МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ



**Звіт**

До лабораторної роботи №7-8

З дисципліни:

«Веб-технології та веб-дизайн»

На тему:

**«Розробка програм асинхронного обміну даними у Web-середовищі засобами бібліотеки JQuery. Ajax.**»

Варіант №25

*Виконав:*

Студент групи КН-37

Шандра О.С.

*Прийняв:*

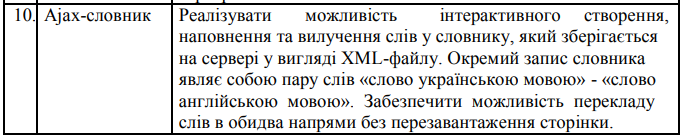
доцент каф. ІСМ

Василюк А. С.

**Львів 2018**

**Мета роботи:** Метою лабораторної роботи є вивчення основ асинхронного обміну даними мовою JavaScript за допомогою технології Ajax та бібліотеки JQuery при реалізації практичних задач Web-програмування.

**Завдання (Варіант 25-15 = 10):**



**Теоретичні відомості:**

Активний розвиток застосування різноманітних Web-додатків, зокрема, соціальних мереж та засобів електронної комерції в Internet, зумовив необхідність створення інтерактивних інтерфейсів користувача, які за функціональними можливостями наближуються до традиційних віконних додатків. Місце програмного інструментарію для вирішення цієї задачі за останні 5-7 років зайняли різноманітні бібліотеки на основі вбудованої в Webбраузер мови Javascript. Серед таких засобів найбільш поширеною та вдалою виявилась бібліотека JQuery. Вона поєднує в собі лаконічних синтаксис, надзвичайно розвинуті засоби пошуку елементів Web-сторінки на основі CSS та XML-орієнтованої спеціалізованої мови XPath, функції маніпуляції XHTML DOM, анімаційні ефекти, а також ефективний інструментарій асинхронного обміну інформацією з Web-сервером Ajax.

Оскільки бібліотека JQuery написана мовою Javascript, то застосовувати код можна всюди, де допустимий код Javascript: в оброблювачах подій, викликаючи функції тощо.

Спрощено запис команди JQuery виглядає наступним чином: $(selector).action1(args1).action2(args2) … або $.action(args) де знак $ означає скорочену назву об‘єкта JQuery. Цей знак можна замінити на виклик: JQuery(selector);

selector визначає один або групу елементів Web-документа, об‘єднаних за певним критерієм (назвою тега, ідентифікатору, CSS-класу тощо);

action, action1, action2 – визначають функцію, що треба застосувати для відібраних селектором елементів (встановити стиль CSS, заповнити/стерти вміст елемента, додати/вилучити оброблювач деякої події тощо);

arg, arg1, arg2 – необов‘язкові аргументи функцій, що застосовуються до елементів (новий вміст елемента, його атрибути тощо).

Ajax (асинхронний Javascript) – це підхід до створення інтерактивних Web-додатків, згідно з яким обмін даними між Web-браузером та Web-сервером відбувається у фоновому режимі та надає можливості оновлювати лише частину Web-сторінки.

Такий підхід зменшує мережний трафік та дозволяє будувати додатки, наближені до «віконних», які є більш звичними для користувача. Головним чином підхід базується на використанні засобів мови Javascript. При цьому з серверного боку жодних змін виконувати не потрібно у порівнянні з традиційним підходом.

На клієнті за Ajax відповідає об‘єкт Javascript XMLHttpRequest. Передача даних за допомогою цього об‘єкта може відбуватись як у синхронному, так і у асинхронному режимі. Складність безпосереднього використання цього об‘єкта зумовлена неповною сумісністю реалізації в різних браузерах.

Тому, як правило, використовують кросплатформні бібліотеки, які уніфікують доступ до даного об‘єкта. Однією з таких бібліотек є JQuery.

**Хід виконання завдання:**

Сервер було розроблено на мові програмування node.js.

