cURL is not an API client

$ whoami

Evgeni Golov

Software Engineer at Red Hat

One-time Consultant at Red Hat

Debian and Grml Developer

♥ FOSS ♥

♥ automation ♥

background

- customer support request
- script worked with Katello version X
- but stopped when Katello was upgraded to X+1
- debugging the issue was tricky
- let’s share the wisdom!

the script

```
_URL="https://foreman.example.com"
_CCVID=2
_ENVID=1
_NAME="exapp01"

curl -H "Accept:application/json,version=2" \
     -H "Content-Type:application/json" \
     -X PUT -s -k -u admin:changeme \
     -d "{\"organization_id\": 1,
          \"included\":{\"search\":[\"name=$_NAME\"]},
          \"excluded\":[],\"content_view_id\":$_CCVID,
          \"environment_id\":$_ENVID}" \
     $_URL/api/hosts/bulk/environment_content_view
```

the error

```
{
  "displayMessage":
    "Unsupported query object: [\"name = exapp01\"]!",
  "errors":
    ["Unsupported query object: [\"name = exapp01\"]!"]
}
```

Clearly the API didn’t like what we sent, but it looks correct on the first glance?

And who is to blame? Did the API break? Or did it just got stricter and the script was just accidentaly working before?

the debugging

- Foreman documents its API using Apipie
- HTML version of the documentation is served at `https://foreman.example.com/apidoc/`
- Writing API clients in Ruby is very simple by using Apipie Bindings
- The bindings will also validate your data!
- So instead of trying to fix the original script, let's rewrite it?

- According to the API documentation, `/api/hosts/bulk/environment_content_v` is a call to the `environment_content_view` ac of the `hosts_bulk_actions` resource.
- In Ruby:

```
api.resource(:hosts_bulk_actions)
   .call(:environment_content_view, …)
```

With the required setup and data, we get:

```
api = ApipieBindings::API.new(
        {:uri => 'https://foreman.example.com',
         :username => 'admin', :password => 'changeme',
         :api_version => '2'})

data = {"organization_id": 1, "environment_id": 1,
        "included": {"search": ["name = exapp01"]},
        "excluded": [],
        "content_view_id": 2}

api.resource(:hosts_bulk_actions)
   .call(:environment_content_view, data)
```

```
ApipieBindings::InvalidArgumentTypesError:
  excluded - Hash was expected
```

That’s… a *different* error?

Well, let's read the API documentation for `environment_content_view` again:

- `excluded`, *required*, Validations: Hash
- `excluded[ids]`, *optional*, List of host ids to exclude and not run an action on, Validations: Must be an array of any type

```
api = ApipieBindings::API.new(
        {:uri => 'https://foreman.example.com',
         :username => 'admin', :password => 'changeme',
         :api_version => '2'})

data = {"organization_id": 1, "environment_id": 1,
        "included": {"search": ["name = exapp01"]},
        "excluded": {},
        "content_view_id": 2}

api.resource(:hosts_bulk_actions)
   .call(:environment_content_view, data)
```

This passes the Apipie Bindings validation, but would still raise

```
ArgumentError: Unsupported query object: ["name = exapp01"]
```

when executed.

The API documentation for *included* says:

- `included`, *required*, Validations: Hash
- `included[search]`, *optional*, Search string for hosts to perform an action on, Validations: **String**
- `included[ids]`, *optional*, List of host ids to perform an action on, Validations: Must be an array of any type

Oh, the *search* needs to be a String? Let's try that!

```
api = ApipieBindings::API.new(
        {:uri => 'https://foreman.example.com',
         :username => 'admin', :password => 'changeme',
         :api_version => '2'})

data = {"organization_id": 1, "environment_id": 1,
        "included": {"search": "name = exapp01"},
        "excluded": {},
        "content_view_id": 2}

api.resource(:hosts_bulk_actions)
   .call(:environment_content_view, data)
```

```
{
  "id"=>"7dcce620-b821-4ffd-987e-54dc1af98e28",
  "label"=>"Actions::BulkAction",
  "pending"=>true, "action"=>"Bulk action",
  "username"=>"admin",
  "started_at"=>"2019-01-28 11:34:46 UTC", "ended_at"=>nil,
  "state"=>"planned", "result"=>"pending", "progress"=>0.0,
  "input"=>{…}, "output"=>{},
  "humanized"=>{"action"=>"Bulk action", "input"=>nil,
                "output"=>nil, "errors"=>[]},
  "cli_example"=>nil
}
```

That’s the API telling us it created a task to update the host.

the fixed script

```
_URL="https://foreman.example.com"
_CCVID=2
_ENVID=1
_NAME="exapp01"

curl -H "Accept:application/json,version=2" \
     -H "Content-Type:application/json" \
     -X PUT -s -k -u admin:changeme \
     -d "{\"organization_id\": 1,
          \"included\":{\"search\":\"name=$_NAME\"},
          \"excluded\":{},\"content_view_id\":$_CCVID,
          \"environment_id\":$_ENVID}" \
    $_URL/api/hosts/bulk/environment_content_view
```

(The *excluded* mistake would raise "`no implicit conversion of Symbol into Integer`")

Apipie/Ruby benefits

- no need for passing JSON-related headers (cURL wants to improve that)
- no need to pass the *right* request type (PUT instead of POST)
- data validation (at least sometimes)
- easier to generate the data (it's just a Hash) and read the response
- IMHO nicer to read than double-escaped shell ;-)

Python?

- Python version of Apipie Bindings is under development: Apypie
- Please test and report bugs :)

Alternatives?

Yes, **Swagger**!

- Supports more language bindings
- Has a nicer doc viewer Swagger UI
- Slightly more complicated DSL to describe the API

Shell helpers?

- Instead of wrangling JSON by hand, use `jo`:

```
$ jo organization_id=1 content_view_id=2 \
  environment_id=1 \
  excluded={} 'included[search]=name = exapp01'
{"organization_id":1,"content_view_id":2,
 "environment_id":1,
 "excluded":{},"included":{"search":"name = exapp01"}}
```

- Instead of grepping JSON, use `jq`:

```
$ jo 'included[search]=name = exapp01' | jq .included.search
"name = exapp01"
```

```
_URL="https://foreman.example.com"
_CCVID=2 _ENVID=1
_NAME="exapp01"

jo organization_id=1 content_view_id=$_CVID \
    environment_id=$_ENVID \
    excluded={} "included[search]=name=$_NAME" \
    > data.json

curl -H "Accept:application/json,version=2" \
     -H "Content-Type:application/json" \
     -X PUT -s -k -u admin:changeme \
     -d @data.json \
     $_URL/api/hosts/bulk/environment_content_view
```

Thanks!

 evgeni@golov.de

 die-welt.net

 @zhenech

@zhenech@chaos.social

 @evgeni

 zhenech