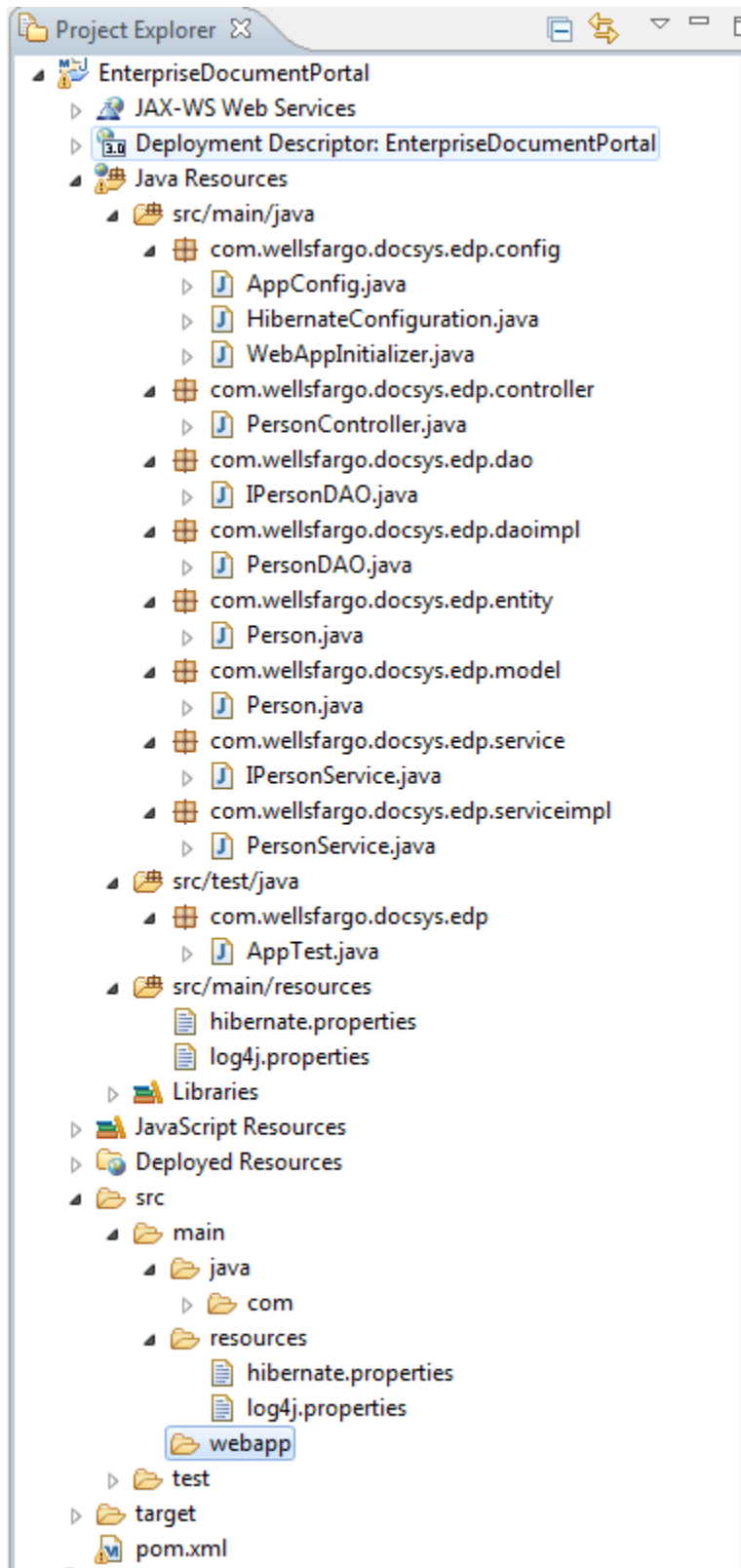


## 1. Structure



## 2. AppConfig

```
package com.wellsfargo.docsys.edp.config;

import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;

@Configuration
@ComponentScan("com.wellsfargo.docsys.edp")
@EnableWebMvc
@Import({HibernateConfiguration.class})
public class AppConfig {

}
```

