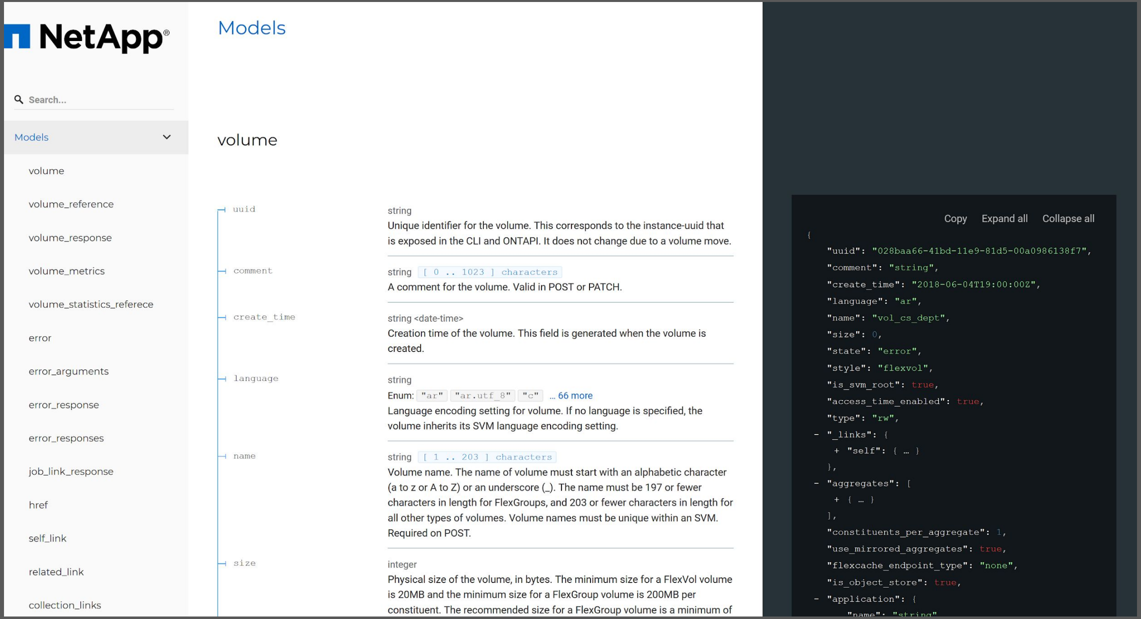
- ❑ Parameter type listed
- ❑ Required parameters explicitly stated
- ❑ Deprecated parameters explicitly stated

- ❑ Open-ended value entry
  - ❑ Future expansion: input validation

- ❑ Removed parameter description
  - ❑ Shown in middle panel

Model Rendering - Rework

Importing the models



Models

volume

uuid

string
Unique identifier for the volume. This corresponds to the instance-uuid that is exposed in the CLI and ONTAPI. It does not change due to a volume move.

comment

string [0 .. 1023] characters
A comment for the volume. Valid in POST or PATCH.

create_time

string <date-time>
Creation time of the volume. This field is generated when the volume is created.

language

string
Enum: "ar" "ar.utf_8" "c" ... 66 more
Language encoding setting for volume. If no language is specified, the volume inherits its SVM language encoding setting.

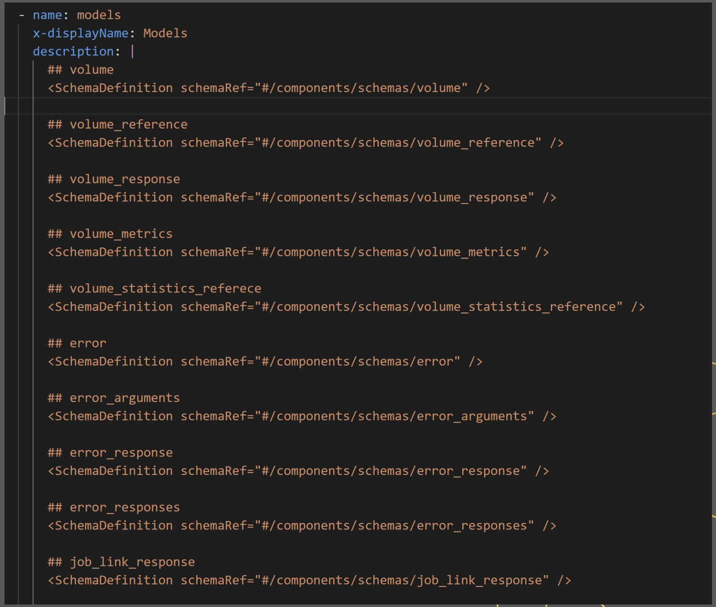
name

string [1 .. 203] characters
Volume name. The name of volume must start with an alphabetic character (a to z or A to Z) or an underscore (_). The name must be 197 or fewer characters in length for FlexGroups, and 203 or fewer characters in length for all other types of volumes. Volume names must be unique within an SVM. Required on POST.

