




GraphQL – A New Method for Instant Data Insights on Haystack

Patrick Coffey – WideSky

Patrick Coffey
CTO @ WideSky

 [in/patrickcoffeyAus](https://www.linkedin.com/in/patrickcoffeyAus)

Agenda

1. Haystack API today
2. GraphQL
3. GraphQL on Haystack Examples
4. Demos

Haystack API Today

May 13-15, 2019



API clients



API

Haystack Server



Tagging



Time-series
Data



Application
users

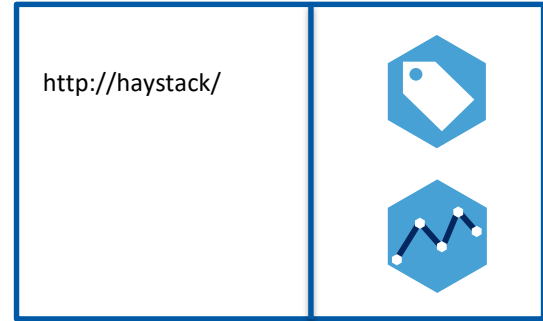
Integrated Systems



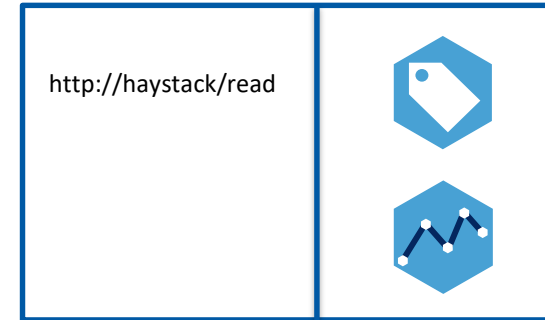
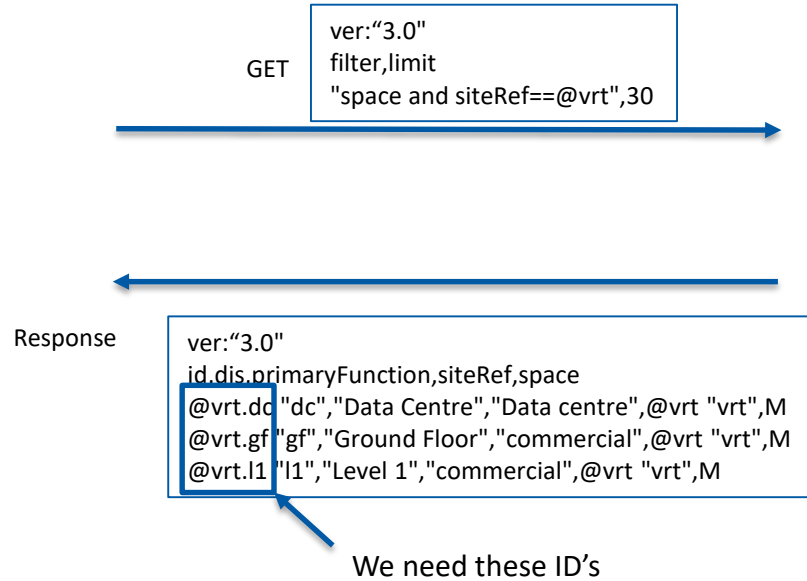
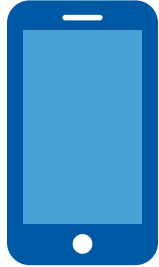
Haystack REST API Example



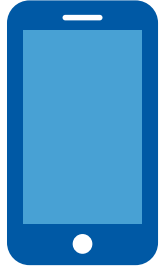
What was yesterday's max power readings for each space at the head office?