## 3. WebApplicationInitializer

```
package com.wellsfargo.docsys.edp.config;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletRegistration.Dynamic;

import org.apache.log4j.Logger;
import org.springframework.web.WebApplicationInitializer;
import org.springframework.web.context.support.AnnotationConfigWebApplicationContext;
import org.springframework.web.servlet.DispatcherServlet;

public class WebAppInitializer implements WebApplicationInitializer {

    final Logger LOG = Logger.getLogger(WebAppInitializer.class);

    public void onStartup(ServletContext servletContext) throws ServletException {
        LOG.debug("onStartup::START");
        AnnotationConfigWebApplicationContext context = new AnnotationConfigWebApplicationContext();
        context.register(AppConfig.class);
        context.setServletContext(servletContext);
        Dynamic dynamic = servletContext.addServlet("dispatcher", new DispatcherServlet(context));
        dynamic.addMapping("/");
        dynamic.setLoadOnStartup(1);
        LOG.debug("onStartup::END");
    }
}
```

## 4. HibernateConfig

```

1 package com.wellsfargo.docsys.edp.config;
2
3 import java.util.Properties;
4
5 import javax.persistence.EntityManagerFactory;
6 import javax.sql.DataSource;
7
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.context.annotation.Bean;
10 import org.springframework.context.annotation.Configuration;
11 import org.springframework.context.annotation.PropertySource;
12 import org.springframework.core.env.Environment;
13 import org.springframework.dao.annotation.PersistenceExceptionTranslationPostProcessor;
14 import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
15 import org.springframework.jdbc.datasource.DriverManagerDataSource;
16 import org.springframework.orm.jpa.JpaTransactionManager;
17 import org.springframework.orm.jpa.JpaVendorAdapter;
18 import org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean;
19 import org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter;
20 import org.springframework.transaction.PlatformTransactionManager;
21 import org.springframework.transaction.annotation.EnableTransactionManagement;
22
23 @Configuration
24 @EnableTransactionManagement
25 @EnableJpaRepositories("com.wellsfargo.docsys.edp.dao")
26 @PropertySource(value = { "classpath:hibernate.properties" })
27 public class HibernateConfiguration {
28
29     @Autowired
30     private Environment environment;
31
32     @Bean
33     public LocalContainerEntityManagerFactoryBean entityManagerFactory() {
34         LocalContainerEntityManagerFactoryBean em = new LocalContainerEntityManagerFactoryBean();
35         em.setDataSource(dataSource());
36         em.setPackagesToScan(new String[] { "com.wellsfargo.docsys.edp.entity" });
37
38         JpaVendorAdapter vendorAdapter = new HibernateJpaVendorAdapter();
39         em.setJpaVendorAdapter(vendorAdapter);
40         em.setJpaProperties(additionalProperties());
41         return em;
42     }
43
44     @Bean
45     public DataSource dataSource() {
46         DriverManagerDataSource dataSource = new DriverManagerDataSource();
47         dataSource.setDriverClassName(environment.getRequiredProperty("jdbc.driverClassName"));
48         dataSource.setUrl(environment.getRequiredProperty("jdbc.url"));
49         dataSource.setUsername(environment.getRequiredProperty("jdbc.username"));
50         dataSource.setPassword(environment.getRequiredProperty("jdbc.password"));
51         return dataSource;
52     }
53
54     @Bean
55     public PlatformTransactionManager transactionManager(EntityManagerFactory emf) {

```

```

56     JpaTransactionManager transactionManager = new JpaTransactionManager();
57     transactionManager.setEntityManagerFactory(emf);
58     return transactionManager;
59 }
60
61 @Bean
62 public PersistenceExceptionTranslationPostProcessor exceptionTranslation() {
63     return new PersistenceExceptionTranslationPostProcessor();
64 }
65
66 Properties additionalProperties() {
67     Properties properties = new Properties();
68     properties.put("hibernate.dialect", environment.getRequiredProperty("hibernate.dialect"));
69     properties.put("hibernate.show_sql", environment.getRequiredProperty("hibernate.show_sql"));
70     properties.put("hibernate.format_sql", environment.getRequiredProperty("hibernate.format_sql"));
71     return properties;
72 }
73 }

```

## 5. Property Files - Hibernate & Log4j

```

1 jdbc.driverClassName = com.mysql.jdbc.Driver
2 jdbc.url = jdbc:mysql://localhost:3306/test_db
3 jdbc.username=root
4 jdbc.password=
5 hibernate.dialect = org.hibernate.dialect.MySQLDialect
6 hibernate.show_sql = true
7 hibernate.format_sql = true

