Kavalan Management System Documentation
Release 0.4.7

Yu-Po (Luke) Chen

September 09, 2011

CONTENTS

Contents:

PROJECT OVERVIEW

1.1 Introduction

Kavalan Management System is an enterprise resource planning (ERP) application suite designed to facilitate the business processes at small- to medium-size firms. It is currently being developed with the support of Wu-Fu Laboratories, Ltd

1.2 Sub-systems

Several apps have been planned for inclusion in the suite. They are all at very early stage of development:

- *Doc Engine* for document search and retrieval
- *Dynamo* for creating database query dynamically based on user input.
- Stream (for posting status updates and sharing links)
- Forum (based on a modified version of LBForum)
- Account and Profile Management
- The *API* sub-system is used to centralize the management of public APIs exposed by the other sub-systems.

1.3 Dependencies

- Python >= 2.6
- Django 1.3
- Django-piston >= 0.2.2
- pyPDF >= 1.13 and ReportLab ToolKit >= 2.5
- guess-language >=0.2
- South >= 0.7.3
- Django-axes >= 1.2.4
- Django-notification >= 0.2.0
- LBForum >= 0.9.20
- Sphinx >= 1.0.7

• Database connectors for MySQL or PostgreSQL

1.4 Author(s)

Luke (Yu-Po) Chen, nuemail@gmail.com

1.5 LICENSE

QUICK START FOR USERS

Welcome to the Kavalan Management System. As a new user, you should find this guide helpful in guiding you through performing some of the most common takes you will be doing with this system.

2.1 Logging in

Click on Login () to log in.

2.2 Changing Interface Language

You can change the interface language by clicking on the Gear, selecting the language to use, and then clicking 'Go'.

2.3 Viewing your user profile

After you log in, you can see your profile and additional options by clicking on your user name.

2.4 Creating Users

If you are authorized to create new users, you will find the link to **Create New User** under **Manage Account**.

2.5 Searching Documents or Batch Records

Go to . On that page, you will the the tabs for or . Click on the one you want to use.

- For , enter the Serial Number () or the Title () of the document you want to search. You do not need to press *Enter*.

- For , enter the serial number or the product name of the record you want to search. In addition, you can specify the date range you are interested in. Just click on the From() or To () field and select the date you want to specify.

2.6 Adding/Editing/Deleting Documents or Batch Records

If you are authorized to make changes to the database, you will see Admin() on your navigation bar. Click on that, and you will be guided to the administration page.

You will find Documents() and/or Batch Records(). Click on the one you want to add/edit/delete, and then follow the on-screen instruction.

DOC ENGINE

3.1 Overview

Doc Engine is the system for document search and retrieval. It current handles the following data types

- **Document**: a database record for a digital file residing on the server.
- **BatchRecord**: a searchable record in the database that points to the physical location of a batch record

Doc Engine applies strict access control and records all requests for documents in the database.

3.2 Client-Server Communication

The search functionality of *Doc Engine* is exposed via its public API

`/api/documents` for Document search `/api/batchrecords` for Batch Record search

3.3 Document

3.3.1 Search API

GET /documents

Resource URL: /api/documents

Parameter:

- q(required): *JSON string of the search parameters (serial number/title, document level, and labels)*
- page_number(optional): *if there are more than one page of results (each page has 10 results), this specifies which page the server should return*

Response: A JSON string containing the records matching the query, or an empty []

Example Request:

GET /api/documents?query={"sn_title":"he","document_level":"1","labels":["2"]}&page_number=1

(Shown as the original strings. In reality, they must be URL-encoded before sending.)

```
{
"page_number": 1,
"num_pages": 1,
"data":
    [
        {
            "title": "Test 124",
            "labels": [
                {
                    "content": "TEST"
                }
            ],
            "file_url": "/doc_engine/access/4/",
            "version": "1.0",
            "location": "",
            "serial_number": "HE 124"
        },

        {
            "title": "Home, sweet home",
            "labels": [
                {
                    "content": "TEST"
                }
            ],
            "file_url": "/doc_engine/access/10/",
            "version": "1.0",
            "location": "",
            "serial_number": "HE 123121212"
        }
    ]
}
```

3.3.2 Model

There are two models for storing documents in Doc Engine: Document and FileObject. **Document** stores the metadata about each document, i.e. the serial number, title, and group permission. It also contains a version number that corresponds to a **FileObject**, which stores the file path to the actual file on the disk. The user can change the version number for a given Document instance and have it serve a specific version of the file.

3.3.3 Create/Update/Delete

Creating/updating/deleting of records is done via the Django admin interface

3.3.4 Access the actual file and security

The search result provides a link to the actual file at /doc_engine/access/*primary key*/ . Access control is employed to ensure that only authorized users are permitted to access the document.