**App.js**:

var express = require("express");

var bodyParser = require("body-parser");

var fs = require("fs");

var xml2js = require('xml2js'); //https://www.npmjs.com/package/xml2js

var app = express();

var jsonParser = bodyParser.json();

var FILE\_NAME = "dictionary.xml";

//START APP COMMAND >>>node app.js

app.use(express.static(\_\_dirname + "/public"));

app.get("/api/populate\_test\_data", function(req, res){

var testData = {

dictionary: {

"писати": "write",

"мама": "mother",

"тато": "father",

"історія": "history",

"мишка": "mouse",

"нога": "leg",

"машина": "car"

}

};

var builder = new xml2js.Builder();

var xml = builder.buildObject(testData);

fs.writeFile(FILE\_NAME, xml, () => {

res.status(200).send();

});

});

var getDictionary = () => {

return new Promise((resolve, reject) => {

fs.readFile(\_\_dirname + "\\" + FILE\_NAME, (err, data) => {

if(err){

reject(err);

}

var parser = new xml2js.Parser();

parser.parseString(data, (err, result) => {

if(err) {

reject(err);

}else{

resolve(result.dictionary);

}

});

});

});

};

var saveDictionary = dictionary => {

return new Promise((resolve, reject) => {

var builder = new xml2js.Builder();

var xml = builder.buildObject({ dictionary });

fs.writeFile(\_\_dirname + "\\" + FILE\_NAME, xml, err => {

if(err){

reject();

}

else{

resolve();

}

});

});

};

var saveWord = (ua\_word, en\_word) => {

return getDictionary().then(dictionary => {

dictionary[ua\_word] = en\_word;

return saveDictionary(dictionary);

});

}

// Додати переклад українського слова

app.post("/api/dictionary/ua\_word/:word", jsonParser, function(req, res){

var ua\_word = req.params.word;

var en\_word = req.body.en\_word;

saveWord(ua\_word, en\_word).then(() => {

res.status(200).send();

});

});

// Додати переклад англійського слова

app.post("/api/dictionary/en\_word/:word", jsonParser, function(req, res){

var en\_word = req.params.word;

var ua\_word = req.body.ua\_word;

saveWord(ua\_word, en\_word).then(() => {

res.status(200).send();

});

});

// Дістати переклад українського слова

app.get("/api/dictionary/ua\_word/translate/:word", function(req, res){

var ua\_word = req.params.word;

getDictionary().then(dictionary => {

if(!dictionary[ua\_word]){

res.status(404).send();

} else{

res.status(200).send(dictionary[ua\_word][0]);

}

});

});

// Дістати переклад англійського слова

app.get("/api/dictionary/en\_word/translate/:word", function(req, res){

var en\_word = req.params.word;

var searched = "";

getDictionary().then(dictionary => {

for (ua\_word in dictionary) {

if(dictionary[ua\_word] == en\_word){

searched = ua\_word;

break;

}

}

if(searched == ""){

res.status(404).send();

} else{

res.status(200).send(searched);

}

});

});

// Видалити переклад українського слова

app.delete("/api/dictionary/ua\_word/:word", function(req, res){

var ua\_word = req.params.word;

getDictionary().then(dictionary => {

if(!dictionary[ua\_word]){

res.status(404).send();

} else{

delete dictionary[ua\_word];

saveDictionary(dictionary).then(() => {

res.status(200).send();

});

}

});

});

// Видалити переклад англійського слова

app.delete("/api/dictionary/en\_word/:word", function(req, res){

var en\_word = req.params.word;

getDictionary().then(dictionary => {

for (ua\_word in dictionary) {

if(dictionary[ua\_word] == en\_word){

delete dictionary[ua\_word];

return saveDictionary(dictionary);

}

}

}).then(() => {

res.status(200).send();

});

});

app.listen(3000, function(){

console.log("Conecting...");

});

**Index.html**:

<html>

<head>

<script language="JavaScript" src="jquery-3.3.1.min.js"></script>

<script language="JavaScript" src="script.js"></script>

</head>

<body>

Уркраїнська - English

<form style="margin-bottom: 100px">

<input id="ua\_word"/>

<input type="button" onclick="translate\_ua()" value="Перекласти">

<input type="button" onclick="delete\_ua()" value="Видалити">

<br/><br/>

Переклад:

<input id="ua\_word\_translate"/>

<input type="button" onclick="add\_ua()" value="Додати в словник">

</form>

English - Уркраїнська

<form>

<input id="en\_word"/>

<input type="button" onclick="translate\_en()" value="Перекласти">

<input type="button" onclick="delete\_en()" value="Видалити">

<br/><br/>

Переклад:

<input id="en\_word\_translate"/>

<input type="button" onclick="add\_en()" value="Додати в словник">

</form>

</body>

</html>

**Script.js**:

var elementIds = {

ua\_word: "ua\_word",

ua\_word\_translate: "ua\_word\_translate",

en\_word: "en\_word",

en\_word\_translate: "en\_word\_translate"

}

var baseUrl = "http://localhost:3000";

var requestUrls = {

add\_ua: baseUrl + "/api/dictionary/ua\_word/", //:word

add\_en: baseUrl + "/api/dictionary/en\_word/", //:word

translate\_ua: baseUrl + "/api/dictionary/ua\_word/translate/", //:word

translate\_en: baseUrl + "/api/dictionary/en\_word/translate/", //:word

delete\_ua: baseUrl + "/api/dictionary/ua\_word/", //:word

delete\_en: baseUrl + "/api/dictionary/en\_word/" //:word

}

function add\_ua(){

var en\_word = $("#" + elementIds.ua\_word\_translate).val();

var ua\_word = $("#" + elementIds.ua\_word).val();

$.ajax({

url: requestUrls.add\_ua + ua\_word,

type: "POST",

contentType: "application/json",

data: JSON.stringify({ en\_word })

}).then(() => {

alert("Переклад додано!");

}).catch(() => {

alert("Помилка!");

});

}

function add\_en(){

var ua\_word = $("#" + elementIds.en\_word\_translate).val();

var en\_word = $("#" + elementIds.en\_word).val();

$.ajax({

url: requestUrls.add\_en + en\_word,

type: "POST",

contentType: "application/json",

data: JSON.stringify({ ua\_word })

}).then(() => {

alert("Переклад додано!");

}).catch(() => {

alert("Помилка!");

});

}

function translate\_ua(){

var ua\_word = $("#" + elementIds.ua\_word).val();

$.ajax({

url: requestUrls.translate\_ua + ua\_word,

type: "GET",

contentType: "application/json"

}).then(result => {

$("#" + elementIds.ua\_word\_translate).val(result);

}).catch(() => {

alert("Помилка!");

});

}

function translate\_en(){

var en\_word = $("#" + elementIds.en\_word).val();

$.ajax({

url: requestUrls.translate\_en + en\_word,

type: "GET",

contentType: "application/json"

}).then(result => {

$("#" + elementIds.en\_word\_translate).val(result);

}).catch(() => {

alert("Помилка!");

});

}

function delete\_ua(){

var ua\_word = $("#" + elementIds.ua\_word).val();

$.ajax({

url: requestUrls.delete\_ua + ua\_word,

type: "DELETE",

contentType: "application/json"

}).then(() => {

alert("Слово видалено!");

$("#" + elementIds.ua\_word\_translate).val("");

$("#" + elementIds.ua\_word).val("");

}).catch(() => {

alert("Помилка!");

});

}

function delete\_en(){

var en\_word = $("#" + elementIds.en\_word).val();

$.ajax({

url: requestUrls.delete\_en + en\_word,

type: "DELETE",

contentType: "application/json"

}).then(() => {

alert("Слово видалено!");

$("#" + elementIds.en\_word).val("");

$("#" + elementIds.en\_word\_translate).val("");

}).catch(() => {

alert("Помилка!");

});

