**SpringJDBC Guide**

**Target of this workshop:**

* Create Spring Annotation Based Web Application
* Use Spring JDBC for Data access

First of all

* + install OJDBC driver into your maven 
  + Create database schema SPRINGJDBC and run the script 
* Create Maven Project SpringJDBC to create the project using annotation create

WebInitializer class :

**import** org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;

**public** **class** WebInitializer **extends** AbstractAnnotationConfigDispatcherServletInitializer {

@Override

**protected** Class<?>[] getRootConfigClasses() {

**return** **new** Class[] { WebMvcConfig.**class**};

}

@Override

**protected** Class<?>[] getServletConfigClasses() {

**return** **null**;

}

@Override

**protected** String[] getServletMappings() {

**return** **new** String[] { "/" };

}}

And create WebMvcConfig class

**import** java.util.logging.Logger;

**import** org.springframework.beans.factory.annotation.Value;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.ComponentScan;

**import** org.springframework.context.annotation.Configuration;

**import** org.springframework.context.annotation.PropertySource;

**import** org.springframework.jdbc.datasource.DriverManagerDataSource;

**import** org.springframework.web.servlet.ViewResolver;

**import** org.springframework.web.servlet.config.annotation.EnableWebMvc;

**import** org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdapter;

**import** org.springframework.web.servlet.view.InternalResourceViewResolver;

**import** org.springframework.web.servlet.view.JstlView;

@Configuration

@EnableWebMvc

@ComponentScan({"com.ih.spring.components"})

@PropertySource("/WEB-INF/resources/jdbcConfig.properties")

**public** **class** WebMvcConfig **extends** WebMvcConfigurerAdapter{

**final** **static** Logger ***logger***=Logger.*getLogger*(WebMvcConfig.**class**.getName());

@Value("${com.spring.jdbc.url}")

**private** String connectionUrl;

@Value("${com.spring.jdbc.driver.class}")

**private** String driverClass;

@Value("${com.spring.jdbc.username}")

**private** String userName;

@Value("${com.spring.jdbc.password}")

**private** String password;

@Bean

**public** ViewResolver viewResolver() {

InternalResourceViewResolver viewResolver = **new** InternalResourceViewResolver();

viewResolver.setViewClass(JstlView.**class**);

viewResolver.setPrefix("/WEB-INF/views/");

viewResolver.setSuffix(".jsp");

**return** viewResolver;

}

@Bean(name = "dataSource")

**public** DriverManagerDataSource dataSource() {

DriverManagerDataSource driverManagerDataSource = **new** DriverManagerDataSource();

driverManagerDataSource.setDriverClassName(driverClass);

driverManagerDataSource.setUrl(connectionUrl);

driverManagerDataSource.setUsername(userName);

driverManagerDataSource.setPassword(password);

**return** driverManagerDataSource;

}}

This replaces

* + All web.xml configurations
  + Datasource Bean definition
  + Components base package
* Now create User bean to map SECURITY\_USER table
* Create UserDao interface to contain the methods to be available in UserService class
* Create UserService class as follows :

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.Collection;

**import** java.util.List;

**import** javax.sql.DataSource;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.jdbc.core.JdbcTemplate;

**import** org.springframework.jdbc.core.RowMapper;

**import** org.springframework.stereotype.Service;

@Service

**public** **class** UserService **implements** UserDao,RowMapper<User> {

JdbcTemplate jdbcTemplate;

@Autowired

**public** UserService(DataSource datasource) {

jdbcTemplate=**new** JdbcTemplate(datasource);

}

@Override

**public** User findUser(String name) {

String sql="select name, password, enabled from security\_user where name=?";

User user = jdbcTemplate.queryForObject(sql,

**new** Object[]{name}, **this**);

**return** user;

}

@Override

**public** Collection<User> findUsers() {

String sql = "select name, password, enabled from security\_user";

List<User> users = jdbcTemplate.query(sql, **this**);

**return** users;

}

@Override

**public** User mapRow(ResultSet rs, **int** rowNum) **throws** SQLException {

User user=**new** User();

user.setName(rs.getString("name"));

user.setPassword(rs.getString("password"));

user.setEnabled(rs.getBoolean("enabled"));

**return** user;

}

}

Create JdbcHomeController to consume the service class