Security is ensured through the following mechanisms:

1. **Group-based permission model:** Each Document has a set of permitted groups. When a user tries to access the document, his/her group membership is checked against that of the Document. Only a user who passes the test will be given access to the file. Others will see "Access Denied."

2. **Access recording:** Each time a user attempts to access a document, a record is written in the database, regardless of the outcome (success or access denial).

3. **Watermarking of PDF documents:** If the document to be accessed is a PDF file, an access watermark is added to the bottom of every page, specifying the user who downloaded the file and the time of access.

3.4 Batch Record

3.4.1 API

GET /batchrecords

Resource URL: /api/batchrecords

Parameters:

All the parameters are enclosed in a JSON query string

name *optional*	Product name for the batch record
batch_number *optional*	Batch number of the batch record
date_manufactured_from *optional*	Specify the date range to search for
date_manufactured_to *optional*	[date_manufactred_from, date_manufactured_to]

At least one parameter has to be non-empty, otherwise the server returns a 400 error.

Response: A JSON string containing the records matching the query, or an empty []

Example Request:

GET /api/batchrecords?query={"name":"am","batch_number":"AMP12","date_manufactured_from":"2011-04-01","date_manufactured_to":"2011-09-06"}&page_number=1

(Shown as the original strings. In reality, they must be URL-encoded before sending.)

```
{
    "page_number": 1,
    "num_pages": 1,
    "data": [
        {
            "date_manufactured": "2011-08-19",
            "batch_number": "AMP12",
            "name": "Ampicillin",
            "date_manufactured_minguo": "0100-08-19",
            "location": "AB 123"
        }
    ]
}
```

3.4.2 Model

3.4.3 Create/Update/Delete

Creating/updating/deleting of records is done via the Django admin interface.

DYNAMO

4.1 Overview

Dynamo is the sub-system that adds the functionality of dynamic filtering based on user input to any model. e.g. a dictionary-object:

```
{'app': 'doc_engine',
'model':'Document',
'filters':[{'field': 'title', 'lookuptype':'contains', 'value':'AF', 'op':'', 'exclude':False }]}
```

gets translated into:

```
Document.objects.filter(Q(title__contains='AF'))
```

Dynamo does all the necessary validation on the user input to ensure that the fields to which the filters are applied actually exist.

4.1.1 Core Functions

4.1.2 Tests

STREAM

5.1 Overview

Stream is an application in Kavalan that allows users to post short updates and share links with other people in the groups to which they belong.

5.2 Models

SECURITY

Given the importance of IT security in a corporate environment, the entire stack used to run Kavalan is configured/designed to prevent unauthorized access. The following sections deal with each level of the stack and discuss the security mechanisms implemented.

6.1 Server and Firewall

At Wu-Fu Laboratories, the Kavalan Managment System runs on CentOS 6.0 with default settings for SELinux (enforcing). Iptables is configured as follows:

```
:INPUT DROP [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1863:1287415]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
-A INPUT -p icmp -m icmp --icmp-type 8 -j REJECT --reject-with icmp-host-prohibited
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -s 192.168.2.0/24 -p tcp -m state --state NEW -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 21 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
```

Only computers in the local area network is permitted to access Kavalan.

6.2 Database

Kavalan currently runs on MySQL 5.1.52-1 at Wu-Fu Laboratories. It uses an isolated database; the database user that Django uses to access this database is grant permissions ONLY on this database; the root user's login credential is NOT available to the web server.

6.3 Web Server

Kavalan runs on a Nginx + Apache setup, with Nginx serveing as the reverse proxy and Apache binding to the localhost only. ALL traffic to/from the server is transmitted over HTTPS.

6.4 Kavalan Management System/Django

Kavalan requires the user to be authenticated before any action that accesses the database will be permitted. It uses the default Django authentication backend, which stores the user passwords in salted-hashes. As mentioned above, all traffic to/from the server is encrypted, preventing man-in-the-middle attack.

6.4.1 Document Access Security

For information on the security mechanisms specific to Doc Engine, please refer to *Access the actual file and security*.

THIRD-PARTY LICENSES

This file contains license details for 3rd party software which is distributed with Kavalan Management System.

---

1. License for jQuery & jQuery UI

   Copyright (c) 2007 John Resig, http://jquery.com/

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. License for Backbone.js

   Copyright (c) 2010 Jeremy Ashkenas, DocumentCloud

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

3. License for Underscore.js

   Copyright (c) 2011 Jeremy Ashkenas, DocumentCloud

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EX-PRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MER-CHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFT-WARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

4. Chosen, a Select Box Enhancer for jQuery and Protoype by Patrick Filler for Harvest.

   Available for use under the MIT License

   Copyright (c) 2011 by Harvest

   Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

   The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

   THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EX-PRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MER-CHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFT-WARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*