size

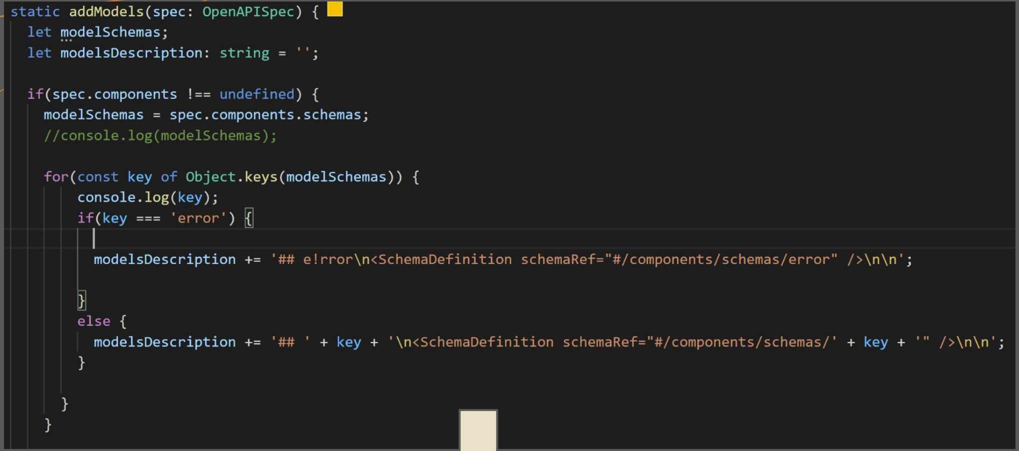
integer
Physical size of the volume, in bytes. The minimum size for a FlexVol volume is 20MB and the minimum size for a FlexGroup volume is 200MB per constituent. The recommended size for a FlexGroup volume is a minimum of

```
{
    "uuid": "028baa66-41bd-11e9-81d5-00a0986138f7",
    "comment": "string",
    "create_time": "2018-06-04T19:00:00Z",
    "language": "ar",
    "name": "vol_cs_dept",
    "size": 0,
    "state": "error",
    "style": "flexvol",
    "is_svm_root": true,
    "access_time_enabled": true,
    "type": "rw",
  - "_links": {
     + "self": { … }
    },
  - "aggregates": [
     + { … }
    ],
    "constituents_per_aggregate": 1,
    "use_mirrored_aggregates": true,
    "flexcache_endpoint_type": "none",
    "is_object_store": true,
  - "application": {
        "name": "string"
```

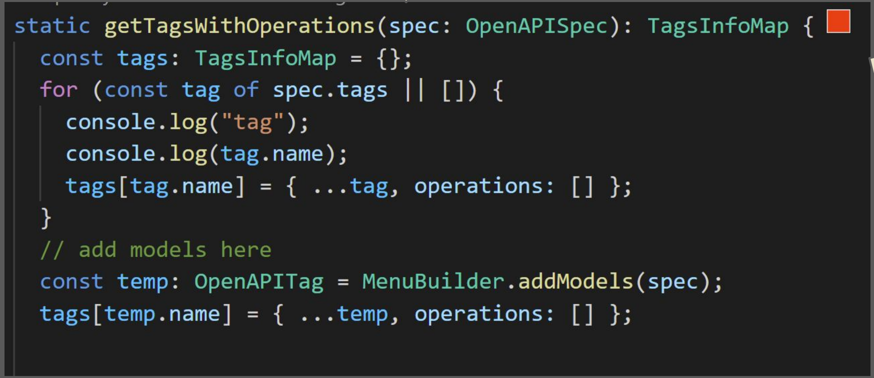


```
- name: models
  x-displayName: Models
  description: |
    ## volume
    <SchemaDefinition schemaRef="#/components/schemas/volume" />

    ## volume_reference
    <SchemaDefinition schemaRef="#/components/schemas/volume_reference" />

    ## volume_response
    <SchemaDefinition schemaRef="#/components/schemas/volume_response" />

    ## volume_metrics
    <SchemaDefinition schemaRef="#/components/schemas/volume_metrics" />

    ## volume_statistics_referece
    <SchemaDefinition schemaRef="#/components/schemas/volume_statistics_reference" />

    ## error
    <SchemaDefinition schemaRef="#/components/schemas/error" />

    ## error_arguments
    <SchemaDefinition schemaRef="#/components/schemas/error_arguments" />

    ## error_response
    <SchemaDefinition schemaRef="#/components/schemas/error_response" />

    ## error_responses
    <SchemaDefinition schemaRef="#/components/schemas/error_responses" />

    ## job_link_response
    <SchemaDefinition schemaRef="#/components/schemas/job_link_response" />
```

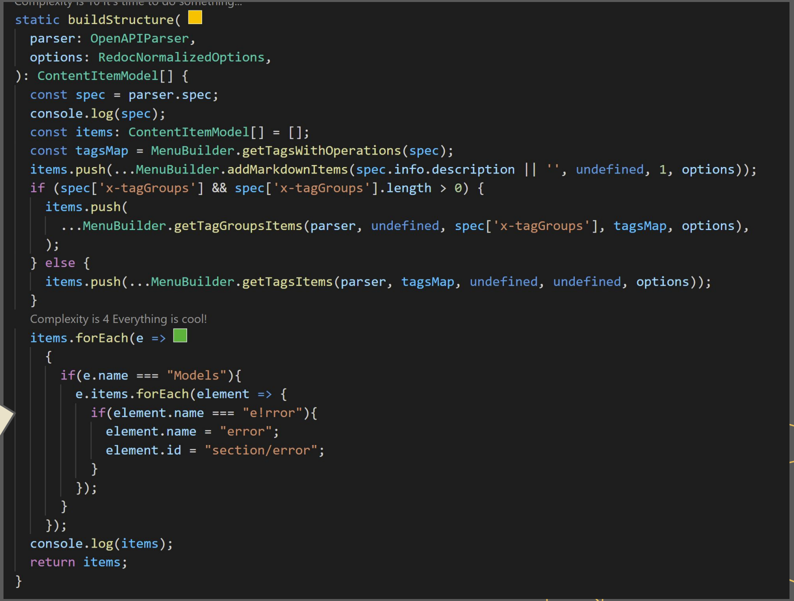
Model Rendering - Rework



```
static addModels(spec: OpenAPISpec) { 🟨
  let modelSchemas;
  let modelsDescription: string = '';

  if(spec.components !== undefined) {
    modelSchemas = spec.components.schemas;
    //console.log(modelSchemas);

    for(const key of Object.keys(modelSchemas)) {
        console.log(key);
        if(key === 'error') {

          modelsDescription += '## e!rror\n<SchemaDefinition schemaRef="#/components/schemas/error" />\n\n';

        }
        else {
          modelsDescription += '## ' + key + '\n<SchemaDefinition schemaRef="#/components/schemas/' + key + '" />\n\n';
        }

      }
    }
```

