**README**

**Overview:**

This document is intended to guide the user with the deployment of Digi-Diary Address Book application along with few snapshots of sample outputs.

**Software Specification:**

1. JDK 1.7
2. Jersey 1.17
3. JTA 1.1
4. Hibernate 4.0
5. MySQL 5.0.8
6. Apache Tomcat v6.0
7. JSON 1.3.2
8. JUnit4
9. ANT Scripts
10. Versioning - GitHub

**General Instructions:**

1. Code is versioned using GitHub and the source code is available at:

<https://github.com/sasmitanair/AddressBook/tree/Release1>

1. Download the project and execute build.xml to generate the class files.

Path: AddressBook\build.xml

1. Source Code contains the following:
   1. src – src folder contains the core business logic
   2. test – test packages contain JUnit test cases built on TDD approach
   3. jar – jar folder contains all dependant libraries.

**How To Run The Application:**

1. Start Apache Tomcat server and deploy the application
2. Download open source REST Client named Poster

<https://addons.mozilla.org/en-us/firefox/addon/poster/>

Follow the instructions on the website to download and install the plug-in on to the firebox browser.

Select Tools🡪Poster to launch the Client window.

1. Use this tool to send HTTP request to access REST APIs. Sample snapshots given below for reference.

**Sample Output:**

**Case 1: Get Address Book Contact List**

Retrieves the contact details from the database using GET HTTP request

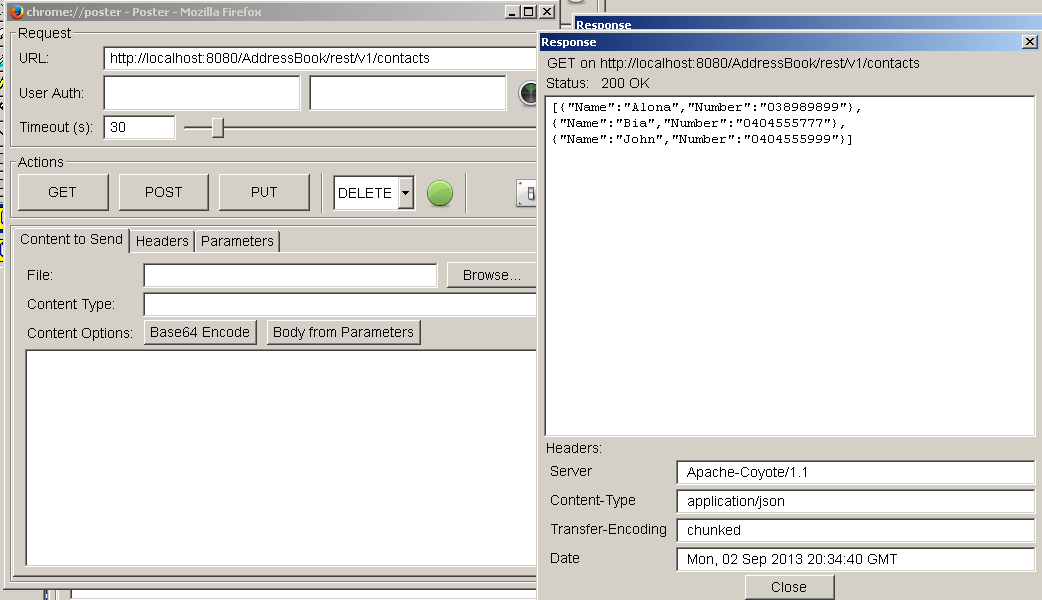
Request Type: GET

Request URL: <http://localhost:8080/AddressBook/rest/v1/contacts>

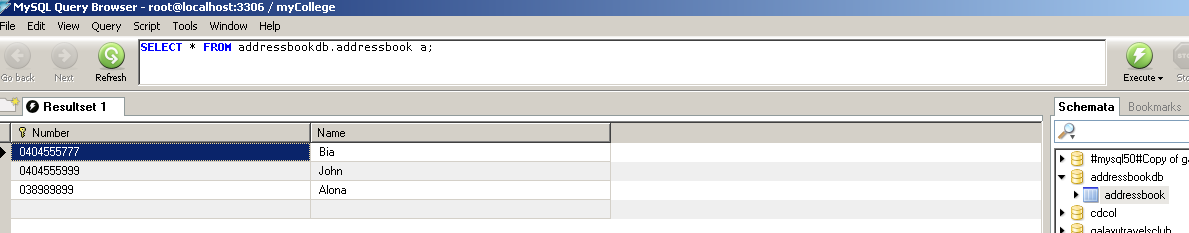
Response Type: JSON

Output Snapshot:

