



Assignments › Fall2021 Lab 5 AJAX, DB, Express, hosted at two origins

## Fall2021 Lab 5 AJAX, DB, Express, hosted at two origins

▼ Hide Folder Information

Group Category

AssignmentGroups

Group Name

U1

Instructions

In this lab two web hosting accounts will be used. One part of your assignment will be deployed to one hosting service, the other part will be hosted on another hosting service. The two will communicate via API calls.

The client side files (HTML or the js which initiate the AJAX calls) are being hosted at server1 (e.g. <https://yourDomainName1.ca>\* ). It shall send AJAX API calls to the API service which is hosted at another server, server 2. The API calls shall be used to ask the nodejs app to write onto the DB or read from the DB. ( remember that JavaScript that runs in browser cannot directly talk to DB. That is why it sends its requests to a server side script ( such as node js, PHP, APS .net etc) to do the DB related work.

The server2 ( <https://yourDomainName2.ca>\*) that hosts the node js app receives the API request from server1, communicate with the DB and responds back accordingly.

\*: http or https ( but if you are sharing the https, make sure ssl is already installed and your web pages do not issues any security risk warning ).  
Remember that you don't need to buy two domain names, you only need to have two hosting accounts ( please refer to the week 1 lecture notes or homepage of the course

### Learning outcome:

**Addressing CORS when cross communication between two origins.**  
**Connecting relational DB to a web server to do CRUD**  
**Using Express**

Pages visible to public users:

<https://yourDomainName1.ca/COMP4537/labs/4/writeDB.html> ,  
<https://yourDomainName1.ca/COMP4537/labs/4/readDB.html>

<https://yourDomainName2.ca> will be used to host node js files ( server side scripts, no page will be visible to public visitors of your website)

Note: As stated before you dont need a domain name to register. For example, yourDomainName1.ca could refer the root of your app at heroku, the other one in another server, such as yourDomainName....ca/com/etc or some other server other than heroku. ( if you are using heroku to host the nodejs files, use another service provider to host the client files)

### Writing to a DB, Reading from a DB,

In this lab, you are going to use AJAX request to send a string(name) and a number(score) from HTML client files (hosted in server 1) to be processed by nodejs server files (hosted in server 2) and to be stored in the DB table (table named score, hosted again in server 2, or any other server but not in server 1)

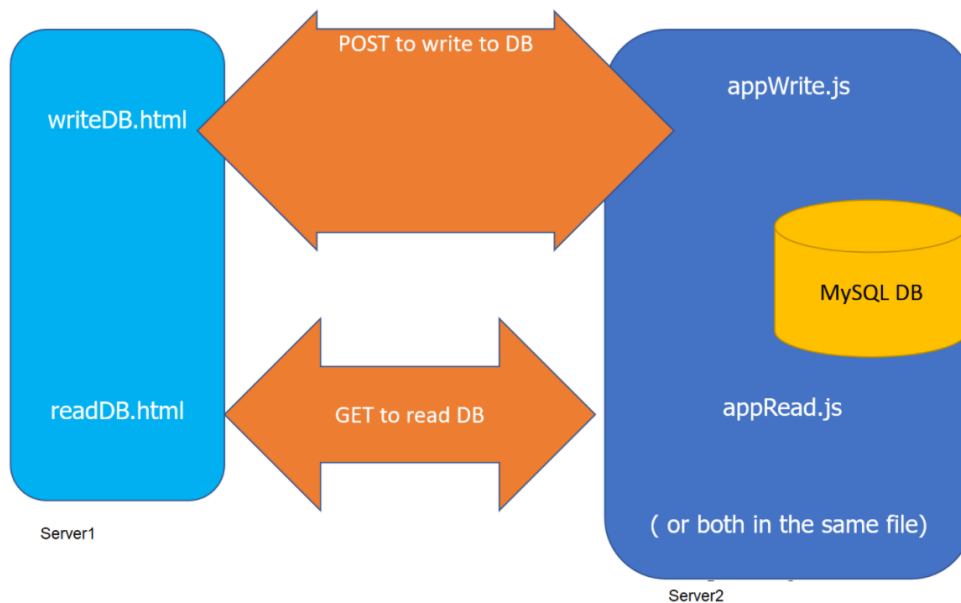
Submit

Cancel

connects to and communicates with the DB that you created.

