

LogHarbour: logging - Task #17935

WSC: show_highprilog()

15/02/2024 10:04 AM - Sachin Divekar

Status:	Closed	Start date:	18/03/2024
Priority:	Normal	Due date:	18/03/2024
Assignee:	Aniket Shinde	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	Sprint 15 Mar		
Description			
Related issues:			
Follows LogHarbour: logging - Task #18066: Go lib: GetLogs()			Closed 15/03/2024

Associated revisions

Revision 8c51f3c7 - 08/03/2024 01:44 PM - remiges-aniket

refs: #17935 - refactor: added library function to carry out request

Revision c7553499 - 08/03/2024 01:46 PM - Aniket Shinde

Merge pull request #23 from remiges-aniket/wsc

refs: #17935 - refactor: added library function to carry out request

Revision 5fa1c33b - 09/03/2024 05:00 PM - remiges-aniket

refs: #17935 - fix: search_after filter

Revision 6fe87319 - 12/03/2024 02:10 PM - remiges-aniket

refs: #17935 - implemented test cases for show_highprilog()

Revision db46d4d3 - 12/03/2024 02:11 PM - Aniket Shinde

Merge pull request #31 from remiges-aniket/wsc

refs: #17935 - implemented test cases for show_highprilog()

History

#1 - 18/02/2024 03:15 AM - Shuvam Misra

- Target version set to Sprint 29 Feb

#2 - 21/02/2024 01:12 PM - Aniket Shinde

- Assignee set to Aniket Shinde

- Status changed from New to In Progress

Going through Elasticsearch.

#3 - 21/02/2024 06:00 PM - Aniket Shinde

- File clipboard-202402211800-nc86o.png added

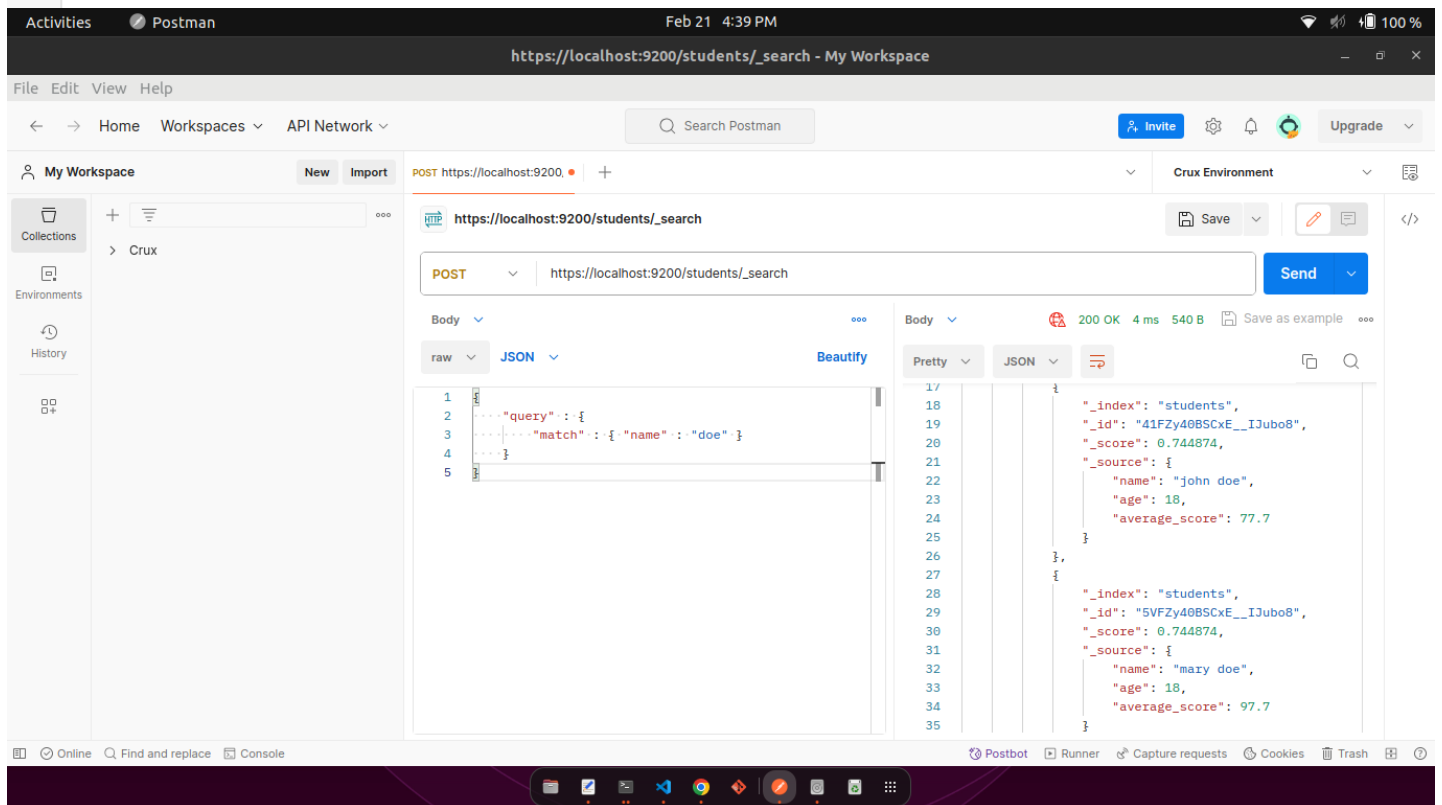
- File clipboard-202402211758-woppl.png added