```

```

1 # Direct log messages to a log file
2 log4j.appender.file=org.apache.log4j.RollingFileAppender
3 log4j.appender.file.File=EnterpriseDocumentPortal.log
4 log4j.appender.file.MaxFileSize=1MB
5 log4j.appender.file.MaxBackupIndex=1
6 log4j.appender.file.layout=org.apache.log4j.PatternLayout
7 log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
8
9 # Direct log messages to stdout
10 log4j.appender.stdout=org.apache.log4j.ConsoleAppender
11 log4j.appender.stdout.Target=System.out
12 log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
13 log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n
14
15 # Root logger option
16 log4j.rootLogger=INFO, file, stdout
17
18 # Log everything. Good for troubleshooting
19 log4j.logger.org.hibernate=INFO
20
21 # Log all JDBC parameters
22 log4j.logger.org.hibernate.type=ALL

```

## 6. POM

```

10 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema
11     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven
12     <modelVersion>4.0.0</modelVersion>
13
14     <groupId>com.wellsfargo.docsys</groupId>
15     <artifactId>EnterpriseDocumentPortal</artifactId>
16     <version>1.0.0</version>
17     <packaging>war</packaging>
18
19     <name>EnterpriseDocumentPortal</name>
20     <url>http://maven.apache.org</url>
21
22     <properties>
23         <!-- Spring -->
24         <org.springframework.version>4.1.4.RELEASE</org.springframework.version>
25         <org.springframework.security.version>3.2.4.RELEASE</org.springframework.security.v
26
27         <!-- persistence -->
28         <hibernate.version>4.3.6.Final</hibernate.version>
29         <mysql-connector-java.version>5.1.31</mysql-connector-java.version>
30         <spring-data-jpa.version>1.6.2.RELEASE</spring-data-jpa.version>
31
32         <!-- marshalling -->
33
34         <jackson.version>2.4.1</jackson.version>
35
36         <!-- logging -->
37         <org.slf4j.version>1.7.7</org.slf4j.version>
38         <logback.version>1.1.2</logback.version>
39
40         <!-- various -->
41         <hibernate-validator.version>5.1.2.Final</hibernate-validator.version>
42
43         <!-- util -->
44         <guava.version>17.0</guava.version>
45         <commons-lang3.version>3.3.2</commons-lang3.version>
46
47         <httpcore.version>4.3.2</httpcore.version>
48         <httpclient.version>4.3.5</httpclient.version>
49
50         <rest-assured.version>2.3.2</rest-assured.version>
51
52         <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
53     </properties>
54
55     <parent>
56         <groupId>org.springframework.boot</groupId>
57         <artifactId>spring-boot-starter-parent</artifactId>
58         <version>1.1.1.RELEASE</version>
59     </parent>

```

```

51
52 <dependencies>
53
54     <dependency>
55         <groupId>org.springframework.boot</groupId>
56         <artifactId>spring-boot-starter-web</artifactId>
57     </dependency>
58
59     <!-- Spring Security -->
60
61     <dependency>
62         <groupId>org.springframework.security</groupId>
63         <artifactId>spring-security-web</artifactId>
64         <version>${org.springframework.security.version}</version>
65     </dependency>
66     <dependency>
67         <groupId>org.springframework.security</groupId>
68         <artifactId>spring-security-config</artifactId>
69         <version>${org.springframework.security.version}</version>
70     </dependency>
71
72     <!-- Spring -->
73
74     <dependency>
75         <groupId>org.springframework</groupId>
76         <artifactId>spring-core</artifactId>
77         <version>${org.springframework.version}</version>
78         <exclusions>
79             <exclusion>
80                 <artifactId>commons-logging</artifactId>
81                 <groupId>commons-logging</groupId>
82             </exclusion>
83         </exclusions>
84     </dependency>
85     <dependency>
86         <groupId>org.springframework</groupId>
87         <artifactId>spring-context</artifactId>
88         <version>${org.springframework.version}</version>
89     </dependency>
90     <dependency>
91         <groupId>org.springframework</groupId>
92         <artifactId>spring-jdbc</artifactId>
93         <version>${org.springframework.version}</version>
94     </dependency>
95     <dependency>
96         <groupId>org.springframework</groupId>
97         <artifactId>spring-beans</artifactId>
98         <version>${org.springframework.version}</version>
99     </dependency>

