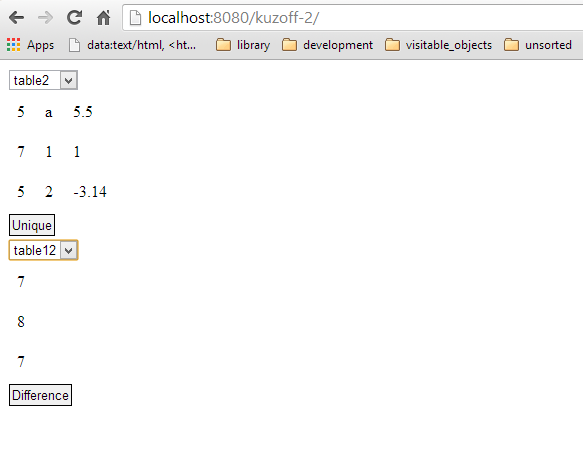
## Етап 15

Web-проект з використанням альтернативної технології.

*Виконання:* було обрано jQuery.

Основна сторінка має такий вигляд:



Веб-сторінка:

<!DOCTYPE html>

<html>

<head>

<title>kuzoff-2</title>

<script src="js/jquery-1.7.js" type="text/javascript"></script>

<script src="js/jquery.tmpl.js" type="text/javascript"></script>

<script src="js/main.js" type="text/javascript"></script>

<link rel="stylesheet" type="text/css" href="css/styles.css"/>

<script id="tablesTmpl" type="text/x-jquery-tmpl">

<option value="${name}">${name}</option>

</script>

<script id="rowTmpl" type="text/x-jquery-tmpl">

<div class="drow">

{{tmpl($data.values) "#cellTmpl"}}

</div>

</script>

<script id="cellTmpl" type="text/x-jquery-tmpl">

<div class="dcell">

${valueAsString}

</div>

</script>

</head>

<body>

<select id="selectTable1">

</select>

<div>

<div id="table1Data" class="dtable">

</div>

</div>

<input id="buttonUnique" type="submit" value="Unique" />

<br>

<select id="selectTable2">

</select>

<div>

<div id="table2Data" class="dtable">

</div>

</div>

<input id="buttonDifference" type="submit" value="Difference" />

</body>

</html>

JavaScript код:

$(document).ready(function () {

loadTableList('#selectTable1', '#table1Data');

loadTableList('#selectTable2', '#table2Data');

prepareUniqueButton();

prepareDifferenceButton();

});

function loadTableList(selectOneSelector, tableSelector) {

$.get("http://localhost:8080/kuzoff-ws/api/table", null,

function (responseData) {

$('#tablesTmpl').tmpl(responseData).appendTo(selectOneSelector);

loadDataForTable($(selectOneSelector).val(), tableSelector);

}

);

$(selectOneSelector).on('change', function () {

loadDataForTable($(this).val(), tableSelector);

});

}

function loadDataForTable(tableName, tableSelector) {

$.get("http://localhost:8080/kuzoff-ws/api/table/" + tableName + "/data", null,

function (responseData) {

$(tableSelector).empty();

$('#rowTmpl').tmpl(responseData).appendTo(tableSelector);

}

);

}

function prepareUniqueButton() {

$("#buttonUnique").click(function (e) {

$.post("http://localhost:8080/kuzoff-ws/api/table/" + $('#selectTable1').val() + "/unique", null,

function (responseData) {

$('#table1Data').empty();

$('#rowTmpl').tmpl(responseData).appendTo('#table1Data');

}

);

e.preventDefault();

});

}

function prepareDifferenceButton() {

$("#buttonDifference").click(function (e) {

$.post("http://localhost:8080/kuzoff-ws/api/table/" + $('#selectTable1').val() + "/diff/" + $('#selectTable2').val(), null,

function (responseData) {

$('#table1Data').empty();

$('#rowTmpl').tmpl(responseData).appendTo('#table1Data');

}

);

e.preventDefault();

});

}

## Етап 6

Другий варіант розподілених версій систем.

*Виконання:* був обраний та реалізований варіант Remote Http Method Invocation, що використовує фреймворк Spring Remoting.

Приклад налаштування для клієнтської сторони:

package cyberwaste.kuzoff.http.client;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import org.springframework.beans.factory.config.BeanDefinition;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.Scope;

import org.springframework.remoting.httpinvoker.HttpInvokerProxyFactoryBean;

import cyberwaste.kuzoff.core.DatabaseManager;

@Configuration

@ComponentScan("cyberwaste.kuzoff")

class Config {

@Bean

public HttpClientManager httpClientManager() {

HttpClientManager httpClientManager = new HttpClientManager();

httpClientManager.setDatabaseManager(remoteDatabaseManager());

httpClientManager.setReader(new BufferedReader(new InputStreamReader(System.in)));

return httpClientManager;

}

@Bean

@Scope(BeanDefinition.SCOPE\_PROTOTYPE)

public DatabaseManager remoteDatabaseManager() {

HttpInvokerProxyFactoryBean httpInvokerProxyFactoryBean = new HttpInvokerProxyFactoryBean();

httpInvokerProxyFactoryBean.setServiceUrl("http://localhost:8080/kuzoff-http/kuzoff.service");

httpInvokerProxyFactoryBean.setServiceInterface(DatabaseManager.class);

httpInvokerProxyFactoryBean.afterPropertiesSet();

return (DatabaseManager) httpInvokerProxyFactoryBean.getObject();

}

}

Приклад налаштування для серверної сторони:

Налаштування веб-контейнера:

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"

version="3.0">

<!-- Spring configuration -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>

classpath:httpInvokerServerContext.xml

</param-value>

</context-param>

<servlet>

<servlet-name>httpInvokerServer</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>httpInvokerServer</servlet-name>

<url-pattern>\*.service</url-pattern>

</servlet-mapping>

</web-app>

Налаштування сервлета:

<?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:mvc="http://www.springframework.org/schema/mvc"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package="cyberwaste.kuzoff.http.server" />

<mvc:annotation-driven />

<bean class="org.springframework.web.servlet.handler.SimpleUrlHandlerMapping">

<property name="mappings">

<value>

/kuzoff.service=httpInvokeDatabaseManager

</value>

</property>

</bean>

</beans>

Налаштування Remote Http Method Invocation:

<?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:p="http://www.springframework.org/schema/p"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:annotation-config />

<bean

id="httpInvokeDatabaseManager"

class= "org.springframework.remoting.httpinvoker.HttpInvokerServiceExporter"

p:service-ref="databaseManager"

p:serviceInterface="cyberwaste.kuzoff.core.DatabaseManager" />

<bean name="databaseManager" class="cyberwaste.kuzoff.core.impl.DatabaseManagerImpl">

<property name="fileSystemManager" ref="fileSystemManager" />

</bean>

<bean name="fileSystemManager" class="cyberwaste.kuzoff.core.impl.FileSystemManagerImpl" />

</beans>