Project Proposal
RESTful Conversation Mining

Ilija Gjorgjiev
ilija.gjorgjiev@usi.ch

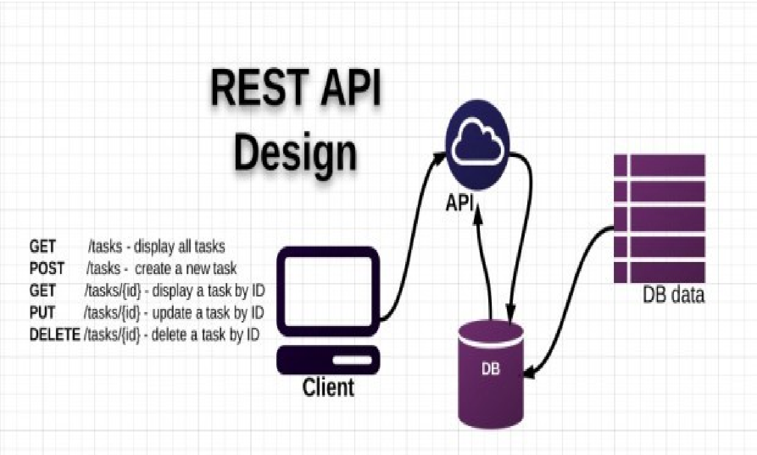
Tutor: Cesare Pautasso

March 7, 2018

Introduction & Motivation

Application programming interafaces(**APIs**) have enabled the possibility to exchange information between different systems over the network. A good API is responsible for the success of a Web service. A **RESTful** API is an application program interface that uses HTTP requests to get, create, mutate and delete data respectively. REST web services are based on the REpresentational state transfer, an architectural style, which provides interoperability between computer systems on the Internet. By using the stateless protocol and standard operations, REST systems aim for better performance, reliability and the ability to grow, by reusing components that can be managed and updated without affecting the system as a whole, even while it is running. For solving a specific problem by using the RESTful API, often multiple client-server interactions are required. Such interactions are comprised by sequences of HTTP requests/responds, which we describe them as **RESTful conversations**. The HTTP protocol offers simplicity and standardization which often makes these conversations to be limited to single interactions. However, when addressing non-functional requirements(e.g., security, reliability) these conversations can be comprised of several HTTP interactions. By analyzing such conversations(client-server interactions), we are able to grasp which patterns are more suitable than others. These patterns then provide useful information to the API designers, by setting up a common language to describe such conversations.

Figure 1: REST API


REST API
Design
GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID
Client
API
DB
DB data

Project Description

The main goal of this project is to mine conversation models from Web servers logs, thus tracking the interactions of multiple clients with a RESTful API. The result of these minings will then be visualized using the RESTalk visual language. The RESTalk visual language is an extension of the Business Process Model and Notation (BPMN).

Example 1

Input: Sample HTTP Log

```
20/7/2017 11:00:27 3.171.112.202 POST /job/42 202
20/7/2017 11:01:20 3.171.112.202 GET /job/42 200
20/7/2017 11:01:25 3.171.112.202 GET /job/42 303
20/7/2017 11:01:35 204.20.48.194 POST /job/42 202
20/7/2017 11:02:30 3.171.112.202 DELETE /job/42/output 200
```

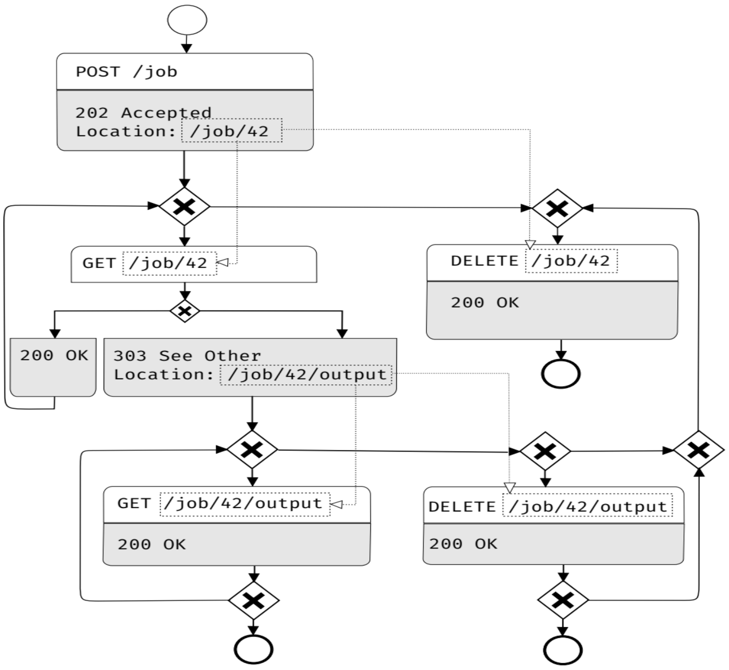
Explanation: Each HTTP log is comprised of the following data:

$$\overbrace{20/7/2017\ 11:00:27}^{\text{Date \& Exact Time}}\ \overbrace{3.171.112.202}^{\text{IP Address}}\ \overbrace{POST}^{\text{Method}}\ \overbrace{/job/42}^{\text{URI}}\ \overbrace{202}^{\text{Status}}$$

The above example can be represented as a long running operation with polling pattern.

Output: Visualizing by using RESTalk Pattern

Figure 2: Long Running Operation with Polling



Project division

The project can be divided in 7 components:

1. Given a file(HTTP log) read it and parse it to objects in memory.

2. Segmentation: Distribute the conversations of each client respectively, plus breaking down each clients conversations to different time windows.
3. Comparison of different conversations. Set up a difference measure algorithm.
4. New Pattern inference: Cluster similar conversations to produce candidate patterns.
5. Existing Pattern detection: Given a conversation pattern, find whether it is present in the HTTP log and how frequent it is.
6. Visualize conversations using the RESTalk language.
7. Visualize comparisons (how frequent are the various patterns/conversation classes).

Project Plan

The project will be divided in 6 milestones and it will be considered complete when all 6 milestones are done.

Milestone 1:

- Build the Parser. Read a HTTP log and parse it to objects in memory.
- Basic Conversation Visualization. Produce a basic visualization from the objects in memory using the RESTalk model.

Milestone 2:

- Segmentation.
- Advanced Conversation Visualization.

Milestone 3:

- Comparison: Define a difference measure algorithm, being able to compare diverse conversations which have the exact URI match.
- Visualize these comparisons.

Milestone 4:

- Detect if an existing pattern is present in the HTTP log.
- Visualize how frequent an existing pattern is.

Milestone 5:

- Given a sample of various conversations, group them into similar categories. In this comparisons we consider similar URIs too.
- Visualize the result by producing various candidate patterns.

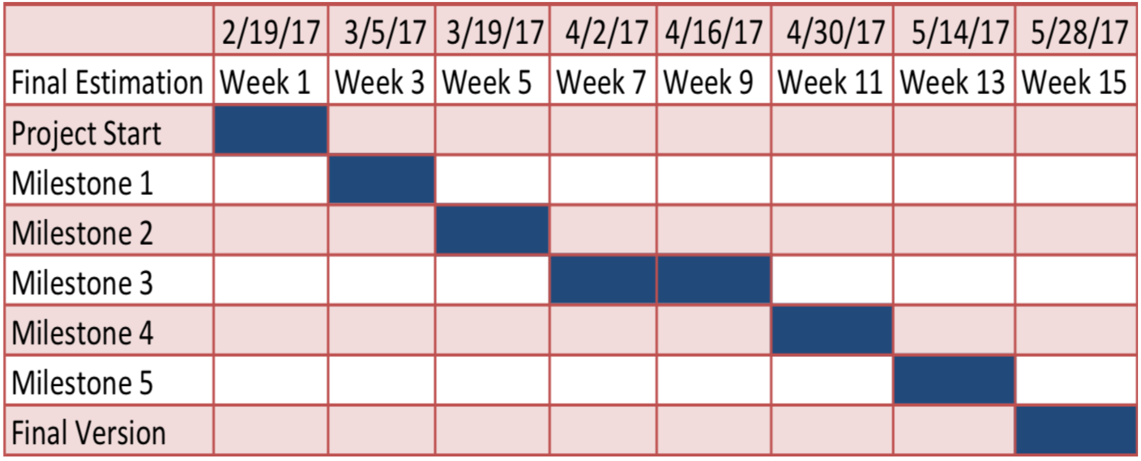
Final Milestone

- Testing with large data sets.
- Bug fixing.
- Project report.
- Project poster.

Extra Milestone:

- Online Web service.

Figure 3: Time Schedule for the Project



	2/19/17	3/5/17	3/19/17	4/2/17	4/16/17	4/30/17	5/14/17	5/28/17
Final Estimation	Week 1	Week 3	Week 5	Week 7	Week 9	Week 11	Week 13	Week 15
Project Start	■							
Milestone 1		■						
Milestone 2			■					
Milestone 3				■	■			
Milestone 4						■		
Milestone 5							■	
Final Version								■