# Resource-Oriented Architecture: Hue Demonstration, Swagger, RAML, Apiary.io, Hydra-CG, Node.Js, Restlet, & Restlet Web Studio

# Phillip Thames

# ITIS 3310

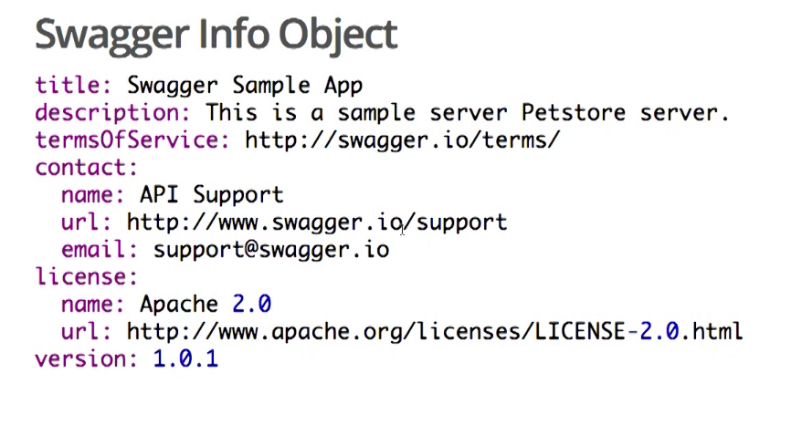
# 6/18/2017

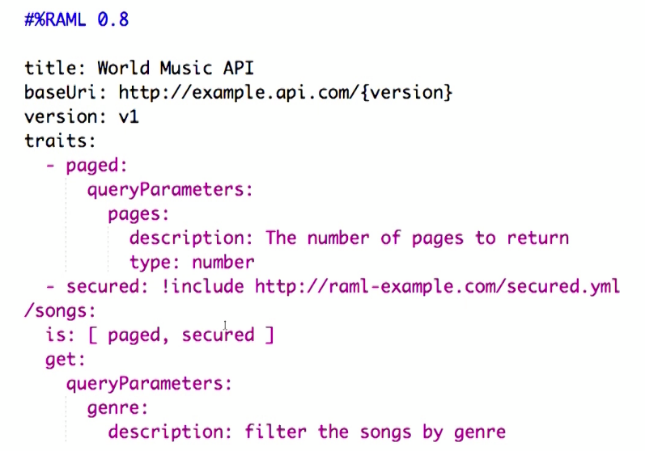
Introduction  
  
This paper is a brief overview describing some of the technologies used in Resource-Oriented Architecture with a focus on web development. I will discuss how these technologies such as Swagger, RAML, and Node.Js, assist in the development process. Being that these technologies can be intertwined amongst many language platforms, I will not go too in depth to each topic. However, I will discuss a main theme that each of these technologies possesses. When it comes to web development, there are a series of platforms that aid in the process of API design, server/client communication, and data representation that make built resources more transferable and ultimately extend their usable life.

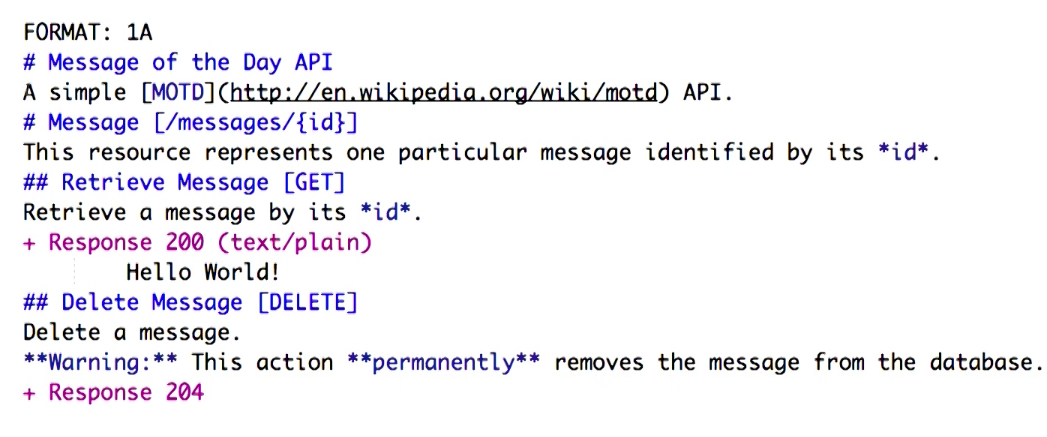
Hue Demonstration  
  
To set the general theme of the following technologies, Brian Sletten demonstrates the Hue lighting system from Phillips in the video Hue Demonstration. He uses this lighting system because it’s an easy technology to describe hypermedia and how its RESTful API allows the system to be used across many platforms.   
  
The Phillips hue lighting system has its own API with a network and a bridge. This is a level 2 system, meaning the user must have some type of development interaction to set up the system. Each bridge on the system controls 50 lights and the API is designed to give the user all sorts of functions to control the lights or get information about the lights. Using RESTful routes, the user can “GET lights” to find out which ones are on or use “GET light id” to return the attributes of a specific light. The RESTful routes also can return the attributes which are editable so that the user can customize the lighting system as well.

