E-Mailer Engine – E2

Minor Project

IBM Career Education

Disclaimer

This Software Requirements Specification document is a guideline. The document details all the high level requirements. The document also describes the broad scope of the project. While developing the solution if the developer has a valid point to add more details being within the scope specified then it can be accommodated after consultation with IBM designated Mentor.

INTRODUCTION

The purpose of this document is to define scope and requirements of an e-Mailer Engine – E2 for a leading web based retailer, who needs to send mass personalized mailers to its customers. Some typical examples are sending out birthday or anniversary greetings, and shipment notifications. A template driven engine that supports parameter substitution is envisaged for generating "personalized" mass mailers.

The proposed system, E2 will provide a simple and effective way to send such mailers from mailing list containing details like recipient's mail id, required parameters & their values, and the template id that will be used to create the mailer.

This document is the primary input to the development team to architect a solution for this project.

System Users

The administrator of the web-based retailer will primarily use the e-Mailer Engine (E2).

Assumptions

1. The mailing list will be generated from their back-end customer database and will be uploaded using a CSV file. This file will have 2 columns. The first column will contain the template id to be used; the second column will contain all the details in JSON, needed to process template to generate e-mails.
2. To simplify further, it is assumed that templates will be directly entered using the DB2 back-end directly in to the "template table". Such a table will have 2 columns viz. (a) template id, and (b) template text. The template id will be an auto generated field and will serve as the primary key.

REQUIREMENTS

E2 is a template driven e-mailer engine that leverages parameter substitution to generate "personalized" mass mailers to the customers.

Basic System Operation

The following steps outlines E2's basic operation:

1. E2 will accept a mailing list in a CSV format.
2. Once the mailing list is uploaded, E2 will generate a few sample mails for each row of data in the mailing list for review.
3. At this stage, the administrator will have an option to continue or cancel the operation.
4. If the administrator cancels the operation, the uploaded CSV is discarded.

5. If the administrator continues the operation, all the e-mails are generated and sent automatically.
6. All processed mailing list are stored in the system for future reference and audit purpose.

For the purpose of this project, you can simply generate a text file of all the e-mails to test the operation. However, students are encouraged to send out few mails using a Google Mail account.

Template Format

The E2's templates are in plain text format. Parameter substitution is supported through the use special embedded symbols in the form of [[param-name]] that will be replaced by the actual value of "param-name" when the template is evaluated to a mailing list. In other words, a parameter name is always enclosed inside '[[' and ']]' for easy detection by program through techniques like regular expressions. Here is a sample template to notify shipping of an order:

```
Dear [[salutation]] [[cust-last-name]]:

We are pleased to inform that your order number [[cust-order-id]] has
been shipped on [[ship-date]] by [[courier-service-name]] courier
service. Its airway bill number is [[order-awb-no]].

Sincerely,

John Ellison
Customer Servicing
```

In the above example, the parameters such as [[salutation]], [[cust-last-name]] etc. will be substituted with actual values obtained from the mailing list to generate the final e-mail to be sent to the customer.

JSON Structure

As outlined in the assumptions, each row in CSV will have a template id and the corresponding JSON text. This JSON will contain all the parameter name & values required to process the template for each recipient.

This CSV is generated from the back-end customer database every day for processing. The structure of this JSON is explained through the following example in context of the template sample outlined in the previous sub section:

```
{list: [
  {
    “mail-to”: “john@acme.com”,
    “salutation”: “Mr.”,
    “cust-last-name”: “Ellison”,
    “order-id”: “JE-004-2012”,
    “ship-date”: “March 12, 2012”,
    “courier-service-name”: “FedEx”,
    “order-awb-no”: “AC1239-22-06”
  },
  {
    “mail-to”: “tina@acme.com”,
    “salutation”: “Ms”,
    “cust-last-name”: “Simpson”,
    “order-id”: “TE-024-2012”,
    “ship-date”: “April 22, 2012”,
    “courier-service-name”: “Blue Dart”,
    “order-awb-no”: “MP1235-22-09”
  },
  {
    “mail-to”: “vijay@acme.com”,
    “salutation”: “Mr.”,
    “cust-last-name”: “Kalra”,
    “order-id”: “VK-001-2012”,
    “ship-date”: “January 09, 2012”,
    “courier-service-name”: “FedEx”,
    “order-awb-no”: “AC0027-22-06”
  }
]}
```

The JSON contains “mail-to” parameter whose value defines the mail id of the recipent along with all the additional parameter-value pairs required to process the template.

DEVELOPMENT ENVIRONMENT

E2-E-Mailer Engine will be developed as a web application using Java/JSP/JSON and Regular Expressions and DB2 database. Eclipse will be used as the IDE for the same.