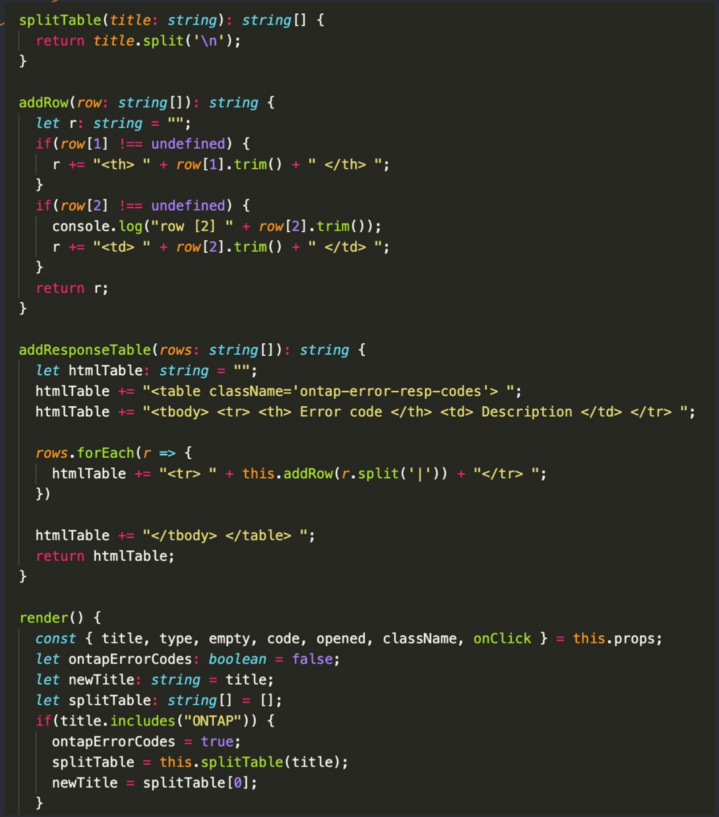


```
static getTagsWithOperations(spec: OpenAPISpec): TagsInfoMap { 🟥
  const tags: TagsInfoMap = {};
  for (const tag of spec.tags || []) {
    console.log("tag");
    console.log(tag.name);
    tags[tag.name] = { ...tag, operations: [] };
  }
  // add models here
  const temp: OpenAPITag = MenuBuilder.addModels(spec);
  tags[temp.name] = { ...temp, operations: [] };
```

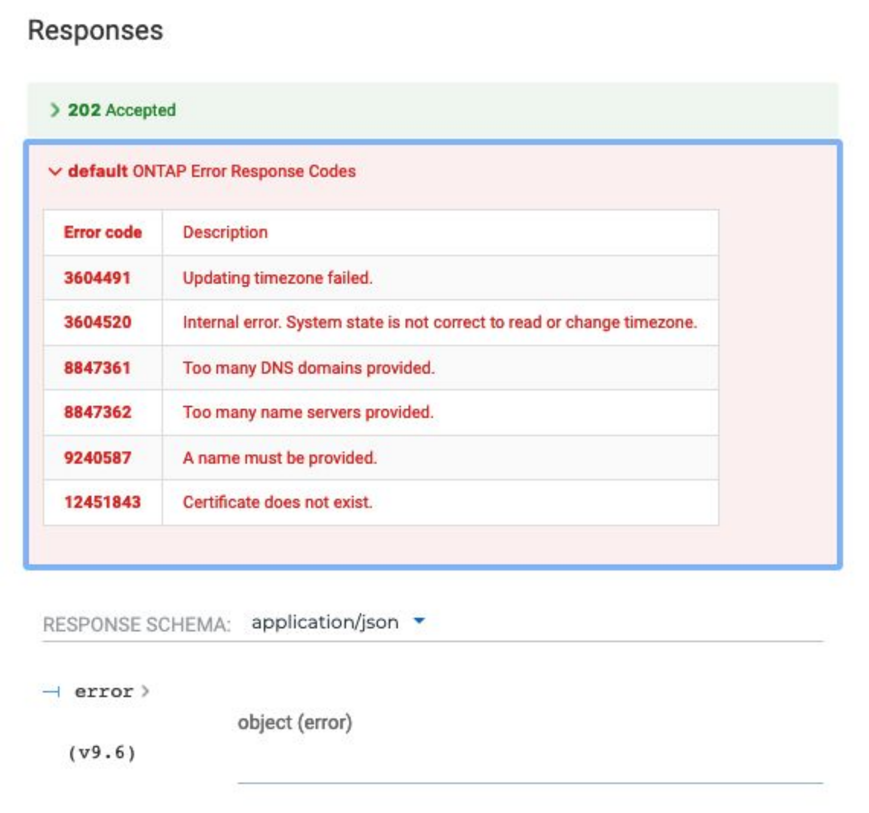


```
Complexity is 10 it's time to do something...
static buildStructure( 🟨
  parser: OpenAPIParser,
  options: RedocNormalizedOptions,
): ContentItemModel[] {
  const spec = parser.spec;
  console.log(spec);
  const items: ContentItemModel[] = [];
  const tagsMap = MenuBuilder.getTagsWithOperations(spec);
  items.push(...MenuBuilder.addMarkdownItems(spec.info.description || '', undefined, 1, options));
  if (spec['x-tagGroups'] && spec['x-tagGroups'].length > 0) {
    items.push(
      ...MenuBuilder.getTagGroupsItems(parser, undefined, spec['x-tagGroups'], tagsMap, options),
    );
  } else {
    items.push(...MenuBuilder.getTagsItems(parser, tagsMap, undefined, undefined, options));
  }
  Complexity is 4 Everything is cool!
  items.forEach(e => 🟩
    {
      if(e.name === "Models"){
        e.items.forEach(element => {
          if(element.name === "e!rror"){
            element.name = "error";
            element.id = "section/error";
          }
        });
      }
    });
  console.log(items);
  return items;
}
```