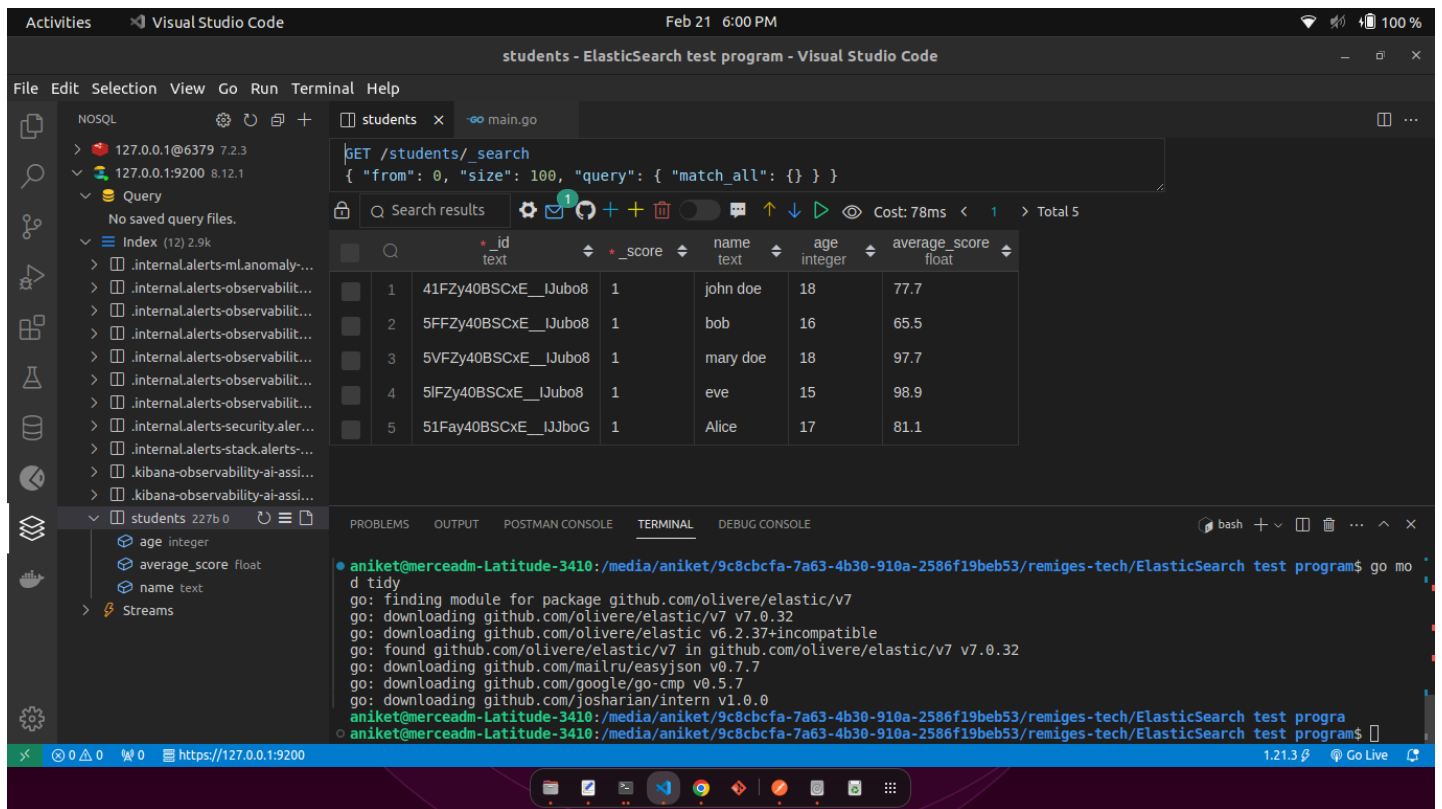
- File clipboard-202402211758-ynu6e.png added