(Note: it is easy for your TA to identify you have hosted these two parts in two different servers or only one server)

create a database like the one explained in the lecture with one table name **score** ( to store name and score just like the one explained in the lecture)



## Client side, hosted at server 1

**writeDB.html**

<https://yourDomainName1.ca/COMP4537/labs/4/writeDB.html>

Gets input from user and upon pressing the "store in DB", sends a POST request to the server side ( nodejs) hosted at server 2. Then stores them in a new row of the table score . Once the server side stores the data, it shall return a confirmation

such as

**Jahn:200 was stored in the DB**

and you need to show it on the client page underneath the button ( in a p tag, div tag or something)

Note: You don't no need to stack up the messages every time user sends a new request, just display the last message )

## Enter name and score

name:

score:

Note: you have to do some input **validation** making sure the score is a number ( you can use parseInt etc ... NAN..)

tip: this is how your query strings looks like in your GET request with multiple arguments : .../?  
name=John&score=23

( note: you can use GET request to send data to be stored in DB in this lab, later the better practice is to use a POST request for this)

## readDB.html

<https://yourDomainName1.ca/COMP4537/labs/4/readDB.html>

Once the user opens this page, it fetches all the data from the table score and dumps them here:

Jahn:200

Sarah:300

Jach:200

js files

<http://yourDomainName1.ca/COMP4537/labs/4/js/>

## Server side, hosted at partner 2 hosting account (server)

Your application makes a connection to DB to write and read accordingly

( AJAX call to the URI end points of the server 2)

### Technologies allowed to use

As stated before, before the midterm, including this assignment, you are only allowed to use the following technologies

- 1- Vanilla Javascript ( means pure javascript)
- 2- Node js
- 3- MySQL or any relational DB( SQL based DB) such as Postgres
- 4- External nodejs modules: mySQL or Express
- 5- HTML5
- 6- CSS, CSS3, Bootstrap ( no styling is needed for this lab, so no Bootstrap shall be used)

### Deliverable:

- 1) Source code of the server side in a directory then zipped ( compressed with \*.zip not rar) file named group#\_Server\_LastNameLabpartner1\_LastNameLabpartner2\_lab4.zip
- 2) Source code of the client side from the root directory zipped ( compressed with \*.zip not rar) file named group#\_Client\_LastNameLabpartner1\_LastNameLabpartner2\_lab4.zip
- 3) and the urls. e.g.

<https://yourDomainName1.ca/COMP4537/labs/4/writeDB.html>

<https://yourDomainName1.ca/COMP4537/labs/4/readDB.html>

( https or http)

### Rubrics (Deductions )

(-4) if the part client and server are from the same origin ( e.g. if are hosted on the same server)

(-4) if not hosted

(-4) if the urls are not posted at the comment sections ( even if only one url is missing)

up to -5 for not properly working of client side scripts

up to -5 for not properly working of server side scripts

(-3) if https/http are not posted together with your url ( for each url)

(-3) if you use JQuery or other libraries not allowed in your program

-10% mark deduction for each day late, 0 after three days late submission

Oct 25, 2021 3:30 PM

Nov 7, 2021 11:59 PM

## Files to submit \*

(0) file(s) to submit

After uploading, you must click **Submit** to complete the submission.

**Add a File**

Record Audio

Record Video

### Comments

A screenshot of a rich text editor toolbar. The toolbar contains various icons for text formatting and editing, including a paragraph selector, bold (B), italic (I), underline (U), strikethrough (A), bulleted list, numbered list, link, unlink, image, sum, table, and a plus sign for more options. The font is set to 'Lato (Recom...' and the text color is black. The background is a light gray.