ONTAP Error Response codes



```
splitTable(title: string): string[] {
  return title.split('\n');
}

addRow(row: string[]): string {
  let r: string = "";
  if(row[1] !== undefined) {
    r += "<th> " + row[1].trim() + " </th> ";
  }
  if(row[2] !== undefined) {
    console.log("row [2] " + row[2].trim());
    r += "<td> " + row[2].trim() + " </td> ";
  }
  return r;
}

addResponseTable(rows: string[]): string {
  let htmlTable: string = "";
  htmlTable += "<table className='ontap-error-resp-codes'> ";
  htmlTable += "<tbody> <tr> <th> Error code </th> <td> Description </td> </tr> ";

  rows.forEach(r => {
    htmlTable += "<tr> " + this.addRow(r.split('|')) + "</tr> ";
  })

  htmlTable += "</tbody> </table> ";
  return htmlTable;
}

render() {
  const { title, type, empty, code, opened, className, onClick } = this.props;
  let ontapErrorCodes: boolean = false;
  let newTitle: string = title;
  let splitTable: string[] = [];
  if(title.includes("ONTAP")) {
    ontapErrorCodes = true;
    splitTable = this.splitTable(title);
    newTitle = splitTable[0];
  }
```



Responses

› **202** Accepted

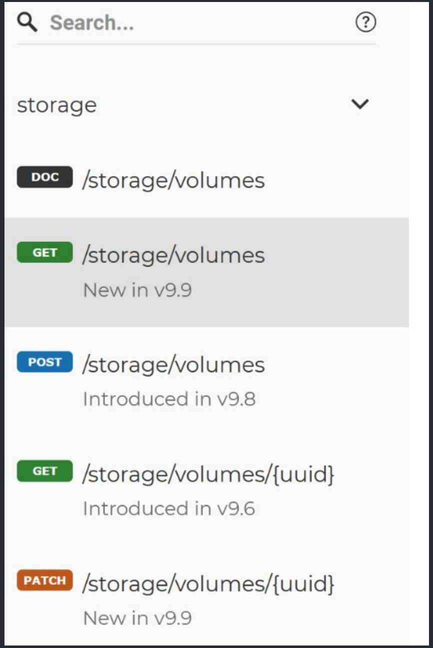
⌄ **default** ONTAP Error Response Codes

Error code	Description
3604491	Updating timezone failed.
3604520	Internal error. System state is not correct to read or change timezone.
8847361	Too many DNS domains provided.
8847362	Too many name servers provided.
9240587	A name must be provided.
12451843	Certificate does not exist.

RESPONSE SCHEMA: application/json ▾

object (error)

Auto Version Control - endpoints



Search...

storage

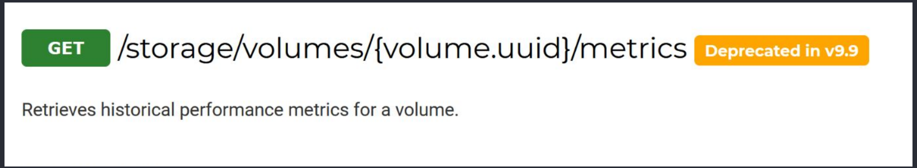
DOC /storage/volumes

GET /storage/volumes
New in v9.9

POST /storage/volumes
Introduced in v9.8

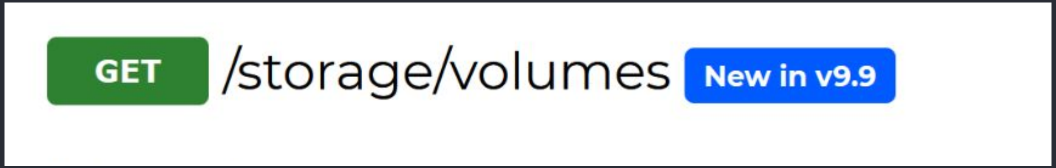
GET /storage/volumes/{uuid}
Introduced in v9.6