installed elasticsearch and tested with dummy data:

```
← → ↺ 🏠 Not secure https://localhost:9200

1 // 20240221150932
2 // https://localhost:9200/
3
4 {
5   "name": "merceadm-Latitude-3410",
6   "cluster_name": "elasticsearch",
7   "cluster_uuid": "X7mbxSndRFm9hXgkFP2v7w",
8   "version": {
9     "number": "8.12.1",
10    "build_flavor": "default",
11    "build_type": "tar",
12    "build_hash": "6185ba65d27469afabc9bc951cded6c17c21e3f3",
13    "build_date": "2024-02-01T13:07:13.727175297Z",
14    "build_snapshot": false,
15    "lucene_version": "9.9.2",
16    "minimum_wire_compatibility_version": "7.17.0",
17    "minimum_index_compatibility_version": "7.0.0"
18  },
19  "tagline": "You Know, for Search"
20 }
```





#4 - 23/02/2024 08:47 PM - Aniket Shinde

Update: implemented elasticsearch db and golang connection, started writing API

#5 - 28/02/2024 06:39 PM - Aniket Shinde

- % Done changed from 0 to 30

added raw api to do the pagination.

working on business logic and demo seed data to do the testing

#6 - 01/03/2024 05:29 PM - Shuvam Misra

- Target version changed from Sprint 29 Feb to Sprint 15 Mar

#7 - 01/03/2024 05:59 PM - Shuvam Misra

- Follows Task #18066: Go lib: GetLogs() added

- Start date set to 18/03/2024

- Due date set to 18/03/2024

#8 - 03/03/2024 12:56 PM - Aniket Shinde

- Status changed from In Progress to On Hold

Working on Go lib, hence putting this task on hold

Ref (WhatsApp group msg from Shuvam on(01/03/2024))

#9 - 08/03/2024 02:26 PM - Aniket Shinde

- Status changed from On Hold to In Progress

implemented library function GetLogs() to carry out request

#10 - 08/03/2024 06:03 PM - Shuvam Misra

- Status changed from In Progress to Testing

Important:

```
GET /logharbour/_search
{
  "from": 0,
  "size": 100,
  "query": {
    "match": {
      "_id": "fLhMI44BKYLHEbgtQwAR"
    }
  },
  "sort": [
    {
      "_doc": {
        "order": "desc"
      }
    },
    {
      "when": {
        "order": "desc"
      }
    }
  ]
}
```

created seed data for docker db test:

[logharbour_demo_data.es](https://github.com/remiges-tech/logharbour_demo_data.es)

#12 - 10/03/2024 05:17 PM - Aniket Shinde

Completed: written a demo data insert query that will be used for migration.

