asphaltplant.online

Contents

**Introduction:**

The asphaltplant.online consists of the following components.

1. Web Application (Spring MVC)
2. MySQL (Relational Datastore)
3. Deployment (on AWS EC2 instance)

Following is the brief overview of the above mentioned points:

1. **Web App**

Web app is built in Spring MVC based on model view controller pattern. This webapp is a single monolithic application containing all of the business logic. It is a session based application which can be run of any java based webserver, i.e Apache Tomcat. All of the configuration is java annotation based.

Following are the core components with detailed description of the web app:

1. Java Server Pages (JSP) is used for view layer. The html, js and css is from an opensource frontend template <https://coreui.io/demo/#main.html>. The common components such as header, sidebar and footer are included in the WEB-INF/pages/fragments dir. These fragments are imported in the required pages. All of the JSPs are included in the WEB-INF/pages/ dir. Other than the page calls most of the actions such as form submits and updates are invoked from javascript using ajax calls. All of the javascript is included in /resources/js dir. The files names are self explanatory and can also be found in the bottom of the relevant pages. As for the css it is mostly provided by the coreui template, the custom css is written in /resources/css/style.css. JSP configuration and properties are defined in com.mka.configuration.AppConfig class. com.mka.configuration.core. SpringMvcInitializer class contains some basic configurations including the MultipartConfigElement for file uploading.
2. Hibernate ORM is used for interaction with the database. Hibernate configuration is specified in com.mka.configuration.AppConfig class.
3. Spring role based security with jdbcAuthentication is used for authentication. There are three type of roles supported. ADMIN, MANAGER and READONLY. Spring security configuration is specified in the com.mka.configuration.SecurityConfig class. The class also contains all the mappings of external paths and the caches properties.
4. All the requests are landed on the controller defined in the com.mka.controller package.
5. The controllers return the view models as per the route. All the business logic is on the service layer.
6. The service layer is responsible for applying all the business logic as per the request, it also interacts with the database
7. The database code is all under the com.mka.dao packages. All databases operations are performed in classes under this package.
8. **Database**

The datastore used for this project is the relation database MySQL. All the tables are named as per the convetions to allow easy regeneration of objects models from the database schemas.

The tables also have a brief description which specifies the purposes of table on a specific column.

1. **Deployment**

All the deployments are being done on AWS EC2 instance running both mysql and tomcat on a single vm instance. Database is accessed using basic authentication and is only accessible to localhost. Only port 80 and 443 is open to public. The nginx proxy server is running as a proxy server on top of the tomcat which is running on default port 8080. The ssl is the standard free ssl provided by LetsEncrypt, which is autorenewed every 90 days.