PATCH /storage/volumes/{uuid}
New in v9.9



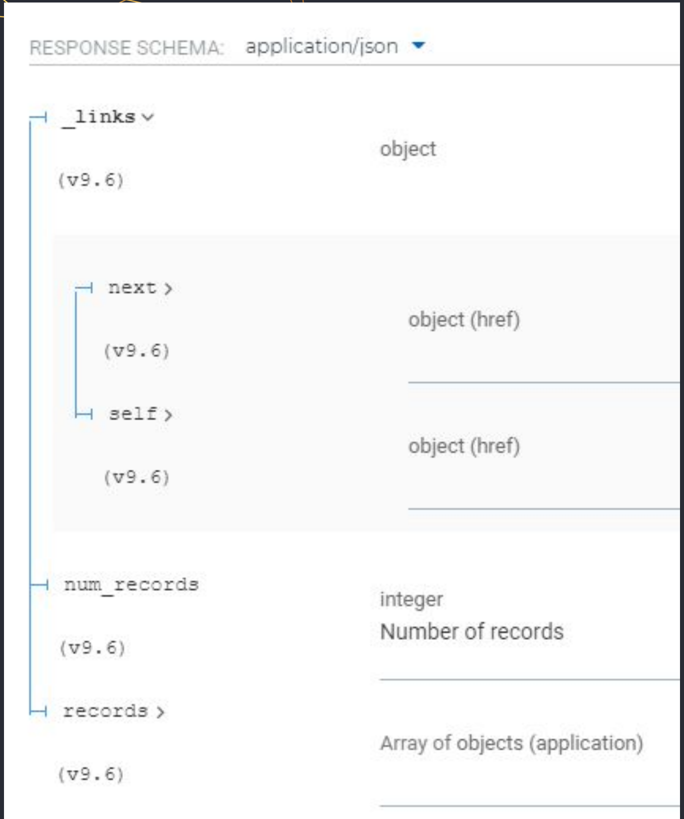
GET /storage/volumes/{volume.uuid}/metrics Deprecated in v9.9

Retrieves historical performance metrics for a volume.



GET /storage/volumes New in v9.9

Auto Version Control - objects, fields and parameters



RESPONSE SCHEMA: application/json

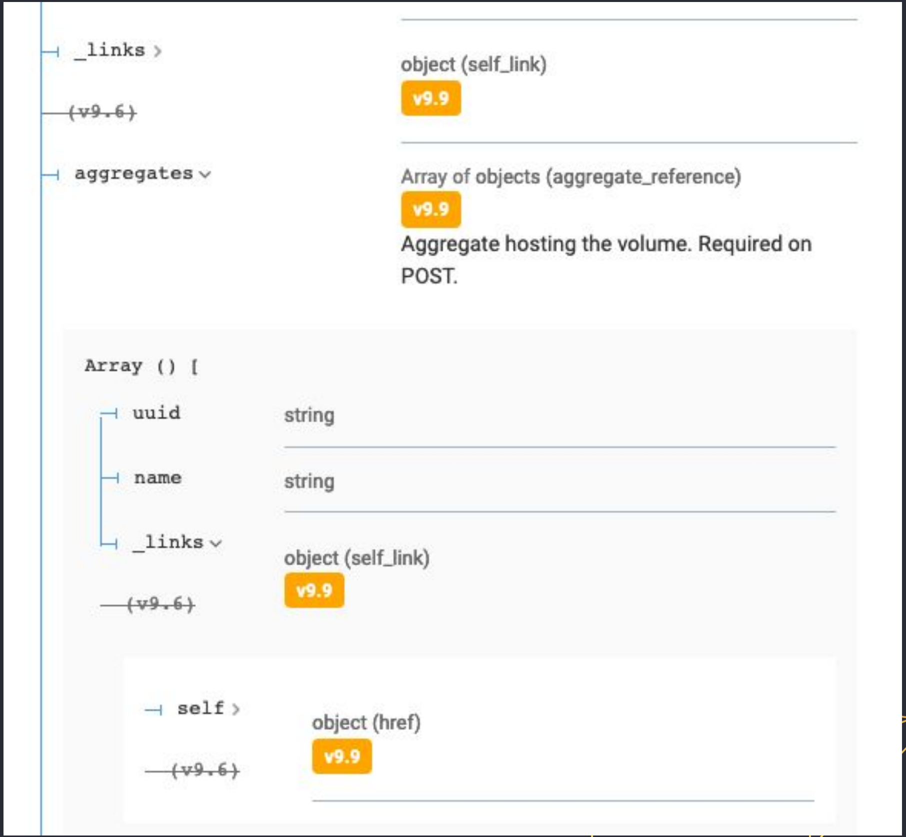
_links
(v9.6)
object

next
(v9.6)
object (href)

self
(v9.6)
object (href)

num_records
(v9.6)
integer
Number of records

records
(v9.6)
Array of objects (application)



_links
~~(v9.6)~~
object (self_link)
v9.9

aggregates
Array of objects (aggregate_reference)
v9.9
Aggregate hosting the volume. Required on POST.

