Go Examples

This is an example in Go, please modify it to fit your environment, be sure to change the video input and output paths as well as the API_KEY and AWS_ACCESS and AWS_SECRET keys.

```
package main

import (
	"bytes"
	"encoding/json"
	"fmt"
	"io/ioutil"
	"net/http"
)

func main() {
	url := "https://streamengine.igolgi.com/api/v0/job"

	// Authentication credentials
	username := "INSERT_YOUR_API_KEY"
	password := "a"

	// Data to be sent in the POST request
	data := map[string]interface{}{
		"videoProfiles": []map[string]interface{}{
			{
				"output":              "s3://igolgi-se-out/go-testcwaudio3f.mp4",
				"width":               1920,
				"height":              1080,
				"video_bitrate_mode":  "cbr",
				"video_bitrate":       2301,
				"video_format":        "progressive",
				"video_framerate":     "1x",
				"audio_profiles":      "1",
			},
		},
		"audioProfiles": []map[string]interface{}{
```

```
		{
			"audio_bitrate":   256,
			"audio_channels":  2,
			"audio_codec":     "aac",
			"source_stream":   1,
		},
	},
	"master_variant_mode": false,
	"output_container":    "mp4",
	"video_codec":         "h.264",
	"input":               "s3://igolgi-se-in/GOPR8297-2.7K-30-fps.MP4",
	"auto_detelecine_flag": false,
	"cloud_credentials": map[string]interface{}{
		"input": map[string]string{
			"cloud_provider": "aws",
			"access_key":     "INSERT_YOUR_AWS_ACCESS_KEY",
			"secret_key":     "INSERT_YOUR_AWS_SECRET_KEY",
			"region":         "us-east-2",
		},
		"output": map[string]string{
			"cloud_provider": "aws",
			"access_key":     "INSERT_YOUR_AWS_ACCESS_KEY",
			"secret_key":     "INSERT_YOUR_AWS_SECRET_KEY",
			"region":         "us-east-2",
		},
	},
}

jsonData, err := json.Marshal(data)
if err != nil {
	fmt.Println("Error marshalling JSON:", err)
	return
}

req, err := http.NewRequest("POST", url, bytes.NewBuffer(jsonData))
if err != nil {
	fmt.Println("Error creating request:", err)
	return
}
```

```
	req.Header.Set("Content-Type", "application/json")
	req.SetBasicAuth(username, password)

	client := &http.Client{}
	resp, err := client.Do(req)
	if err != nil {
		fmt.Println("Error sending request:", err)
		return
	}
	defer resp.Body.Close()

	body, err := ioutil.ReadAll(resp.Body)
	if err != nil {
		fmt.Println("Error reading response:", err)
		return
	}

	fmt.Println("Response:", string(body))
}
```

save the script above as se_api_aws.go or similar

run
  go run se_api_aws.go

it will give print something like:
Response: {"_auto_generated_id_":**13455**,"cloud":"aws" ....

Remember your job number above and use it with the following script to check the status.

```
package main

import (
	"encoding/base64"
	"encoding/json"
	"fmt"
	"io/ioutil"
	"net/http"
	"os"
)
```

```
func main() {
    if len(os.Args) < 2 {
        fmt.Println("Please provide a job ID as an argument.")
        os.Exit(1)
    }

    jobID := os.Args[1]
    url := fmt.Sprintf("https://streamengine.igolgi.com/api/v0/job/%s", jobID)
    apiKey := "INSERT_YOUR_API_KEY" // Replace with your actual API key

    req, err := http.NewRequest("GET", url, nil)
    if err != nil {
        fmt.Println("Error creating request:", err)
        return
    }

    // Set Authorization header
    authHeader := base64.StdEncoding.EncodeToString([]byte(apiKey + ":"))
    req.Header.Set("Authorization", "Basic "+authHeader)

    client := &http.Client{}
    resp, err := client.Do(req)
    if err != nil {
        fmt.Println("Error sending request:", err)
        return
    }
    defer resp.Body.Close()

    body, err := ioutil.ReadAll(resp.Body)
    if err != nil {
        fmt.Println("Error reading response:", err)
        return
    }

    if resp.StatusCode == 200 {
        var data map[string]interface{}
        err = json.Unmarshal(body, &data)
        if err != nil {
            fmt.Println("Error parsing JSON:", err)
            return
```

```
    }

    fmt.Println("Job Status:")
    fmt.Println("------------------------")
    status := data["status"].(map[string]interface{})
    fmt.Printf("Job ID: %s\n", status["job_id"])
    fmt.Printf("State: %s\n", status["state"])
    fmt.Println("Progress:")
    fmt.Printf("  Transcode: %v%%\n", status["tcode_progress"])
    fmt.Printf("  Transfer: %v%%\n", status["xfer_progress"])
    fmt.Println("Time:")
    fmt.Printf("  Transcode: %v seconds\n", status["tcode_time"])
    fmt.Printf("  Transfer: %v seconds\n", status["xfer_time"])
    fmt.Println("Speed:")
    fmt.Printf("  Transcode: %v x\n", status["tcode_speed"])
    fmt.Printf("  Transfer: %v MB/s\n", status["xfer_speed"])
    fmt.Printf("Job Completed: %v\n", status["job_completed"])

    fmt.Println("\nJob Config:")
    fmt.Println("------------------------")
    config, err := json.MarshalIndent(data["config"], "", "  ")
    if err != nil {
        fmt.Println("Error formatting config:", err)
    } else {
        fmt.Println(string(config))
    }
} else {
    fmt.Printf("Failed to retrieve job. Status code: %d\n", resp.StatusCode)
    fmt.Printf("Response: %s\n", string(body))
}
}
```

Save the script above as se-api-status.go or similar.

Run
  go run se-api-status.go **13455**

You'll see a result similar to the one below. The job ID of status will be different than the submit job id. You can rerun the status until you see Job Completed: true.

Job Status:

Job ID: 4920

State: ready

Progress:

  Transcode: 0%

  Transfer: 0%

Time:

  Transcode: <nil> seconds

  Transfer: 78460.45 seconds

Speed:

  Transcode: -1 x

  Transfer: -1.0 MB/s

Job Completed: false

Job Config:

```
{
  "audioProfiles": [
    {
      "audio_bitrate": 256,
      "audio_channels": 2,
      "audio_codec": "aac",
      "source_stream": 1
    }
  ],
  "audio_volume": 100,
  "gop_length": 1,
  "h264_quality": "good",
  "input": "s3://igolgi-se-in/GOPR8297-2.7K-30-fps.MP4",
  "output_container": "mp4",
  "picture_transform": "none",
  "scte35_passthough": false,
  "segmented_output_dash": false,
  "segmented_output_hls": false,
  "separate_audio": false,
  "videoProfiles": [
    {
      "audio_profiles": "1",
      "height": 1080,
```

```
      "output": "s3://igolgi-se-out/go-testcwaudio3f.mp4",
      "video_bitrate": 2301,
      "video_bitrate_mode": "cbr",
      "video_framerate": "1x",
      "width": 1920
    }
  ],
  "video_aspect_ratio": "passthrough",
  "video_codec": "h.264"
}
```


On linux/Ubuntu if you have never used go.

You can run:

$ sudo apt update
$ sudo apt install golang-go

Revision #2
Created 16 July 2024 21:31:22 by Admin
Updated 25 July 2024 16:43:34 by Admin