1: Find Spaces



2: Find points



For each ID

@vrt.dc
@vrt.gf
@vrt.l1

GET

```
ver:"3.0"  
filter,limit  
"point and his and power and equipRef->spaceRef==@vrt.dc",30
```

http://haystack/read



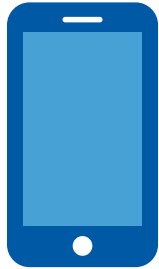
Response

```
ver:"3.0"  
id,dis,elec,equipRef,his,kind,point,power,sensor,siteRef,tz,unit  
"@vrt.dc_ac1.elecPower"elecPower,"Active Power Total",M,@vrt.dc_ac1  
"dc_ac1",M,"Number",M,M,M,@vrt "vrt","GMT-10","kW"  
"@vrt.dc_ups.elecPower"elecPower,"Active Power Total",M,@vrt.dc_ups  
"dc_ups",M,"Number",M,M,M,@vrt "vrt","GMT-10","kW"
```

More ID's



3: Query history and compute max()



For each ID

vrt.dc

@vrt.dc_ac1.elecPower
@vrt.dc_ups.elecPower

vrt.gf

@vrt.gf.ac1
@vrt.gf.ac2
@vrt.gf.gl1
@vrt.gf.gp1
@vrt.gf.gp2
@vrt.gf.hw1

vrt.l1

@vrt.gf.ac1
@vrt.gf.ac2
@vrt.gf.gl1
@vrt.gf.gl2
@vrt.gf.gp1

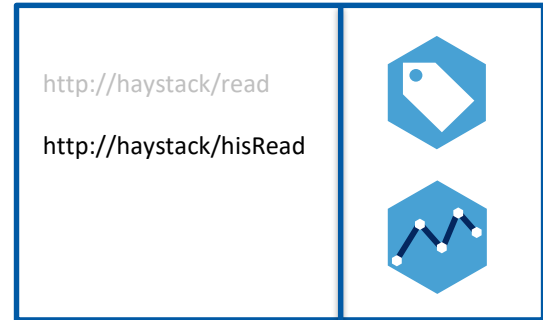
...

GET

```
ver:"3.0"  
id,range  
@vrt.dc_ac1.elecPower,"yesterday"
```

Response

```
ver:"3.0" hisEnd:2019-05-04T00:00:00+10:00 GMT-10  
hisStart:2019-05-03T00:00:00+10:00 GMT-10  
id:@vrt.dc_ac1.elecPower  
ts,val  
2019-05-03T00:00:00+10:00 GMT-10 0.097kW  
2019-05-03T00:01:00+10:00 GMT-10 0.653kW  
2019-05-03T00:02:00+10:00 GMT-10 0.631kW  
2019-05-03T00:03:00+10:00 GMT-10 0.482kW  
2019-05-03T00:04:00+10:00 GMT-10 0.544kW  
...
```



Client computes max() of 1440 records

4: Display results



vrt.dc

@vrt.dc_ac1.elecPower 1.1
@vrt.dc_ups.elecPower 1.2

vrt.gf

@vrt.gf.ac1 1.3
@vrt.gf.ac2 0.1
@vrt.gf.gl 0.5
@vrt.gf.gp1 1.5
@vrt.gf.gp2 2.2
@vrt.gf.hw1 4.6

vrt.l1

@vrt.gf.ac1 7.9
@vrt.gf.ac2 0.6
@vrt.gf.gl1 0.8
@vrt.gf.gl2 1.2
@vrt.gf.gp1 1.5

...

Haystack REST API



- Simple data exchange
 - Full data fidelity
 - All timeseries data
- Plenty of open source tools
- Real-time monitoring and control



- Back-and-forth communications
- Data Over-Fetching
- Hard for non-devs to build insights
- Underutilized Haystack servers

Our API Needs

- Build insights from the browser
- Simple to learn
- Nested query
- Leverage existing server capabilities

Why didn't you just use...