Array () [

uuid
string

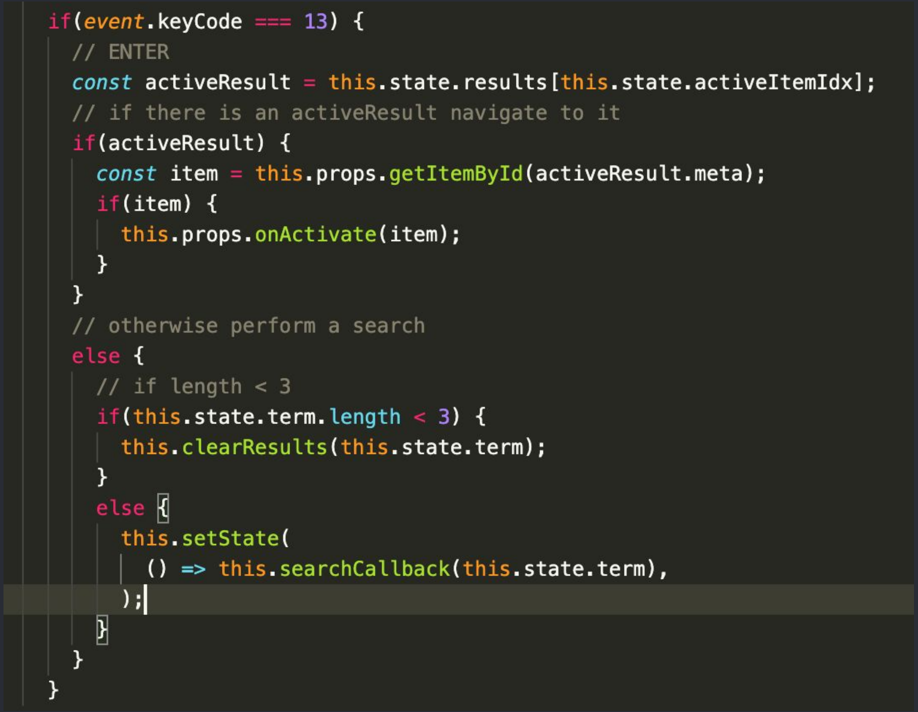
name
string

_links
~~(v9.6)~~
object (self_link)
v9.9

self
~~(v9.6)~~
object (href)
v9.9

Performance Enhancements

Search Rework - Enter



```
if(event.keyCode === 13) {
  // ENTER
  const activeResult = this.state.results[this.state.activeItemIdx];
  // if there is an activeResult navigate to it
  if(activeResult) {
    const item = this.props.getItemById(activeResult.meta);
    if(item) {
      this.props.onActivate(item);
    }
  }
  // otherwise perform a search
  else {
    // if length < 3
    if(this.state.term.length < 3) {
      this.clearResults(this.state.term);
    }
    else {
      this.setState(
        () => this.searchCallback(this.state.term),
      );
    }
  }
}
```

SearchBox.tsx

Performing a live search every time the user presses/deletes a key is impractical for the size of ONTAP’s API.

Instead, we made it so if the user presses enter in the search box, a search is performed.

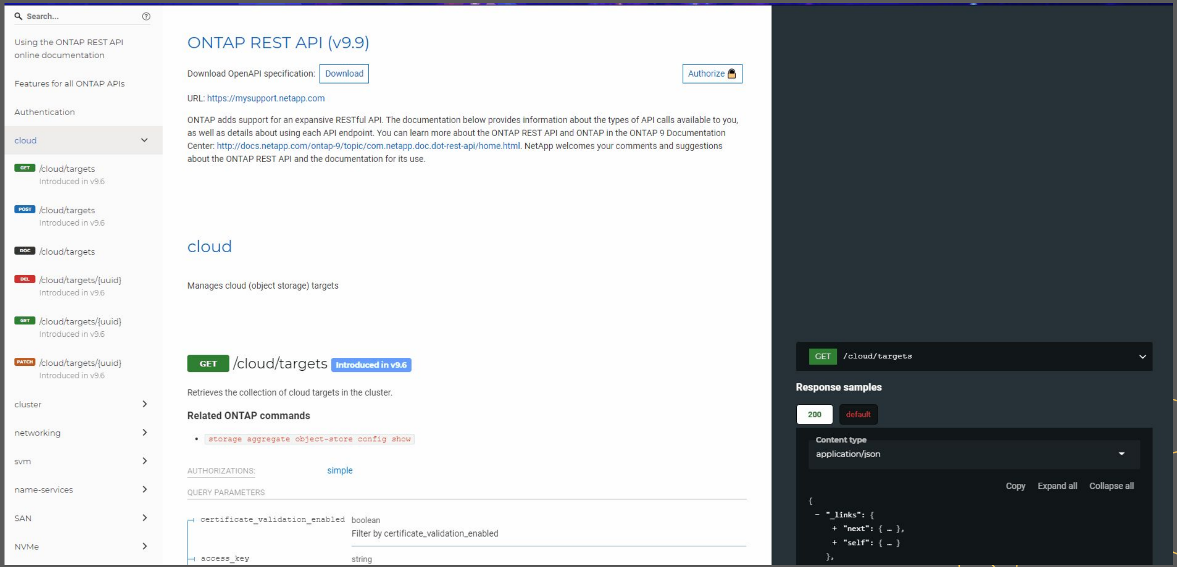
Pagination

Performance Boost:

- Data available on the next slide

No longer loads entire specification

Each Endpoint has its own "page"



Search...