```

```

100 <dependency>
101     <groupId>org.springframework</groupId>
102     <artifactId>spring-aop</artifactId>
103     <version>${org.springframework.version}</version>
104 </dependency>
105 <dependency>
106     <groupId>org.springframework</groupId>
107     <artifactId>spring-tx</artifactId>
108     <version>${org.springframework.version}</version>
109 </dependency>
110 <dependency>
111     <groupId>org.springframework</groupId>
112     <artifactId>spring-expression</artifactId>
113     <version>${org.springframework.version}</version>
114 </dependency>
115
116 <dependency>
117     <groupId>org.springframework</groupId>
118     <artifactId>spring-web</artifactId>
119     <version>${org.springframework.version}</version>
120 </dependency>
121 <dependency>
122     <groupId>org.springframework</groupId>
123     <artifactId>spring-webmvc</artifactId>
124     <version>${org.springframework.version}</version>
125 </dependency>
126
127 <!-- web -->
128
129 <dependency>
130     <groupId>org.apache.httpcomponents</groupId>
131     <artifactId>httpclient</artifactId>
132     <version>${httpclient.version}</version>
133     <exclusions>
134         <exclusion>
135             <artifactId>commons-logging</artifactId>
136             <groupId>commons-logging</groupId>
137         </exclusion>
138     </exclusions>
139 </dependency>
140 <dependency>
141     <groupId>org.apache.httpcomponents</groupId>
142     <artifactId>httpcore</artifactId>
143     <version>${httpcore.version}</version>
144 </dependency>
145
146 <!-- persistence -->
147
148 <dependency>
149     <groupId>org.springframework</groupId>
150     <artifactId>spring-orm</artifactId>
151     <version>${org.springframework.version}</version>
152 </dependency>

```

```

153<dependency>
154    <groupId>org.springframework.data</groupId>
155    <artifactId>spring-data-jpa</artifactId>
156    <version>${spring-data-jpa.version}</version>
157</dependency>
158<dependency>
159    <groupId>org.hibernate</groupId>
160    <artifactId>hibernate-entitymanager</artifactId>
161    <version>${hibernate.version}</version>
162</dependency>
163<dependency>
164    <groupId>org.javassist</groupId>
165    <artifactId>javassist</artifactId>
166    <version>3.18.1-GA</version>
167</dependency>
168<dependency>
169    <groupId>mysql</groupId>
170    <artifactId>mysql-connector-java</artifactId>
171    <version>${mysql-connector-java.version}</version>
172    <scope>runtime</scope>
173</dependency>
174
175<!-- web -->
176
177<dependency>
178    <groupId>javax.servlet</groupId>
179    <artifactId>javax.servlet-api</artifactId>
180    <version>3.0.1</version>
181    <scope>provided</scope>
182</dependency>
183
184<dependency>
185    <groupId>javax.servlet</groupId>
186    <artifactId>jstl</artifactId>
187    <version>1.2</version>
188    <scope>runtime</scope>
189</dependency>
190
191<!-- marshalling -->
192
193<dependency>
194    <groupId>com.fasterxml.jackson.core</groupId>
195    <artifactId>jackson-databind</artifactId>
196    <version>${jackson.version}</version>
197</dependency>
198

```



```

199<dependency>
200    <groupId>com.thoughtworks.xstream</groupId>
201    <artifactId>xstream</artifactId>
202    <version>1.4.7</version>
203</dependency>
204
205<!-- util -->
206
207<dependency>
208    <groupId>com.google.guava</groupId>
209    <artifactId>guava</artifactId>
210    <version>${guava.version}</version>
211</dependency>
212
213<!-- logging -->
214
215<dependency>
216    <groupId>org.slf4j</groupId>
217    <artifactId>slf4j-api</artifactId>
218    <version>${org.slf4j.version}</version>
219</dependency>
220<dependency>
221    <groupId>ch.qos.logback</groupId>
222    <artifactId>logback-classic</artifactId>
223    <version>${logback.version}</version>
224    <!-- <scope>runtime</scope> -->
225</dependency>
226<dependency>
227    <groupId>org.slf4j</groupId>
228    <artifactId>jcl-over-slf4j</artifactId>
229    <version>${org.slf4j.version}</version>
230    <!-- <scope>runtime</scope> --> <!-- some spring dependencies need to compile a
231</dependency>
232<dependency> <!-- needed to bridge to slf4j for projects that use the log4j APIs di
233    <groupId>org.slf4j</groupId>
234    <artifactId>log4j-over-slf4j</artifactId>
235    <version>${org.slf4j.version}</version>
236</dependency>
237<dependency>
238    <groupId>org.slf4j</groupId>
239    <artifactId>slf4j-log4j12</artifactId>
240    <version>${org.slf4j.version}</version>
241</dependency>
242
243<dependency>
244    <groupId>junit</groupId>
245    <artifactId>junit</artifactId>
246    <version>3.8.1</version>
247    <scope>test</scope>
248</dependency>
249</dependencies>