SPARQL

Cypher

SQL

Extend REST ops

Client side scripts

Server side scripts

RDF

OData

OWL

GraphQL

May 13-15, 2019






- Specification open-sourced by Facebook 2015
- A query language for APIs
- Span many resources
- Type system - server describes what's possible
- Significant open source community



POST

```
query {  
  allPeople(first: 1) {  
    people {  
      name  
      gender  
    }  
  }  
}
```

http://starwars/graphql	
---	---



```
{  
  "data": {  
    "allPeople": {  
      "people": [  
        {  
          "name": "Luke Skywalker",  
          "gender": "male"  
        }  
      ]  
    }  
  }  
}
```

Response

Our Approach

- Simple static schema
- Leverage nesting capability
- Expose timeseries functions to user
- Complement Haystack REST API

Examples

What are the tags on entity x?

```
▼ {
  ▼ "data": {
    ▼ "haystack": {
      ▼ "entity": {
        ▼ "tags": [
          {
            "name": "dis",
            "value": "Active Energy Import"
          },
          {
            "name": "elec",
            "value": "marker"
          },
          {
            "name": "energy",
            "value": "marker"
          },
          {
            "name": "equipRef",
            "value": "782d7bd4-6bcc-11e9-838a-0242ac120004"
          },
          {
            "name": "fqname",
            "value": "vrt.gf_inc.elecEnergy"
          },
          {
            "name": "his",
            "value": "marker"
          },
          {
            "name": "hisTotalized",
            "value": "marker"
          },
          {
            "name": "id",
            "value": "b97b1bf0-6bcc-11e9-85a6-0242ac120004"
          }
        ]
      }
    }
  }
}
```

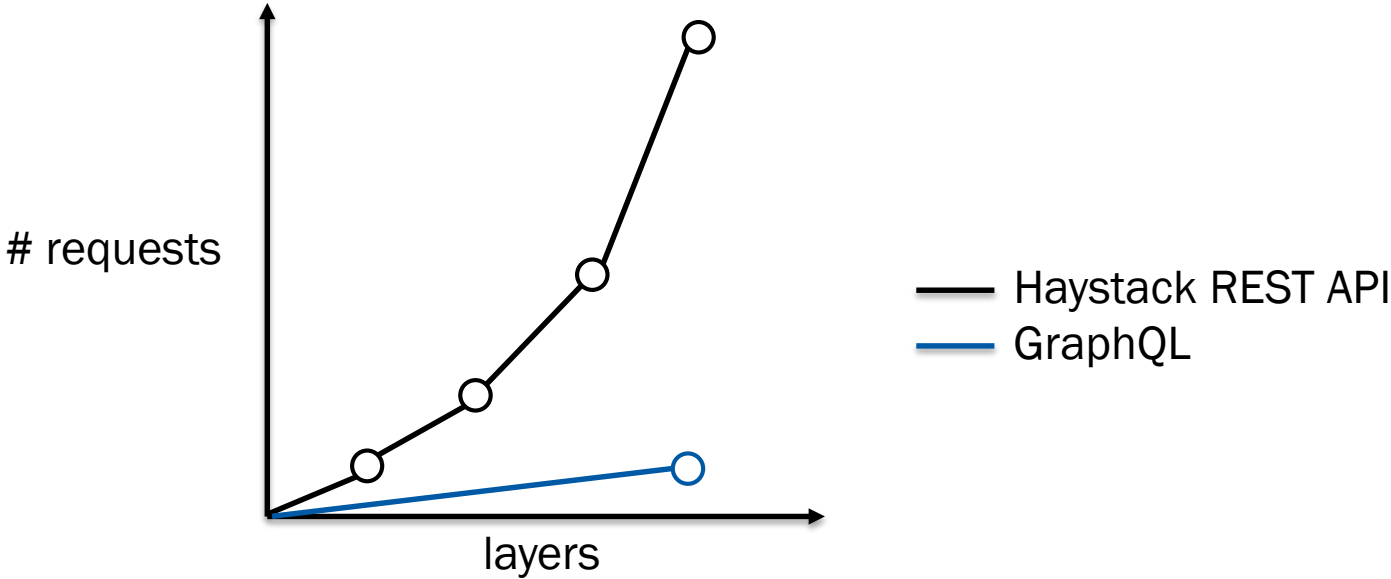
What was yesterday's history for entity x?

```
▼ {  
▼   "data": {  
▼     "haystack": {  
▼       "entity": {  
▼         "history": {  
▼           "timeSeries": [  
▼             {  
▼               "dataPoints": [  
▼                 {  
▼                   "time": 1557288900000,  
▼                   "value": "241863.798"  
▼                 },  
▼                 {  
▼                   "time": 1557289800000,  
▼                   "value": "241866.842"  
▼                 },  
▼                 {  
▼                   "time": 1557290700000,  
▼                   "value": "241869.807"  
▼                 },  
▼                 {  
▼                   "time": 1557291600000,  
▼                   "value": "241872.884"  
▼                 },  
▼                 {  
▼                   "time": 1557292500000,
```