Using the ONTAP REST API online documentation

Features for all ONTAP APIs

Authentication

cloud

- GET /cloud/targets Introduced in v9.6
- POST /cloud/targets Introduced in v9.6
- DOC /cloud/targets
- DEL /cloud/targets/{uuid} Introduced in v9.6
- GET /cloud/targets/{uuid} Introduced in v9.6
- PATCH /cloud/targets/{uuid} Introduced in v9.6

cluster

networking

svm

name-services

SAN

NVMe

ONTAP REST API (v9.9)

Download OpenAPI specification: Download

Authorize

URL: https://mysupport.netapp.com

ONTAP adds support for an expansive RESTful API. The documentation below provides information about the types of API calls available to you, as well as details about using each API endpoint. You can learn more about the ONTAP REST API and ONTAP in the ONTAP 9 Documentation Center: http://docs.netapp.com/ontap-9/topic/com.netapp.doc.dot-rest-api/home.html. NetApp welcomes your comments and suggestions about the ONTAP REST API and the documentation for its use.

cloud

Manages cloud (object storage) targets

GET /cloud/targets Introduced in v9.6

Retrieves the collection of cloud targets in the cluster.

Related ONTAP commands

- `storage aggregate object-store config show`

AUTHORIZATIONS: simple

QUERY PARAMETERS

certificate_validation_enabled	boolean Filter by certificate_validation_enabled
access_key	string

GET /cloud/targets

Response samples

200 default

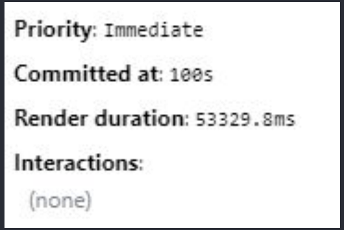
Content type
application/json

Copy Expand all Collapse all

```
{
 - "_links": {
    + "next": { … },
    + "self": { … }
   },
```

Performance Testing

Load/render times before pagination



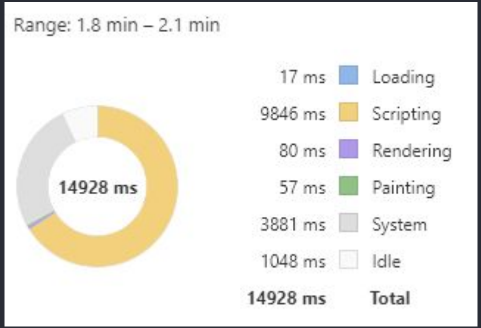
Priority: `Immediate`

Committed at: `100s`

Render duration: `53329.8ms`

Interactions:

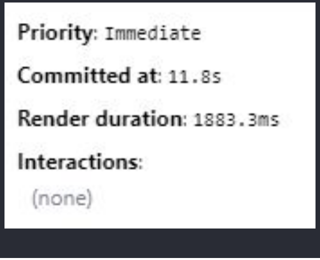
(none)



Range: 1.8 min – 2.1 min

```
   17 ms  🟦 Loading
 9846 ms  🟨 Scripting
   80 ms  🟪 Rendering
   57 ms  🟩 Painting
 3881 ms  ⬜ System
 1048 ms  □ Idle
14928 ms     Total
```

Load/render times after pagination



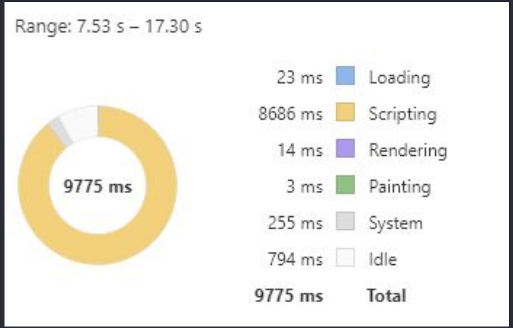
Priority: `Immediate`

Committed at: `11.8s`

Render duration: `1883.3ms`

Interactions:

(none)



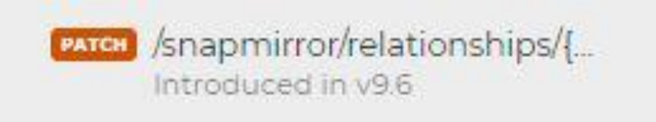
Range: 7.53 s – 17.30 s

```
  23 ms  🟦 Loading
8686 ms  🟨 Scripting
  14 ms  🟪 Rendering
   3 ms  🟩 Painting
 255 ms  ⬜ System
 794 ms  □ Idle
9775 ms     Total
```

Browser Rendering

Chrome & Microsoft Edge:

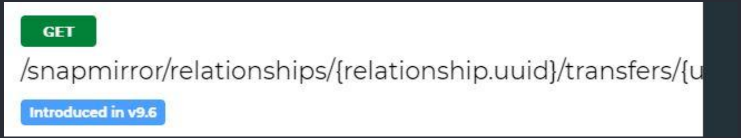
❑ Abbreviates search results



PATCH /snapmirror/relationships/{...

Introduced in v9.6

❑ Cuts off heading

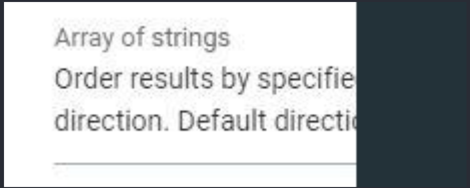


GET

/snapmirror/relationships/{relationship.uuid}/transfers/{u

Introduced in v9.6

❑ Cuts off query parameters



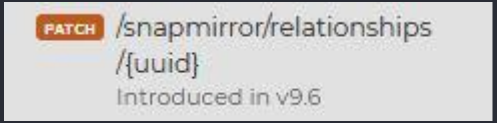
Array of strings

Order results by specifie

direction. Default directi

FireFox:

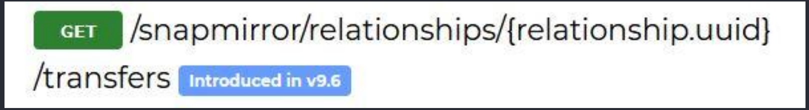
❑ Wraps around search results



PATCH /snapmirror/relationships
/{uuid}

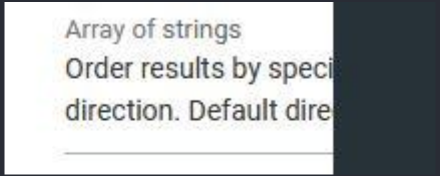
Introduced in v9.6

❑ Wraps around heading



GET /snapmirror/relationships/{relationship.uuid}
/transfers Introduced in v9.6

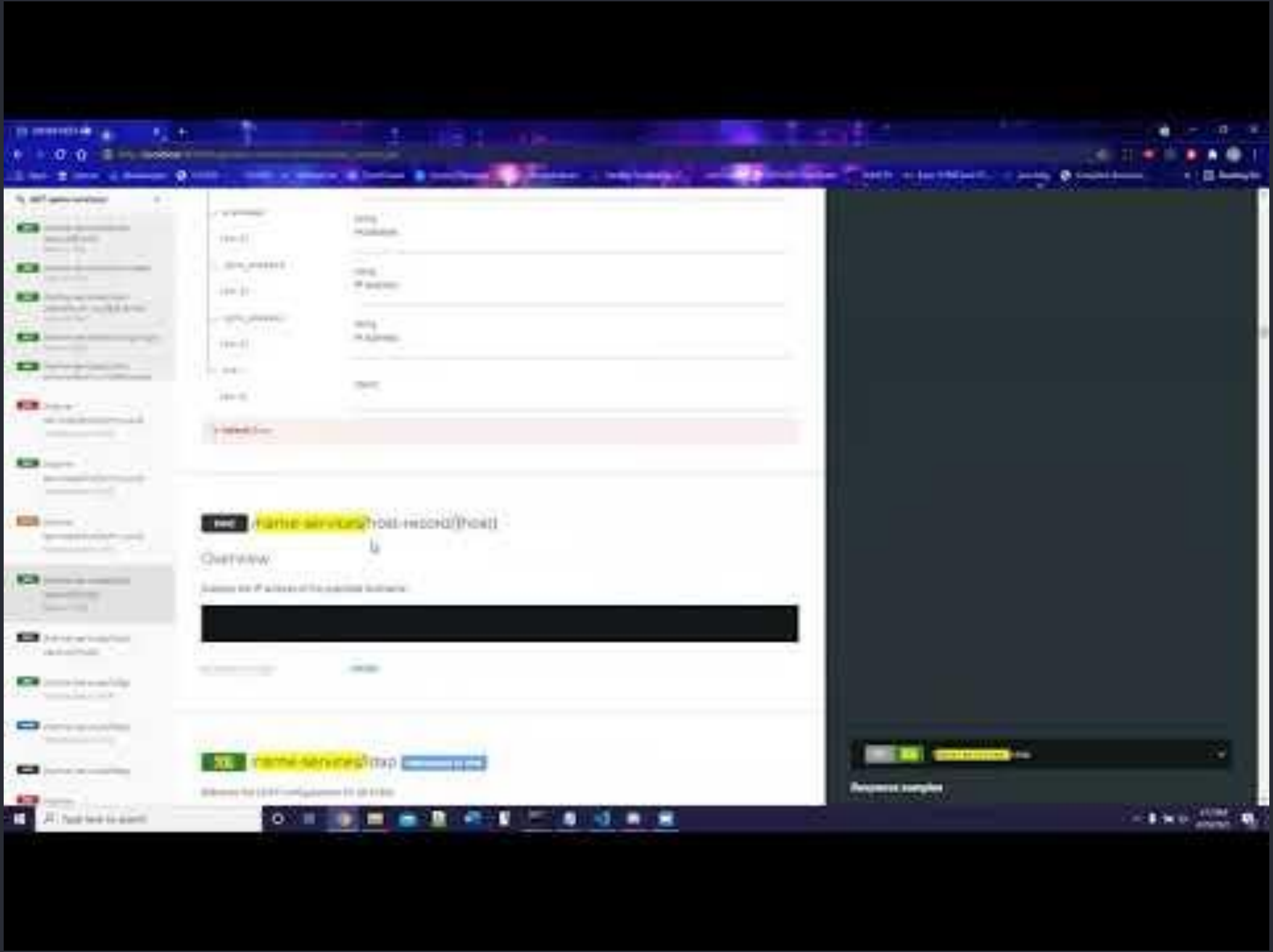
❑ Cuts off query parameters



Array of strings

Order results by speci

direction. Default dire



/name-service/host-record/{host}

Overview

/name-service/ldap

Suggestions for the Future

❑ Consider our implementation as a starting point or an example of how a three-panel design could benefit usability and user experience

TO DO:

- ❑ Improve overall performance
- ❑ Implement Try-It-Out Console and Authentication
- ❑ Add additional unit tests to cover our implemented code
- ❑ Automated testing

❑ Consider:

- ❑ Redoc.ly Enterprise Edition https://redoc.ly/why-enterprise

Redocly

Thank you

Special thanks to Anuradha Kulkarni, Sami Benbourenane, and Brian Kinkade for giving us this opportunity and helping us along the way.

This has been a great experience that will help us as we move to the next chapters of our professional careers.

Questions?

comments , concerns?