|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **□ 수행평가 - 빅데이터를 활용한 IoT 시스템 개발(feat.커넥티드카)** | | | | | | |
|  |  |  | |  | |  |
| **과정명** | | 빅데이터를 활용한 IoT 시스템 개발(feat.커넥티드카) | | | | |
| **교과목명** | | IoT 운영시스템 구축 응용기술 | | **훈련교사** | | 이진만 |
| **과정명** | | Web Client 기술 이해  Dynamic Web 기술 이해  Web Server 기술 이해 | | | | |
| **수행날짜** | | 2019.07.05 | 훈련생명 | | 이경헌 | |
| **과제개요** | | | | | | |
| 1. HTML5, CSS3.0, JavaScript를 이용하여 Dynamic Web Client를 구축 하시오 2. SpringMVC Framework를 이용하여 Web Server Application을 구축 하시오 3. Dynamic Web Client와 Web Server Application을 구축 하시오 4. 시스템 구조 5. 구현 화면 및 소스 코드 정리 | | | | | | |
| Dynamic Web Client & Web Server Apllication 구축 순서   1. Make Project 2. Spring Nature 3. Maven(Add Spring Library)    * pox.xml (List up Library)    * Download Library (automatic)    * maven -> update project 4. Add external Library 5. Biz,Dao 구현 6. 테스트를 위한 src\myspring.xml 7. Dispatcher 를 위한 web\web-inf\web.xml 8. 웹 기능을 위한 web\web-inf\spring.xml | | | | | | |
| 시스템구조  C:\Users\student\Desktop\spring.jpg | | | | | | |
| src\myspring.xml   |  |  |  | | --- | --- | --- | | 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="http://www.springframework.org/schema/beans"      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"      xmlns:aop="http://www.springframework.org/schema/aop"      xmlns:context="http://www.springframework.org/schema/context"      xmlns:p="http://www.springframework.org/schema/p"      xmlns:tx="http://www.springframework.org/schema/tx"      xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.2.xsd          http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.2.xsd          http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.2.xsd          http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.2.xsd">      <context:component-scan base-package="com.\*"/>      <tx:annotation-driven transaction-manager="txManager"/>      <!-- 1. Database Setting -->       <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">           <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>           <property name="url" value="jdbc:oracle:thin:@70.12.114.55:1521:xe"/>           <property name="username" value="db"/>           <property name="password" value="db"/>       </bean>       <!-- 2. Transaction Setting -->       <bean id="txManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">           <property name="dataSource" ref="dataSource"/>       </bean>       <!-- 3. MyBatis Setting -->       <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">           <property name="dataSource" ref="dataSource"/>           <property name="configLocation" value="classpath:com/config/mybatis.xml"/>       </bean>      <!-- 4. Spring Mybatis Connect -->       <bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">           <constructor-arg ref="sqlSessionFactory"/>       </bean>         <!-- 5. Mapper Setting -->       <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">           <property name="basePackage" value="com.mapper"/>       </bean>  </beans> | [cs](http://colorscripter.com/info#e) | | | | | | | |
| web\web-inf\web.xml  기능을 위한 xml 파일연결과 Dispatcher의 선언, 그리고 인코딩  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"* id=*"WebApp\_ID"* version=*"4.0"*>  <display-name>smvc4</display-name>  <welcome-file-list>  <welcome-file>index.html</welcome-file>  <welcome-file>index.htm</welcome-file>  <welcome-file>index.jsp</welcome-file>  <welcome-file>default.html</welcome-file>  <welcome-file>default.htm</welcome-file>  <welcome-file>default.jsp</welcome-file>  </welcome-file-list>  <servlet>  <servlet-name>action</servlet-name>  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  <init-param>  <param-name>contextConfigLocation</param-name>  <param-value>/WEB-INF/config/spring.xml</param-value>  </init-param>  </servlet>  <servlet-mapping>  <servlet-name>action</servlet-name>  <url-pattern>\*.mc</url-pattern>  </servlet-mapping>  <filter>  <filter-name>enc</filter-name>  <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  <init-param>  <param-name>encoding</param-name>  <param-value>UTF-8</param-value>  </init-param>  </filter>  <filter-mapping>  <filter-name>enc</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>  </web-app> | | | | | | |