Contact details returned in alphabetical order

****

Database Snapshot:



**Case 2: Add New Contact Detail To The Existing Address Book**

Adds a new contact details entry to the AddressBook database table using PUT HTTP request

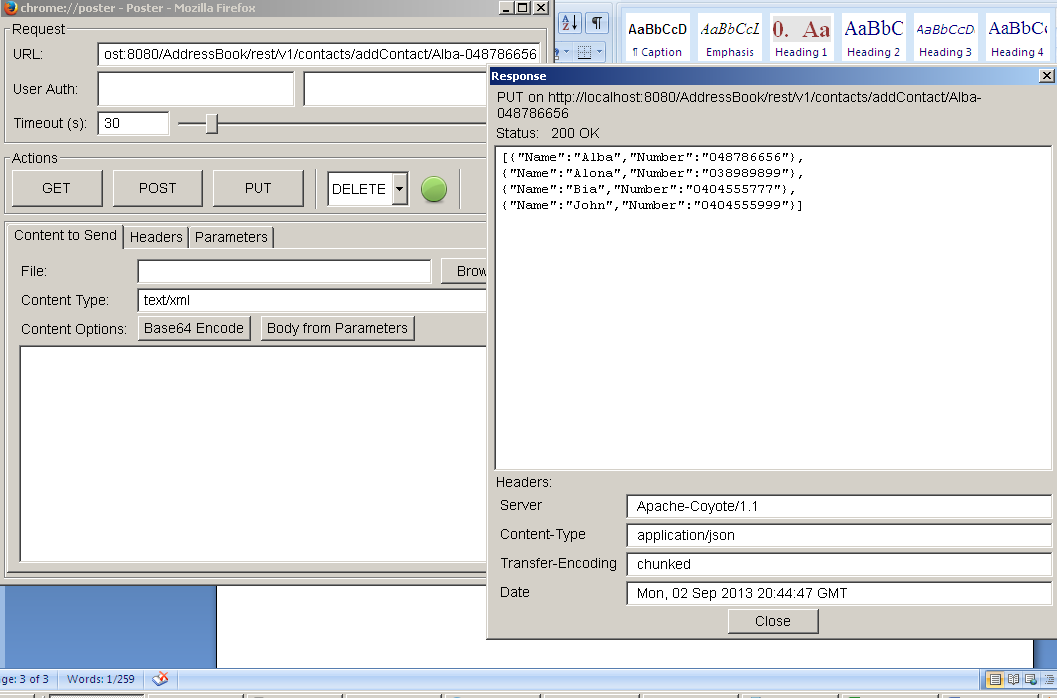
Request Type: PUT

Request URL: http://localhost:8080/AddressBook/rest/v1/contacts/addContact/Alba-048786656

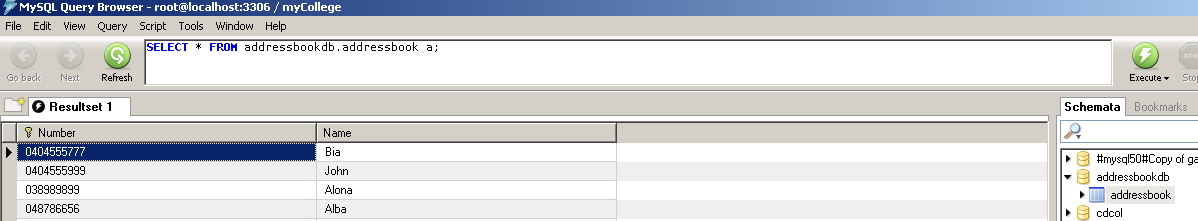
Response Type: JSON

**Output Snapshot:**

Newly added contact information added successfully and list is updated with the new entry placed alphabetically.



