



Improving the FERRY Meta-Data Management System

Antonio Segura

SIST Meeting

Monday, June 10, 2019

Project Overview

Improve the Frontier Experiments RegistRY (FERRY) meta-data management system

- Refactor APIs (in the Go programming language) to reflect a more modular and standardized production of APIs for increased readability, additional conciseness, and overall consistent functionality
- Explore the visual representation of the FERRY database (currently using the Superset visualization tool by Apache)
- Develop additional functionality in Superset, regarding certificate authorization, relevant database views, and explore algebraic query writing- another layer of abstraction from SQL query writing for increased user-friendliness

Current Progress - Superset

← → ↺ 🏠 <https://127.0.0.1:8443/getGroupName?gid=8737>

🌐 Timecard 📧 Gmail

JSON Raw Data Headers

Save Copy Collapse All Expand All

▼ 0:

groupname: "minervag"

FERRY Security Sources Manage Charts Dashboards SQL Lab + New

Untitled Query

Database: postgresql ferry

Select a schema (2)

See table schema (0 in)

Select table

```
1 select full_name, dn from users join user_certificates using(uid)
```

Run Query Save Query Share Query LIMIT 100000 parameters 00:00:02.41

Results Query History

Explore CSV Clipboard Filter Results

full_name	dn
Andreas Kronfeld	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andreas Kronfeld/CN=UID:ask
Andrew Baden	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andrew Baden/CN=UID:drew
Eileen Berman	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Eileen Berman/CN=UID:berman
Brenna Flaughter	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Brenna Flaughter/CN=UID:brenna
Elizabeth Buckley-Geer	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Elizabeth Buckley-Geer/CN=UID:buckley
Joel Butler	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=jbutter/CN=527378/CN=Joel Butler
Joel Butler	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Joel Butler/CN=UID:butler
Frank Chlebana	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Frank Chlebana/CN=UID:chlebana
Stoyan Stoynev	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=stoynev/CN=597450/CN=Stoyan Emilov Stoynev
Stoyan Stoynev	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Stoyan Stoynev/CN=UID:stoyan

Current Progress – Superset (cont.)

FERRY Security Sources Manage Charts Dashboards SQL Lab

CMSUserInfo ☆

New

Edit dashboard

CMSUserFilter

Time range
No filter

Unix Name
Select [Unix Name]

Full Name
Select [Full Name]

CERN User Name
Select [CERN User Name]

CMSUserInfoList

Show 50 entries

uname	full_name	cern_uname	home_dir	shell
3wolf3	Evan Wolfe	ewolfe	/uscms/home/3wolf3	/bin/tcsh
aagarwal	Ayush Agarwal	ayagarwa	/uscms/home/aagarwal	/bin/tcsh
aakpinar	Alp Akpinar	aakpinar	/uscms/home/aakpinar	/bin/tcsh
aapyan2	Aram Apyan	arapyan	/uscms/home/aapyan2	/bin/tcsh
aarond	Aaron Dominguez	NA	/uscms/home/aarond	/bin/tcsh
aarulana	Adhav Arulanandan	NA	/uscms/home/aarulana	/bin/tcsh
aattikis	Alexandros Attikis	attikis	/uscms/home/aattikis	/bin/tcsh

Previous 1 2 3 4 5 ... 23 Next

CMSUserGroupsList

Show 50 entries

uname	full_name	name	is_leader
butler	Joel Butler	us_cms	false
butler	Joel Butler	lnuij	false
butler	Joel Butler	lpclij	false
butler	Joel Butler	lpccsa14	false
stoyan	Stoyan Stoynev	us_cms	false
stoyan	Stoyan Stoynev	lpchzg	false
stoyan	Stoyan Stoynev	lpcmuon	false
cheung	Harry Cheung	us_cms	false
cheung	Harry Cheung	lpclij	false
banerjee	Sudeshna Banerjee	us_cms	false
lammel	Stephan Lammel	us_cms	false
lammel	Stephan Lammel	lpccvmfs	false