| web\web-inf\spring.xml  데이터베이스 연결, Mybatis 설정을 해준다.  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <beans xmlns=*"http://www.springframework.org/schema/beans"*  xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*  xmlns:context=*"http://www.springframework.org/schema/context"*  xmlns:mvc=*"http://www.springframework.org/schema/mvc"*  xmlns:p=*"http://www.springframework.org/schema/p"*  xmlns:tx=*"http://www.springframework.org/schema/tx"*  xmlns:aop=*"http://www.springframework.org/schema/aop"*  xsi:schemaLocation=*"http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.2.xsd*  *http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd*  *http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.2.xsd*  *http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.2.xsd*  *http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.2.xsd"*>  <context:component-scan base-package=*"com.\*"* />  <mvc:annotation-driven />  <tx:annotation-driven transaction-manager=*"txManager"*/>    <bean id=*"viewResolver"*  class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>  <property name=*"prefix"* value=*"/view/"* />  <property name=*"suffix"* value=*".jsp"* />  <property name=*"order"* value=*"0"* />  </bean>  <!-- File Upload -->  <bean id=*"multipartResolver"*  class=*"org.springframework.web.multipart.commons.CommonsMultipartResolver"*>  <property name=*"maxUploadSize"* value=*"500000000"* />  </bean>  <!-- Multi language -->  <bean id=*"messageSource"* class=*"org.springframework.context.support.ResourceBundleMessageSource"*>  <property name=*"basenames"*>  <list>  <value>messages/messages</value>  <!-- 이 파일들에 의해서 -->  </list>  </property>  </bean>  <bean id=*"localeResolver"*  class=*"org.springframework.web.servlet.i18n.SessionLocaleResolver"*>  </bean>    <mvc:interceptors>  <!--사용자의 Browser 에서 접속할때 main Language 를 찾는다. -->  <bean  class=*"org.springframework.web.servlet.i18n.LocaleChangeInterceptor"*>  <property name=*"paramName"* value=*"lang"* />  </bean>  </mvc:interceptors>    <!-- 1. Database Setting -->  <bean id=*"dataSource"*  class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>  <property name=*"driverClassName"*  value=*"oracle.jdbc.driver.OracleDriver"* />  <property name=*"url"*  value=*"jdbc:oracle:thin:@70.12.114.55:1521:xe"* />  <property name=*"username"* value=*"db"* />  <property name=*"password"* value=*"db"* />  </bean>  <!-- 2. Transaction Setting -->  <bean id=*"txManager"*  class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <property name=*"dataSource"* ref=*"dataSource"* />  </bean>  <!-- 3. MyBatis Setting -->  <bean id=*"sqlSessionFactory"*  class=*"org.mybatis.spring.SqlSessionFactoryBean"*>  <property name=*"dataSource"* ref=*"dataSource"* />  <property name=*"configLocation"*  value=*"classpath:com/config/mybatis.xml"* />  </bean>  <!-- 4. Spring Mybatis Connect -->  <bean id=*"sqlSessionTemplate"*  class=*"org.mybatis.spring.SqlSessionTemplate"*>  <constructor-arg ref=*"sqlSessionFactory"* />  </bean>  <!-- 5. Mapper Setting -->  <bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>  <property name=*"basePackage"* value=*"com.mapper"* />  </bean>  </beans> | | | | | | |
|  | | | | | | |
| mybatis.xml  DB 연결을 위해 사용할 SQL 쿼리 문들을 \*mapper.xml 을 통해  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE configuration  PUBLIC "-//mybatis.org/DTD Config 3.0//EN"  "http://mybatis.org/dtd/mybatis-3-config.dtd">  <configuration>  <typeAliases>  <typeAlias type=*"com.vo.User"* alias=*"user"*/>  <typeAlias type=*"com.vo.Product"* alias=*"product"*/>  <!-- I'm gonna add this classes and i'll call these these -->  </typeAliases>    <mappers>  <mapper resource=*"com/config/usermapper.xml"*/>  <mapper resource=*"com/config/productmapper.xml"*/>  </mappers>  </configuration> | | | | | | |
| usermapper.xml  사용할 Mapper.java 를 연결하고, Method 를 id 와 매칭, 각 method가 수행할 query를 작성한다.  <mapper namespace=*"com.mapper.UserMapper"*>  <!-- This xml accessed by this mapper -->  <!-- -->  <insert id=*"insert"* parameterType=*"user"*>  INSERT INTO T\_USER VALUES  (#{id},#{pwd},#{name})  </insert>  <update id=*"update"* parameterType=*"user"*>  UPDATE T\_USER SET  PWD=#{pwd},NAME=#{name} WHERE ID=#{id}  </update>  <delete id=*"delete"* parameterType=*"String"*>  DELETE FROM T\_USER WHERE  ID=#{obj}  </delete>  <select id=*"select"* parameterType=*"String"* resultType=*"user"*>  SELECT \* FROM T\_USER WHERE ID=#{obj}  </select>  <select id=*"selectall"* resultType=*"user"*>  SELECT \*  FROM T\_USER  ORDER BY 1  </select>  </mapper> | | | | | | |
