ITEC833: Web Services

REPORT for Group Assignment

Chanikya Mandapathi

42944511

Date: 1/6/13

Declaration

I hereby declare that the work presented is my own. When other sources of information have been used, they have been acknowledged.

Introduction

I build a system that allows users to fetch content from external content providers like Guardian , Twitter , Weather forecast etc. Performing content search on news provided by Guardian,retrieving the top 15 tweets of the given user, getting Weather forecast information for a given city, getting stock quote value of a given company etc are some of the allowed operations on the system.Also the system provides event scheduler that allows users to create new events and store in a database.

I retrieve content from the following sites:

- News from The Guardian (http://www.guardian.co.uk)
- Tweets (http://www.twitter.com)

Apart from these contents, application also allows users to view weather forecast for a particular city and country. I use a SOAP-based service called 'GlobalWeather' as the source of weather data.I used a SOAP-based service called 'GetQuote' as a source of providing stock quote value of a given company.

Web Service Technologies

The System is python-based. The technologies i am using in the system are:

- SOAP
- REST
- WSDL
- WS-Security
- WS-Addressing
- XmlRpc
- Flask-JSONRPC

SOAP

SOAP is the messaging framework for communication between the web service and the client web application. I am using Soaplib as the SOAP engine.

The interaction with external weather forecast provider,StockQuote web service is done via SOAP. Also the client application communicates with developed web services using soap.

REST

Web service communicates with most external content providers via their RESTful APIs. I depend on the REST-based API of the following content providers:

- The Guardian
- Twitter

I've chosen these content providers because all of them have public APIs that can be used without any need for sign-up and/or authentication and the APIs are well- documented.

WSDL

The web service exposes its interface to the client web application via WSDL.

WS-Security

WS-Security is a specification to apply security to web services.It defines SOAP headers to carry security related data.WS-Security provides no of ways to authenticate a user.I used UsernameToken to authenticate user by passing the user provided credentials along with soap request and checking for match in the database of the user accounts.

WS-Addressing

WS-Addressing is a specification for providing transport neutral communication to the web services.I developed a service called eventScheduler(located in Server2.py) which parses the WS-Address headers added to the soap request to find the 'ReplyTo' address and calls the service at the ReplyTo address to get the list of events after creating a new event.

XML-RPC

XML-RPC is a protocol to make remote procedural calls by encoding the calls to xml.I developed a XMLRPC service called get_events(located in xmlrpcServer.py) and used SimpleXMLRPCServer to make XML-RPC Server engine.'eventScheduler' service(located in Server2.py) which

adds new events to the database will make XML-Remote procedural calls to get_events service to get the list of events.

Flask-JsonRPC

Flask-JsonRPC is a JSONRPC implementation for flask powered applications.I developed a Flask JSONRPC service called 'index'(located in fjson.py) which communicates with external service to give stock value of a given symbol.

Short Description of External Services

Twitter

Twitter has a REST-based public API. It offers a number of resources to fetch the content. It can also return results in multiple formats like Json ,xml etc.

I developed a service called twitterSearch which takes screen name of the user as the argument and returns latest 15 tweets of the given user.The developed service communicates with Twitter rest api and fetches the latest tweets of the user.

Please note that the user account provided should don't have any restrictions to view the timeline.

The Guardian

The Guardian provides a public rest api which allows content search on the news published by it.

I developed a service called news which takes a keyword and performs content search on the news of the Guardian .For this i used the rest api of the Guardian to fetch the content .

GlobalWeather

GlobalWeather is a public web service that offers GetWeather operation which gives the weather forecast of the given city.

I developed a service called WeatherService that takes name of the city and country of the city and gives the weather forecast of the given city.The developed service communicates with the GlobalWeather public web service and gets weather forecast information.

Stock Quote

Stock Quote is a public web service that offers GetQuote operation which gives stock value of a given symbol.

I developed a service which takes an input and gives the stock quote value of the given input.The developed service communicates with the StockQuote public web service and gets stock quote information.