**Database Snapshot:**



**Case 3: Compare Two Address Books**

Provide a sample Address book input data via PUT HTTP request. Input address book is compared with the Address book in the database and the delta/unique list is returned as a JSON object

Request Type: PUT

Request URL: <http://localhost:8080/AddressBook/rest/v1/contacts/compare>

Response Type: JSON

Sample input JSON object:

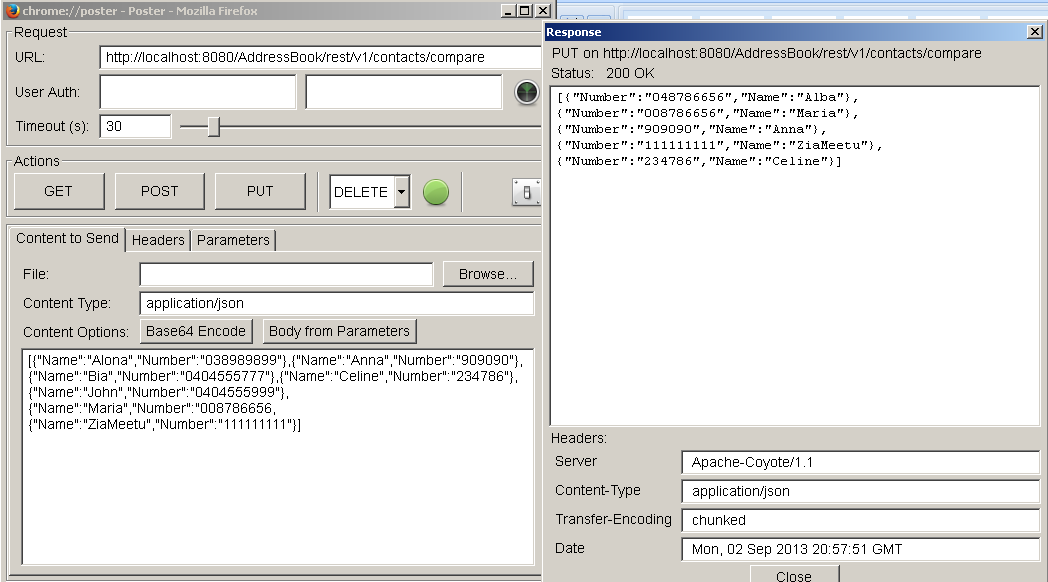
*[{"Name":"Alona","Number":"038989899"},{"Name":"Anna","Number":"909090"},{"Name":"Bia","Number":"0404555777"},{"Name":"Celine","Number":"234786"},{"Name":"John","Number":"0404555999"},{"Name":"Maria","Number":"008786656,{"Name":"ZiaMeetu","Number":"111111111"}]*

**Output Snapshot:**

Database Contact List – {Bias, John, Alona, Alba} (plz see previous db snapshot)

User input Contact List – {Alona, Anna, Bia, Celine, John, Maria, *ZiaMeetu*}

Expected Output List – {Alba, Anna, Celine, Maria, ZiaMeetu}



**Case 4: Update Existing Contact Detail**

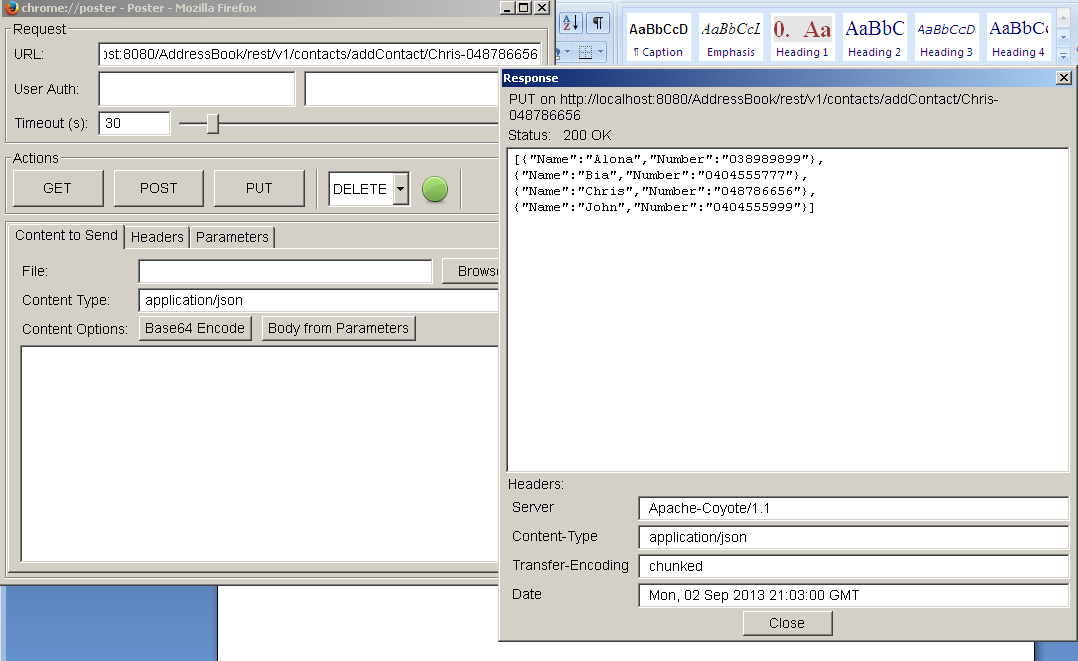
Let us update the newly inserted entry in Case 3 by changing name from Alba to Chris