| UserMapper.java  Mybatis 와 연결되어 있는 interface 구현  **public** **interface** UserMapper {  **public** **void** insert(User obj);  **public** **void** delete(String obj);  **public** **void** update(User obj);  **public** User select(String obj);  **public** ArrayList<User> selectall();  } | | | | | | |
| UserDao.java  Mybatis 와 연결되어 있는 interface의 상속을 통해 DB와 값을 주고 받는다.  @Repository("udao")  **public** **class** UserDao **implements** Dao<String, User>{  @Autowired  UserMapper um;  @Override  **public** **void** insert(User v) **throws** Exception {  um.insert(v);  System.***out***.println(v + "UserOracleDao Inserted...");  }  @Override  **public** **void** update(User v) **throws** Exception {  um.update(v);  System.***out***.println(v + "UserOracleDao Updated...");  }  } | | | | | | |
| UserBiz.java  Mybatis와 연결되어있는 UserDao를 객체로 갖고, 객체의 method 호출을 통해 CRUD를 수행한다.  @Service("ubiz")  **public** **class** UserBiz **implements** Services<String, User>{  @Resource(name="udao")  Dao<String,User> dao;  **public** **void** setDao(Dao<String, User> dao) {  **this**.dao = dao;  }  @Override  **public** **void** insert(User v) **throws** Exception {  dao.insert(v);  }  } | | | | | | |
| DB에 있는 값들을 ajax를 통해, pchart.mc 에 접근하고, 값들을 JSON 형태로 받아HighCharts.com 을 이용하여 그래프로 표현한다.  **function** chart1(pdata) {  // Create the chart  Highcharts.chart('container',{  chart : {  type : 'column'},  title : {  text : 'Browser market shares. January, 2018'},  subtitle : {  text : 'Click the columns to view versions. Source: <a href="http://statcounter.com" target="\_blank">statcounter.com</a>'},  xAxis : {  type : 'category'  },  yAxis : {  title : {  text : 'Total percent market share'  }  },  legend : {  enabled : **false**  },  plotOptions : {  series : {  borderWidth : 0,  dataLabels : {  enabled : **true**,  format : '{point.y:.1f}%'}  }  },  tooltip : {  headerFormat : '<span style="font-size:11px">{series.name}</span><br>',  pointFormat : '<span style="color:{point.color}">{point.name}</span>: <b>{point.y:.2f}%</b> of total<br/>'},  series : [ {  name : "Browsers",  colorByPoint : **true**,  data : pdata  } ]  });  };  $(document).ready(**function**() {  $.ajax({  url : 'pchart.mc',  success : **function**(data) {  chart1(data);  }  })  });  </script>  <body>  <div id=*"container"*  style="min-width: *310px*; height: *400px*; margin: *0 auto*"></div>  </body> | | | | | | |
| DB에 접속하여 받아온 값들을 JSON 형태로 변환하여 출력한다.  @RequestMapping("/pchart.mc")  @ResponseBody  **public** **void** pdata(HttpServletResponse response) {  response.setCharacterEncoding("EUC-KR");  response.setContentType("text/json;charset=UTF-8");  ArrayList<Product> list = **new** ArrayList<Product>();  JSONArray ja = **new** JSONArray();  PrintWriter pt;  **try** {  list = pbiz.select();  **int** l = list.size();  **for**(**int** i=0;i<l;i++) {  JSONObject jo = **new** JSONObject();  jo.put("name",list.get(i).getName());  jo.put("y",list.get(i).getPrice());  ja.add(jo);  }  pt = response.getWriter();  pt.print(ja.toJSONString());  } **catch** (Exception e) {  e.printStackTrace();  }  } | | | | | | |
| Daum 지도 연결하여 현재위치 출력  **var** map;  $(document).ready(**function**() {  **var** container = document.getElementById('map');  **var** options = {  center : **new** daum.maps.LatLng(36.500701, 126.670667),  level : 12  };  map = **new** daum.maps.Map(container, options);  // 마커가 지도 위에 표시되도록 설정합니다  getData();  });  **if** (navigator.geolocation) {    // GeoLocation을 이용해서 접속 위치를 얻어옵니다  navigator.geolocation.getCurrentPosition(**function**(position) {    **var** lat = position.coords.latitude, // 위도  lon = position.coords.longitude; // 경도    **var** locPosition = **new** kakao.maps.LatLng(lat, lon), // 마커가 표시될 위치를 eolocation으로 얻어온 좌표로 생성합니다  message = '<div style="padding:5px;">여기에 계신가요?!</div>'; // 포윈도우에 표시될 내용입니다    // 마커와 인포윈도우를 표시합니다  displayMarker(locPosition, message);    });    } **else** { // HTML5의 GeoLocation을 사용할 수 없을때 마커 표시 위치와 인포윈도우 내용을 설정합니다    **var** locPosition = **new** kakao.maps.LatLng(33.450701, 126.570667),  message = 'geolocation을 사용할수 없어요..'    displayMarker(locPosition, message);  }  // 지도에 마커와 인포윈도우를 표시하는 함수입니다  **function** displayMarker(locPosition, message) {  // 마커를 생성합니다  **var** marker = **new** kakao.maps.Marker({  map: map,  position: locPosition  });    **var** iwContent = message, // 인포윈도우에 표시할 내용  iwRemoveable = **true**;  // 인포윈도우를 생성합니다  **var** infowindow = **new** kakao.maps.InfoWindow({  content : iwContent,  removable : iwRemoveable  });    // 인포윈도우를 마커위에 표시합니다  infowindow.open(map, marker);    // 지도 중심좌표를 접속위치로 변경합니다  map.setCenter(locPosition);  } | | | | | | |