Previous 1 2 3 4 5 ... 13

Next

CMSUserStorageQuotaList

Show 50 entries

uname	full_name	name	path	TB	valid_until
3wolf3	Evan Wolfe	EOS	/eos/uscms/store/user/3wolf3	2	
3wolf3	Evan Wolfe	NOBACKUP3	/uscms_data/d3/3wolf3	0	
aagarwal	Ayush Agarwal	EOS	/eos/uscms/store/user/aagarwal	2	
aagarwal	Ayush Agarwal	NOBACKUP3	/uscms_data/d3/aagarwal	0	
aakpinar	Alp Akpinar	NOBACKUP3	/uscms_data/d3/aakpinar	0.1	
aakpinar	Alp Akpinar	NOBACKUP2	/uscms_data/d2/aakpinar	0.1	
aakpinar	Alp Akpinar	EOS	/eos/uscms/store/user/aakpinar	0.9	
aapyan2	Aram Apyan	EOS	/eos/uscms/store/user/aapyan2	2	
aapyan2	Aram Apyan	NOBACKUP3	/uscms_data/d3/aapyan2	0.1	
aattikis	Alexandros Attikis	NOBACKUP3	/uscms_data/d3/aattikis	0.1	
aattikis	Alexandros Attikis	EOS	/eos/uscms/store/user/aattikis	10.9	
abakshi	Amandeep Singh Bakshi	NOBACKUP3	/uscms_data/d3/abakshi	0.1	
abakshi	Amandeep Singh Bakshi	EOS	/eos/uscms/store/user/abakshi	2	
abakshi	Amandeep Singh Bakshi	EOS	/eos/uscms/store/user/abakshi	2	

Previous 1 2 3 4 5 ... 15 Next

CMSUserDNList

Show 50 entries

uname	full_name	dn
3wolf3	Evan Wolfe	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewolfe/CN=739620/CN=Evan Michael Wolfe
3wolf3	Evan Wolfe	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Evan Wolfe/CN=UID:3wolf3
aagarwal	Ayush Agarwal	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Ayush Agarwal/CN=UID:aagarwal
aagarwal	Ayush Agarwal	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ayagarwa/CN=828978/CN=Ayush Bhushan Agarwal
aakpinar	Alp Akpinar	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Alp Akpinar/CN=UID:aakpinar
aakpinar	Alp Akpinar	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=aakpinar/CN=834208/CN=Alp Akpinar
aapyan2	Aram Apyan	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=arapyan/CN=455306/CN=Aram Apyan

Previous 1 2 3 4 5 ... 20 Next

Current Progress – Superset (cont.)

Expectations of Outcomes:

- To further understanding of SQL query language and relational database management
- To develop knowledge on web development using React.js, HTML, and CSS.
- Learn further cryptographic applications and practical usage via developing a certificate authorization system
- Create a significant advancement on the Superset data visualization application and add said progress to the open source contributions.