Installation Guide

Assuming the python 2.7+,Suds,Soaplib,xml installed on the system , the following additional libraries have to be installed to run the project :

Requests

Mac OS : Using easy_install , $ easy_install requests

Using pip , $ pip install requests

Follow the installation guide of requests using the following url:

http://www.python-requests.org/en/latest/user/install/#install

Flask-JsonRPC

Mac OS :

Using easy_install , $ easy_install Flask-JSONRPC

Using pip , $ pip install Flask-JSONRPC

To install Flask-JsonRPC , Flask 0.9 + should be installed on the system

Follow the Flask-JSONRPC installation guide in the following url :

https://github.com/cenobites/flask-jsonrpc

XMLRPC

xmlrpclib - using easy_install , $ easy_install xmlrpclib

SimpleXMLRPCServer - using easy_install ,

$ easy_install SimpleXMLRPCServer

Documentation : http://wiki.python.org/moin/XmlRpc

SQLite :

Download the SQLite from the official site http://www.sqlite.org/download.html . The installation provides a command line application that will let you create and query databases.

Quick Start :

To run the project ,following steps should be followed :

1. Run the server.py , xmlrpcServer.py , fjson.py , Server2.py on different consoles
2. Run the client.py to interact with the available services
3. Go to the url : http://localhost:8080/ in your browser to visit the home page of the application
4. The user will be prompted for the login details. Enter the details and press login to access the available services . If the you want to create a new account click 'create a new account' hyperlink.

System Architecture

The system consists of three different components providing all the available services and a client application to access the web services .One of the component consists of weather service , news , tweetSearch , userAuth services.

weather : This service provides information about weather forecast of a given city by retrieving the information from external soap based service 'GlobalWeather' .

news : This service provides news items based on content search performed by the client. It depends on ‘Guardian’ external rest api for the news content.

tweetSearch : This service provides latest 10 tweets of the user given by the screen name .It depends on ‘Twitter’ rest api for the retrieving the tweets of the user.

userAuth : This service checks the authenticity of the login details provided by the user.It matches the provided user credentials with those entries in the database.

Second component consists of a flask-JSONRPC service ‘index’

index : This service provides the stock quote value of a given symbol.It fetches the information from external soap based service

‘StockQuote’.