In Progress: Worked on Docker test with elastic search:

```
package schema_test

import (
    "fmt"
    "log"
    "os"
    "testing"
    "time"

    "github.com/elastic/elastic-transport-go/v8/elastictransport"
    "github.com/elastic/go-elasticsearch/v8"
    "github.com/gin-gonic/gin"
    "github.com/ory/dockertest/v3"
    "github.com/ory/dockertest/v3/docker"

    "github.com/remiges-tech/alya/service"
    "github.com/remiges-tech/alya/wscutils"
    "github.com/remiges-tech/logharbour/logharbour"
)

var r *gin.Engine

// var versionTable string = "schema_version_non_default"

func TestMain(m *testing.M) {

    // Initialize Docker pool to insure it close at the end
    pool, err := dockertest.NewPool("")
    if err != nil {
        log.Fatalf("Could not connect to Docker: %v", err)
    }
    fmt.Println("Initialize Docker pool")

    // Deploy the Elasticsearch container.
    resource, databaseUrl, err := deployElasticsearch(pool)
    if err != nil {
        log.Fatalf("Could not start resource: %v", err)
    }
    fmt.Println("Deploy the Elasticsearch container")
}
```

```

// ternMigrate(databaseUrl)
fmt.Println("tern migrate")

// Register routes.
r, err = registerRoutes(pool, databaseUrl)
if err != nil {
    log.Fatalf("Could not start resource: %v", err)
}
fmt.Println("Register routes")
// Run the tests.
exitCode := m.Run()

// Exit with the appropriate code.
if err := pool.Purge(resource); err != nil {
    log.Fatalf("Could not purge resource: %s", err)
}
fmt.Println("Exit with the appropriate code")

os.Exit(exitCode)

}

func deployElasticsearch(pool *dockertest.Pool) (*dockertest.Resource, string, error) {
    // Define the Elasticsearch Docker container configuration
    elasticsearchOptions := &dockertest.RunOptions{
        Repository: "docker.elastic.co/elasticsearch/elasticsearch",
        Tag:        "8.12.2", // Replace with the desired Elasticsearch version
        Cmd:        []string{"elasticsearch", "-E", "discovery.type=single-node"},
        PortBindings: map[docker.Port][]docker.PortBinding{
            "9200/tcp": {{HostIP: "", HostPort: ""}},
        },
    }

    // Pull the Elasticsearch image
    err := pool.Client.PullImage(docker.PullImageOptions{
        Repository: elasticsearchOptions.Repository,
        Tag:        elasticsearchOptions.Tag,
    }, docker.AuthConfiguration{})
    if err != nil {
        return nil, "", fmt.Errorf("failed to pull Elasticsearch image: %v", err)
    }

    // Start the Elasticsearch container
    resource, err := pool.RunWithOptions(elasticsearchOptions, func(config *docker.HostConfig) {
        // set AutoRemove to true so that stopped container goes away by itself
        config.AutoRemove = true
        config.RestartPolicy = docker.RestartPolicy{Name: "no"}
    })
    if err != nil {
        return nil, "", fmt.Errorf("failed to start Elasticsearch container: %v", err)
    }

    // Ensure that the Elasticsearch container is ready to accept connections
    // ctx := context.Background()
    err = pool.Retry(func() error {
        url := fmt.Sprintf("http://%s", resource.GetHostPort("9200/tcp"))
        fmt.Println(">>>>>URL:", url)
        // resp, err := http.Get(url)
        // if err != nil {
        //     fmt.Println("<<<<<", err)
        //     return err
        // }
        // defer resp.Body.Close()
        dbConfig := elasticsearch.Config{
            Addresses: []string{url},
            // Username:      appConfig.DBUser,
            // Password:      appConfig.DBPassword,
            // CertificateFingerprint: appConfig.CertificateFingerprint,
            // show request query logger
            Logger: &elastictransport.TextLogger{Output: log.Writer(), EnableRequestBody: true},
        }
        client, err := elasticsearch.NewTypedClient(dbConfig)
        if err != nil {
            return err
        }
    })
}

```