Current Progress - APIs

```
4835 func getGroupsInAffiliationLegacy(w http.ResponseWriter, r *http.Request) {
4836     startTime := time.Now()
4837     w.Header().Set("Content-Type", "application/json; charset=UTF-8")
4838     q := r.URL.Query()
4839     unitName := q.Get("unitname")
4840
4841     if unitName == "" {
4842         log.WithFields(QueryFields(r, startTime)).Error("No unit name specified in http query.")
4843         fmt.Fprintf(w, "{ \"ferry_error\": \"No unitname specified.\" }")
4844         return
4845     }
4846     lastupdate, parseerr := stringToParsedTime(strings.TrimSpace(q.Get("last_updated")))
4847     if parseerr != nil {
4848         log.WithFields(QueryFields(r, startTime)).Error("Error parsing provided update time: " + parseerr.Error())
4849         fmt.Fprintf(w, "{ \"ferry_error\": \"Error parsing last_updated time. Check ferry logs. If provided, it should be an integer representing an epoch time.\" }")
4850         return
4851     }
4852
4853     var unitId int
4854     checkerr := DBptr.QueryRow(`select unitid from affiliation_units where name=$1`, unitName).Scan(&unitId)
4855     switch {
4856     case checkerr == sql.ErrNoRows:
4857         // set the header for success since we are already at the desired result
4858         fmt.Fprintf(w, "{ \"ferry_error\": \"Affiliation unit does not exist.\" }")
4859         log.WithFields(QueryFields(r, startTime)).Error("unit " + unitName + " not found in DB.")
4860         return
4861     case checkerr != nil:
4862         w.WriteHeader(http.StatusNotFound)
4863         fmt.Fprintf(w, "{ \"ferry_error\": \"Database error.\" }")
4864         log.WithFields(QueryFields(r, startTime)).Error("deleteUser: Error querying DB for unit " + unitName + ".")
4865         return
4866     default:
4867
4868         rows, err := DBptr.Query(`select gid, groups.name, groups.type, aug.is_primary
4869                                from affiliation_unit_group as aug
4870                                join groups on aug.groupid = groups.groupid
4871                                where aug.unitid=$1 and (aug.last_updated>=$2 or $2 is null)`,
4872                                unitId, lastupdate)
4873         if err != nil {
4874             defer log.WithFields(QueryFields(r, startTime)).Error(err.Error())
4875             w.WriteHeader(http.StatusNotFound)
4876             fmt.Fprintf(w, "{ \"ferry_error\": \"Error in DB query.\" }")
4877             return
4878         }
4879
4880         defer rows.Close()
4881         type jsonout struct {
4882             Gid      int    `json:"gid"`
4883             GName     string `json:"name"`
4884             GType     string `json:"type"`
4885             GPrimary  bool   `json:"is_primary"`
4886         }
4887         var Entry jsonout
4888         var Out []jsonout
4889
4890         for rows.Next() {
4891             var tmpGID int
4892             var tmpGName, tmpGType string
4893             var tmpGPrimary bool
4894             rows.Scan(&tmpGID, &tmpGName, &tmpGType, &tmpGPrimary)
4895             Entry.Gid = tmpGID
4896             Entry.GName = tmpGName
4897             Entry.GType = tmpGType
4898             Entry.GPrimary = tmpGPrimary
4899             Out = append(Out, Entry)
4900         }
4901         var output interface{}
4902         if len(Out) == 0 {
4903             type jsonerror struct {
4904                 Error string `json:"ferry_error"`
4905             }
4906             var queryErr []jsonerror
4907             queryErr = append(queryErr, jsonerror{"This affiliation unit has no groups."})
4908             log.WithFields(QueryFields(r, startTime)).Error("This affiliation unit has no groups.")
4909             output = queryErr
4910         } else {
4911             log.WithFields(QueryFields(r, startTime)).Info("Success!")
4912             output = Out
4913         }
4914         jsonoutput, err := json.Marshal(output)
4915         if err != nil {
4916             log.WithFields(QueryFields(r, startTime)).Error(err.Error())
4917         }
4918         fmt.Fprintf(w, string(jsonoutput))
4919     }
4920 }
```


Current Progress – APIs (cont.)

```
334 func getGroupsInAffiliationUnit(c APIContext, i Input) (interface{}, []APIError) {
335     var apiErr []APIError
336
337     unitid := NewNullAttribute(UnitID)
338
339     checkerr := c.DBtx.QueryRow(`select unitid from affiliation_units where name=$1`, i[UnitName]).Scan(&unitid)
340     if checkerr != nil && checkerr != sql.ErrNoRows {
341         log.WithFields(QueryFields(c.R, c.StartTime)).Error(checkerr)
342         apiErr = append(apiErr, DefaultAPIError(ErrorDbQuery, nil))
343         return nil, apiErr
344     }
345     if !unitid.Valid {
346         apiErr = append(apiErr, DefaultAPIError(ErrorDataNotFound, UnitName))
347         return nil, apiErr
348     }
349
350     rows, checkerr := c.DBtx.Query(`select gid, groups.name, groups.type, aug.is_primary
351 from affiliation_unit_group as aug
352 join groups on aug.groupid = groups.groupid
353 where aug.unitid=$1 and (aug.last_updated>=$2 or $2 is null)`,
354 unitid, i[LastUpdated])
355     if checkerr != nil {
356         log.WithFields(QueryFields(c.R, c.StartTime)).Error(checkerr)
357         apiErr = append(apiErr, DefaultAPIError(ErrorDbQuery, nil))
358         return nil, apiErr
359     }
360
361     defer rows.Close()
362     type jsonentry map[Attribute] interface{}
363     out := make([]jsonentry, 0)
364
365     for rows.Next() {
366         row := NewMapNullAttribute(GID, GroupName, GroupType, Primary)
367         rows.Scan(row[GID], row[GroupName], row[GroupType], row[Primary])
368         entry := jsonentry {
369             GID: row[GID].Data,
370             GroupName: row[GroupName].Data,
371             GroupType: row[GroupType].Data,
372             Primary: row[Primary].Data,
373         }
374         out = append(out, entry)
375     }
376     return out, nil
377 }
```

Current Progress – APIs (cont.)

Expectations of Outcomes:

- Furthered understanding of better Software Engineering practices
- Increased knowledge of Go, Python, and SQL programming languages
- Practical understanding of management of a database system
- Overall increased knowledge across multiple fields of Computer Science and various programming languages