The Hue lighting system gives us a “language encoded representation of a hypermedia representation of an arbitrary collection of resources” (Sletten, 2016). What this means is that the API allows us to use logical identifiers to customize our requests from the system. The API can return many forms of requests from Json to HTML and we can even make the API browser friendly with editable tables to run the lights. Without going too in depth, the main theme of the Hue demonstration video, is to show you how a media designed with hypermedia and a RESTful API can extend the capabilities of your media. (Sletten, Hue Demonstration 2016)

## Swagger

Due to a series of constraints when it comes to user interfaces that communicate with the web and APIs, there is a strong interest in description languages. Some of these languages include, Haml, Json, and Yaml. Another on is Swagger. Swagger is an open source description language that supports Yaml and Json and has editing tools, code generators, and services. Swagger is highly rated and gives developers a series of path objects that a user has access to. Swagger also allows you to automate documentation. You can see Swagger in action at SwaggerPetStor.com. (Sletten, Swagger 2016)

RAML  
  
Another Web development description language is RAML. This is open sourced much like swagger and has the same type metadata as swagger. RAML also supports YAML/Json, has markdown documentation, and has reusable API patterns like traits, resource types and security schemes. (Sletten, RAML 2016)

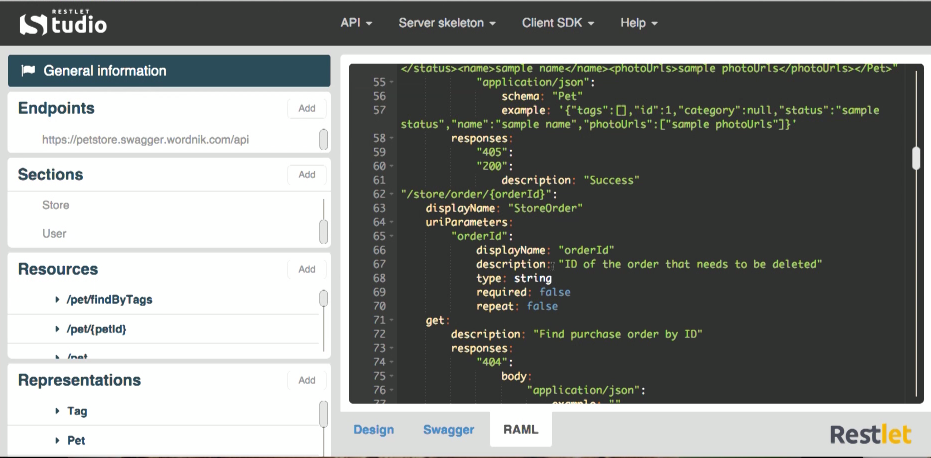
Apiary.io  
  
Apiary.io is a mainstream approach to description languages. It is a hosted service and is intended to be social and collaborative. It supports markdown documentation, mocked APIs, generated documentation and usage examples. Since it is a hosted service, you can sign in with any of your social media platforms. This also gives you the ability to chat with others using the same service to maximize your experience. Apiary’s library features notes on how to implement things using all languages and also documentation connected with their forum to help develop your API, create a mock server, and implement a live API. (Sletten, Apiary.io, 2016)  
  
Hydra CG

Hydra CG is Hydra plus Json-LD. It’s another description language which has a standard format and is hypermedia aware with arbitrary domains. This framework is similar to Apiary but it is designed to be extensible and interoperable w/RDF and linked data. (Sletten, Hyrda CG, 2016)  
  
Node JS

When implementing APIs, developers have their pick of tools to use. Node JS is an implementation tool that lets you use the same tools and libraries in your server and browser. It features an ease of installation and when using the express model, you can attach a database on the back end of your API. (Sletten, Node JS, 2016)  
  
Restlet API

Another implementation tool for your API is Restlet API. This framework is open source yet has a hosted service as well. Hosted service supports client authorizations and security. Restlet also has support to provide RESTful routes over many protocols and lets you test code as well. And when testing, requests that are made can be returned in many formats such as text, html, etc. (Sletten, Restlet API, 2016)

Restlet Web Studio

Restlet Web Studio is another Web development tool. It is open sourced and features resources and representations that can be mapped into object models. These models can be expressed as RAML, Swagger descriptions, or other API descriptions. Restlet Web Studio also gives you the ability to generate skeletons of servers and clients and has content negotiation capabilities. The infrastructure of this platform allows you to develop through all technologies which can give your resources a longer shelf life. (Sletten, Restlet Web Studio, 2016)  
  
Conclusion  
  
No matter how you go about the development process, you should keep in mind the life span of your product. Using the above-mentioned platforms can provide you with RESTful solutions when developing your software so your product is extendable and transferable over its lifetime.

# References

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243487.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243487.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243489.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243489.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243490.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243490.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243491.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243491.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243492.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243492.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243495.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243495.html)

[www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243496.html](http://www.safaribooksonline.com/library/view/learning-path-resource-oriented/9781491960981/video243496.html)

Questions

1. When implementing an API, developers can use which platform?
2. Node JS
3. Eclipse
4. Blue Jay
5. Http
6. What is the common theme of RESTful routes
7. They help you sleep
8. They can extend your resources lifetime
9. They allow your resource to be communicable through many technologies
10. Both B & C.
11. Phillips Hue lighting system has its own API? T/F

true