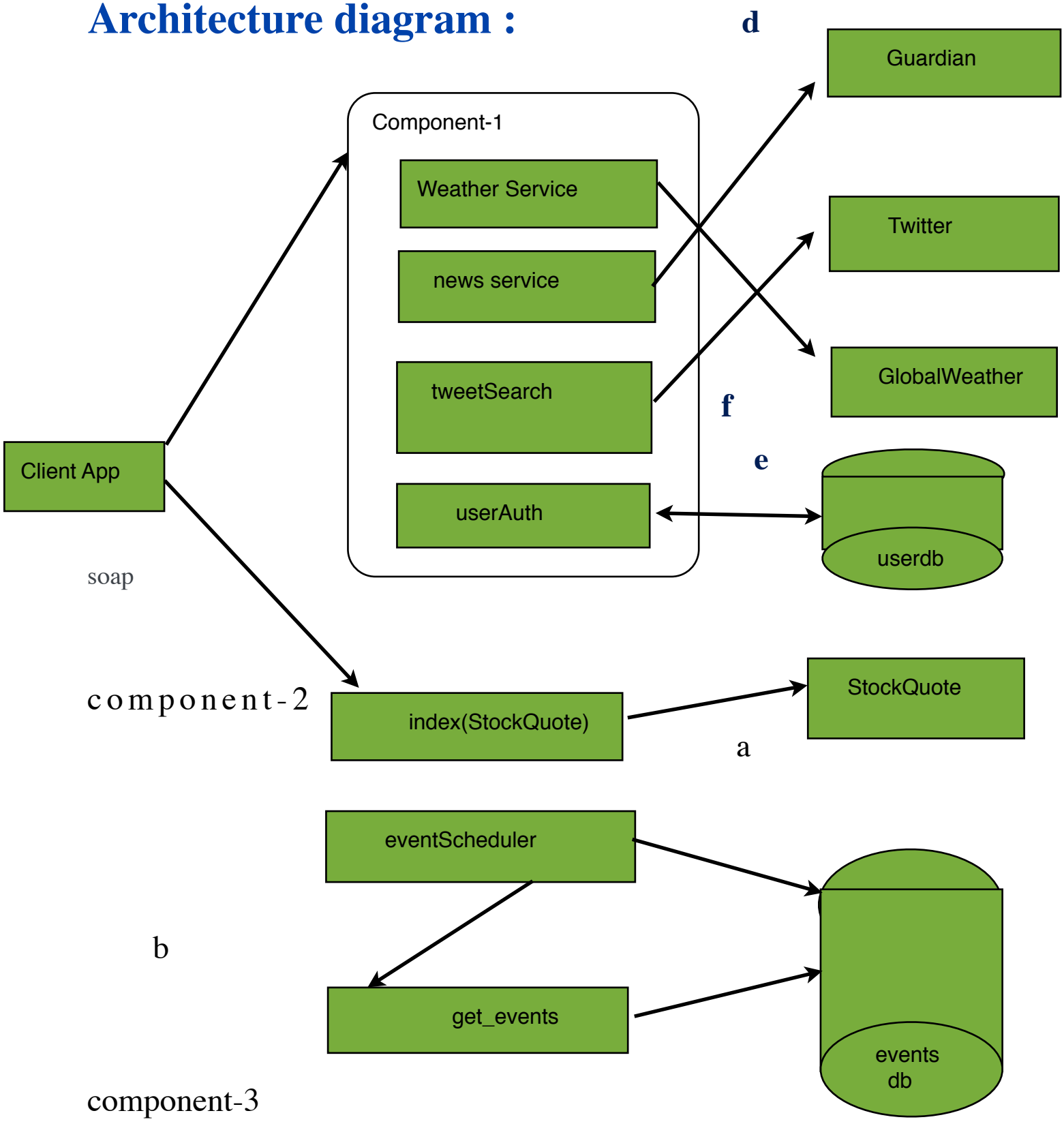
Third component consists of a soap based service ,eventScheduler

eventScheduler : This service creates new events by adding them into database.It performs transaction based inserts for inserting the values into database.It parses the request for WS-Address headers and uses the ‘ReplyTo’ header address to call the call back service get_events to get the list of added events.It makes XMLRPC calls to the call back for this.

Fourth component consists of a xmlrpc service , get_events

get_events : This service gives the list of added events by interacting with the database.

Architecture diagram :



a : Flask-JSONRPC

b : xmlrpc call

d, f : Rest

e : Soap

Last Minute Modifications :

I found some weird some problems with WSAddressing.Without changing anything , my code showing errors sometimes esp. with WSAddressing and executing without any errors other time.So i included some screen shots of output showing that code is executing without any errors.I included two versions of my project-one with WSAddressing and one without WSAddressing.

Group Members :

As i am doing the assignment individually, every task is performed by myself.

References

The Guardian, public API:

<http:// explorer.content.guardianapis.com/#/?format=json&order-by=newest>

Twitter Developers api,Twitter Inc

<https://dev.twitter.com/docs/api/1.1> WebServiceX.NET ,

GlobalWeather

<http://www.webservicex.net/WS/WSDetails.aspx?CATID=12&WSID=56>

Understanding WS-Security

http://msdn.microsoft.com/en-us/library/ms977327.aspx

Stock Quote

http://www.webservicex.net/WS/WSDetails.aspx?CATID=2&WSID=9

Python web services wiki

http://wiki.python.org/moin/WebServices

Suds

https://fedorahosted.org/suds/

Soaplib

http://soaplib.github.io/soaplib/2_0/

XMLRPC

http://wiki.python.org/moin/XmlRpc

Flask-JSONRPC

https://github.com/cenobites/flask-jsonrpc

Lecture materials of Dr. Zehua liu