|  |  |  |  |
| --- | --- | --- | --- |
| **RESTFUL SERVICE FOR CUT2IT** | | | **v1.0** |
| **Project** | CUT2IT | | |
| **Task Id** | CUT93 | **Description**  This task requires the RESTful services are working and functional. The purpose is to integrate the media and user RESTful services for use by the platform. | |
| **Subtask** | CUT94 | **Description**  Integrate the Media RESTful services to the platform.  We want to let application clients be able to update, insert, delete from media schema. | |

# **OVERVIEW**

In order to provide a good performance, satisfy robust and complicated queries for its services, Cut2it chose to deploy RESTFUL web services. The service API provided including the following aspect:

* Media: all information related to the media object such as meta data, license, rating, ranking
* User - information about the user(profile)
* Channel & Station (content will be added in later time)

**Restful Architecture**

Machine generated alternative text:
Database 
1433 
data request 
repose 
Entervise 
Server 
Go Service 
Web Server 
Enter prise 
App Server 

Restful API provides a communication between front end web application and the data that hosting behind the scene.

# **GO PAKAGES**

* **Router package:** Mux router from the Gorilla Web Toolkit.

<http://www.gorillatoolkit.org/pkg/mux>

<http://thenewstack.io/make-a-restful-json-api-go/>

* **GO version 1.4.2**

<https://golang.org/dl/>

* **Pure Go MSSQL driver for Go's database/sql package**

<https://github.com/denisenkom/go-mssqldb>

* **Git -** command integrated git forinstalling distributed app packages on git

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

# **GO ROUTES CONFIGURATION**

*Table 1: Media Service URI*

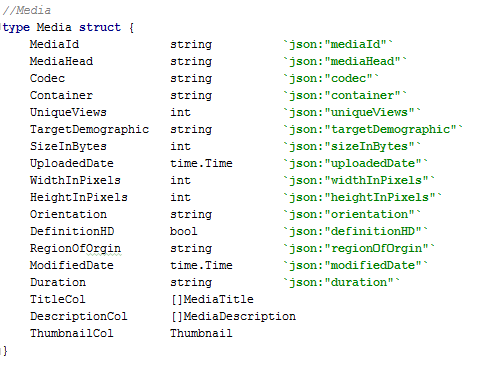
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **URI** | **Method** | **Description** | **Input** | **Output** | **GO Handler** |
| / | GET | Return home page of the system | n/a | Index.html |  |
| /medias | GET | Return a representation of all medias in the system | n/a | Medias collection |  |
| /medias/{mediaTitle} | GET | Return the representation of all media based on the media title. Search are exact match, no case sensitive | Requires header data: “mediaTitle” | Medias collection |  |
|  |  |  |  |  |  |
| /medias/insert | POST | Create a new media in the system, expect the representation of the media in the html body | Media, require data in html body | Media |  |
|  |  |  |  |  |  |
| /medias | PUT | Update a media resource | mediaId, require data in html body | n/a |  |
| /medias/{mediaId} | DELETE | Delete a media in the system | mediaId | n/a |  |
| /medias/insert | POST | Add a new media into the system | Form fields required  (detail below) |  | goAddMedia |
| Response:  200 – OK: Insert succeed  400 – “Image is over [MAX\_SIZE]bytes” if thumbnail is over size  400 – “cannot open the file” if wrong path or not valid image(jpeg/jpg)  MAX\_SIZE = 1000 bytes | | | | | |
|  |  |  |  |  |  |

*Table 2 - Language Service URI*

| **URI** | **Method** | **Description** | **Input** | **Output** | **GO Handler** |
| --- | --- | --- | --- | --- | --- |
| /languages | GET | Return a representation of all language in the system | n/a | Languages | goLanguages() |
| /languages | POST | Create a new language in the system  201(Created) | Language, require data in html body | Language | goInsertLanguage() |
| /languages | PUT | Update a specific language  200(OK)  204(no content)  404( not found) if id is not found or invalid | languageId, require data in html body | n/a | goUpdateLanguage() |
| /languages/{languageId} | GET | Get a specific language  200(OK), single language  404(not found) if id is invalid | languageId | Language | goGetLanguageById() |
| /languages/{languageId} | DELETE | Delete a specific language  200 (OK)  404(not found) if id is invalid or not found | languageId | n/a | goDeleteLanguage() |

# **JSON REPRESENTATIONS**

* **Media**



* **Medias**

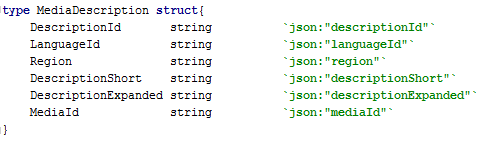
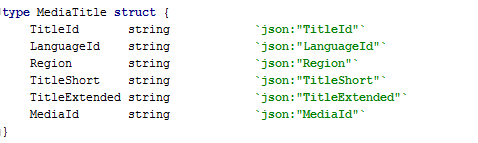
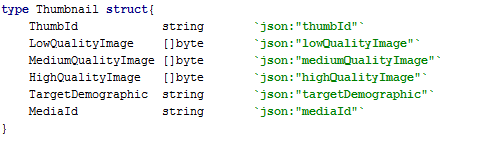
Machine generated alternative text:
/ ,'Azzey of media 
type Medias C) Media 

* **Language**

Machine generated alternative text:
type Language struct 
Json: "id" 
Json: "title" 
Title 
string 
string 

* **Languages**

C:\Users\NGHIEP~1\AppData\Local\Temp\msohtmlclip1\02\clip_image005.png

* **Description**
* **Title**
* **Thumbnail**
* **Thumbnai**