Request Type: PUT

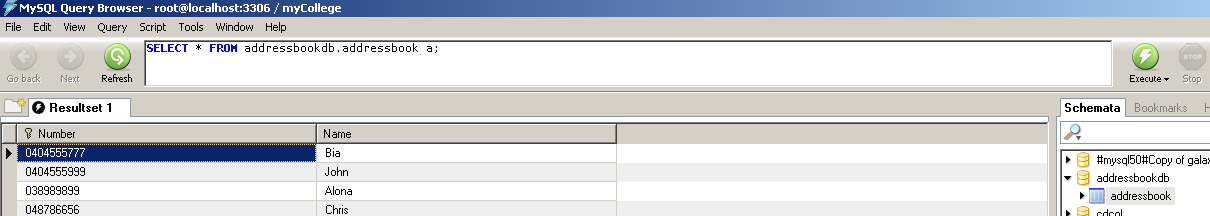
Request URL: http://localhost:8080/AddressBook/rest/v1/contacts/addContact/Chris-048786656

Response Type: JSON

**Output Snapshot:**



**Database Snapshot:**



**Case 5: Invalid Input Format For Adding/Updating Contact List**

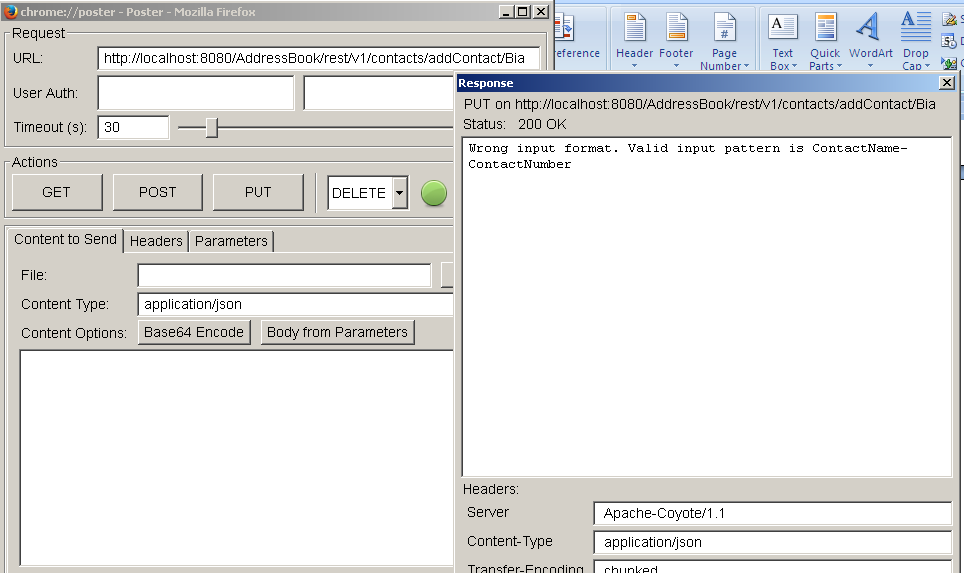
The HTTP PUT Request should carry the input detail in the format “ContactName-ContactNumber”, else a warning message is thrown.

Request Type: PUT

Request URL: http://localhost:8080/AddressBook/rest/v1/contacts/addContact/Bia

Response Type: JSON

**Output Snapshot:**



**Database Snapshot:**

Database table did not undergo any changes due to invalid input format