}

**Dictionary.xml**:

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

<dictionary>

<писати>write</писати>

<мама>mother</мама>

<історія>history</історія>

<скрипт>sctipt</скрипт>

<сайт>site</сайт>

<картинка>image</картинка>

<світло>light</світло>

<сторінка>page</сторінка>

</dictionary>

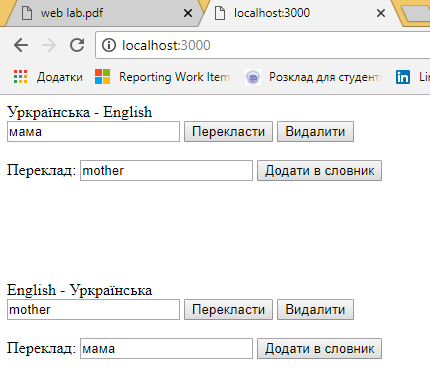


Рис. 1. Переклад в обидві сторони того ж слова.

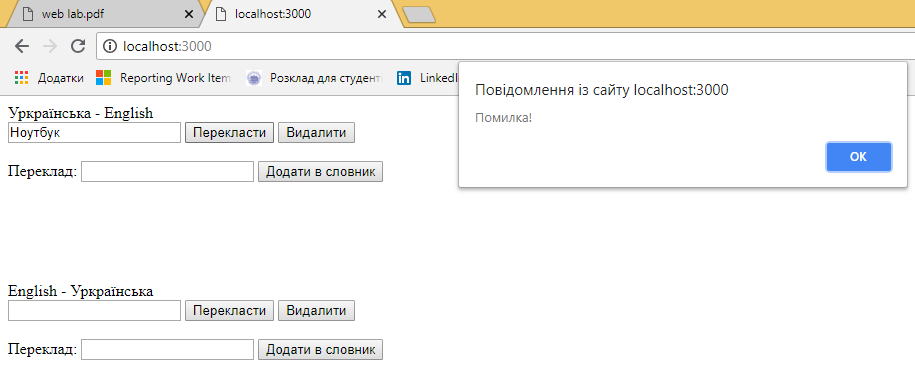


Рис. 2. Спроба перекладу неіснуючого або видаленого слова.

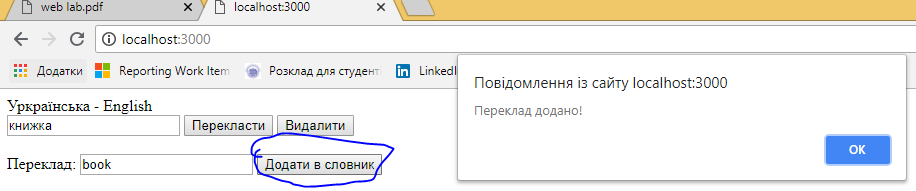


Рис. 3. Додавання слова у словник.

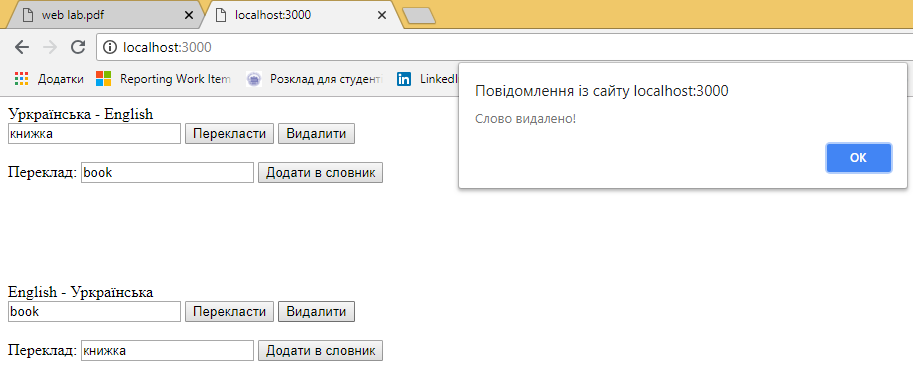


Рис. 4. Видалення слова зі словника.

**Висновок:** Виконавши дану лабораторну роботу ми вивчили основи асинхронного обміну даними мовою JavaScript за допомогою технології Ajax та бібліотеки JQuery при реалізації практичних задач Web-програмування.