Find all spaces at the head office

```
{
  "data": {
    "haystack": {
      "search": {
        "count": 3,
        "entity": [
          {
            "id": "6dee3c9e-6bcc-11e9-8329-0242ac120004",
            "description": "Data Centre"
          },
          {
            "id": "6e056946-6bcc-11e9-832a-0242ac120004",
            "description": "Ground Floor"
          },
          {
            "id": "6e2440d2-6bcc-11e9-832b-0242ac120004",
            "description": "Level 1"
          }
        ]
      }
    }
  }
}
```

Nesting



For each space, get yesterday's power meter data.

```
{
  "data": {
    "haystack": {
      "search": {
        "entity": [
          {
            "description": "Data Centre",
            "search": {
              "history": {
                "timeSeries": [
                  {
                    "dataPoints": [
                      {
                        "time": 1557289260000,
                        "value": "0.09"
                      },
                      {
                        "time": 1557289320000,
                        "value": "0.09"
                      },
                      {
                        "time": 1557289380000,
                        "value": "0.09"
                      },
                      {
                        "time": 1557289440000,
```

Add description from equipRef and compute max() for each



```
{
  "data": {
    "haystack": {
      "search": {
        "entity": [
          {
            "description": "Data Centre",
            "search": {
              "entity": [
                {
                  "refEntity": {
                    "description": "Air Conditioner 1"
                  },
                  "units": "kW",
                  "history": {
                    "select": {
                      "timeSeries": [
                        {
                          "dataPoints": [
                            {
                              "time": 1557360000000,
                              "value": "1.123"
                            }
                          ]
                        }
                      ]
                    }
                  }
                }
              ]
            }
          },
          {
            "refEntity": {
              "description": "UPS"
            },
            "units": "kW",
            "history": {
              "select": {
                "timeSeries": [
                  {
                    "dataPoints": [
                      {
                        "time": 1557337320000,
                        "value": "1.297"
                      }
                    ]
                  }
                ]
              }
            }
          }
        ]
      }
    }
  }
}
```


Demos

May 13-15, 2019



1: Developer IDE - GraphiQL

GraphiQL



Prettify

History

< Docs

```
1 query {  
2  
3 }  
4  
5
```



2: Data interaction in Grafana

Home - Grafana

localhost:8085/?orgId=1

Home Dashboard

Install Grafana

Create your first data source

New dashboard

Invite your team

Install apps & plugins

Starred dashboards

Recently viewed dashboards

Installed Apps

None installed. [Browse Grafana.com](#)

Installed Panels

Breadcrumb v1.0.0	Up to date
Cal-HeatMap v0.0.3	Up to date
Clock v0.0.8	Up to date
Discrete v0.0.7	Up to date
Histogram v0.1.6	Up to date
Pie Chart v1.1.4	Up to date
Radar Graph v1.4.1	Up to date
Realtime Control v0.1.0	Up to date
Worldmap Panel v0.0.20	Up to date

Installed Datasources

Widesky v0.2.5	Up to date
----------------	------------

Summary

- Benefits:
 - Users become data superheroes from the browser
 - Haystack adoption
 - Other use cases – RBAC rules
- Extensions:
 - Auto unit conversion
 - Timeseries functions – (select, aggregate, transform...)
 - Entity filtering based on history
- Haystack v4?

Thank You

Come and say G'Day! 🤝