```

## 7. Controller

```

1 package com.wellsfargo.docsys.edp.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.transaction.annotation.Transactional;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7 import org.springframework.web.bind.annotation.RestController;
8
9 import com.wellsfargo.docsys.edp.model.Person;
10 import com.wellsfargo.docsys.edp.service.IPersonService;
11
12 @RestController
13 @RequestMapping("/data")
14 @Transactional
15 public class PersonController {
16     @Autowired
17     private IPersonService personService;
18
19     @RequestMapping("/person")
20     public Person getPersonDetail(@RequestParam(value = "id", required = false, defaultValue = "0") int id,
21     @RequestParam(value = "name", required = false, defaultValue = "0") String name,
22     @RequestParam(value = "location", required = false, defaultValue = "0") String location) {
23         Person person = new Person();
24         person.setId(id);
25         person.setName(name);
26         person.setLocation(location);
27         personService.getPersonDetail(person);
28         Person p = new Person();
29         p.setLocation("SUCCESS");
30         return p;
31     }
32 }

```

## 8. Service/BO

```

1 package com.wellsfargo.docsys.edp.serviceimpl;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Component;
5
6 import com.wellsfargo.docsys.edp.dao.IPersonDAO;
7 import com.wellsfargo.docsys.edp.model.Person;
8 import com.wellsfargo.docsys.edp.service.IPersonService;
9
10 @Component
11 public class PersonService implements IPersonService {
12
13     @Autowired
14     private IPersonDAO personDAO;
15
16     public void getPersonDetail(Person person) {
17         personDAO.createPerson(person);
18     }
19
20 }
21

```

## 9. Repo/DAO

```

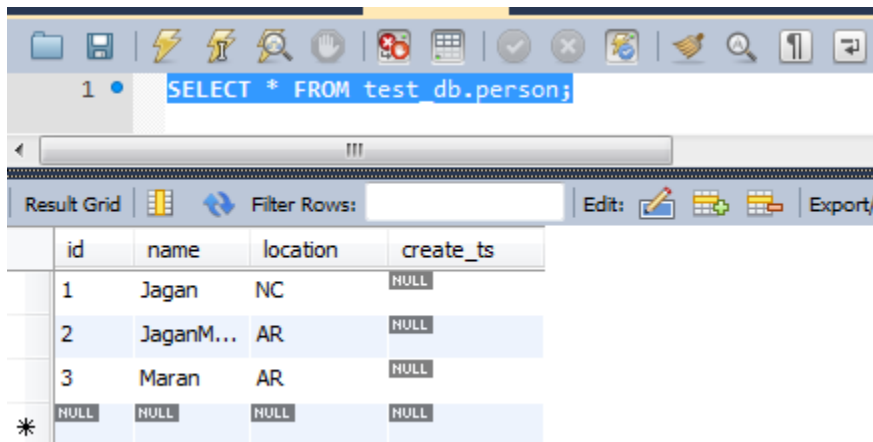
1 package com.wellsfargo.docsys.edp.daoimpl;
2
3 import javax.persistence.EntityManager;
4 import javax.persistence.PersistenceContext;
5
6 import org.apache.log4j.Logger;
7 import org.springframework.stereotype.Repository;
8
9 import com.wellsfargo.docsys.edp.dao.IPersonDAO;
10 import com.wellsfargo.docsys.edp.model.Person;
11
12 @Repository
13 public class PersonDAO implements IPersonDAO {
14
15     final Logger LOG = Logger.getLogger(PersonDAO.class);
16
17     @PersistenceContext
18     private EntityManager entityManager;
19
20     public void createPerson(Person person) {
21         LOG.debug("createPerson::START");
22         com.wellsfargo.docsys.edp.entity.Person personEntity = new com.wellsfargo.docsys.edp.entity.Person();
23         personEntity.setLocation(person.getLocation());
24         personEntity.setName(person.getName());
25         entityManager.persist(personEntity);
26         LOG.debug("createPerson::END");
27     }
28 }
29
30

```

## 10. Output

localhost:8080/EDP\_Test/data/person?id=15

{"id":15,"location":"Varanasi","name":"Ram"}



The screenshot shows a database management tool interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL query is entered in a text box: `SELECT * FROM test_db.person;`. Below the query, there is a 'Result Grid' section. The grid has four columns: 'id', 'name', 'location', and 'create\_ts'. The first three rows contain data: (1, Jagan, NC), (2, JaganM..., AR), and (3, Maran, AR). The fourth row is a summary row with all NULL values. The grid is currently showing 4 rows.

	id	name	location	create_ts
	1	Jagan	NC	NULL
	2	JaganM...	AR	NULL
	3	Maran	AR	NULL
*	NULL	NULL	NULL	NULL