

JavaScript Exercise

This exercise objective on using JavaScript to access and manipulate html elements including react DOM events. It is also working with JavaScript objects to handle requirements as the specifications by the following.

The screenshot shows a web browser window titled "Exercise Java Script" at the URL "localhost/exercise/exam.html". The interface is divided into three main sections, each with a colored border and numbered annotations:

- Top Section (Red border):** Contains a "Message :" input field with "Hello World" (1), a row of buttons: "Set To Layer" (2), "Switch Layer" (3), "Set To Layer In IFrame" (4), "Show Dialog" (5), "Print Frame" (6), and "Talk To Message Frame" (7, circled in blue). Below the buttons is the text "This is Example Layer #1" (8).
- Middle Section (Blue border):** Contains the text "This is demonstrate how to communicate between window/frames on the same origin or domain." (2, circled in blue), "This is Example Layer in IFRAME" (1), a "Hello Frame" input field (2), and a "Talk To Parent" button (3). Below is a paragraph: "This exercise objective on communicate between parent page and child frame java script in child frame can access the parent interface via object window.parent this can be made as a single page working with various html pages."
- Bottom Section (Green border):** Contains the text "This is demonstrate how to communicate between window/frames hosted on different domains." (3, circled in blue), "This is Message Layer in IFRAME" (1), a "Hello Message" input field (2), and a "Send Message To Parent" button (3). Below is a paragraph: "This exercise objective on communicate between parent page and child frame java script in child frame can access the parent interface via object window.parent using.postMessage and onmessage this can be made as a single page working with various html pages with out same origin policy."

Screen 1. exam.html



Screen 2. exam_dialog.html

Screen Definitions & Specifications

Screen 1. exam.html

1. exam.html this is main page

1.1 Message input text

1.2 Button Set To Layer

When this button click then set or assign layer text on 1.8 equals to message input text value from 1.1 (this is depending on current layer #1 and layer #2 visible too)

1.3 Button Switch Layer

When this button click then toggle visible and invisible layer on 1.8 between layer #1 and layer #2

ex. if current layer #1 visible then make it hide and bring layer #2 visible instead and if current layer #2 visible then make it hide and bring layer #1 visible

1.4 Button Set To Layer In IFrame

When this button click then set or assign layer in iframe on 2.1 equals to message input text value from 1.1

1.5 Button Show Dialog

When this button click then make window open as it's dialog from exam_dialog.html with the properties

- Window name = exam_dialog
- Width = 350
- Height = 350
- Make dialog as screen center
- Set or assign dialog layer from 4.1 equals to message input text from 1.1

1.6 Button Print Frame

When this button click then try to print only contents in iframe from 2 (exam_frame.html)

1.7 Button Talk To Message Frame

When this button click then using postMessage with value from message input text 1.1 to iframe from 3 (exam_message.html)

1.8 Layer #1 & Layer #2

1.9 In order to receive message from iframe try to handle onmessage please console out and display to message input text 1.1

2. exam_frame.html is in an iframe under exam.html or main page that come from with the same domain as exam.html (same origin)

2.1 Layer in iframe

2.2 Message input text

2.3 Button Talk To Parent

When this button click then set or assign message input text from 1.1 equals to message input text from 2.2

3. exam_message.html is in an iframe under exam.html or main page but it come from another domain (cross origin)

3.1 Layer in iframe

3.2 Message input text

3.3 Button Send Message To Parent

When this button click then using `postMessage` with value from message input text 3.2 to then parent (exam.html)

3.4 In order to receive message from parent try to handle `onmessage` please console out and display to message input text 3.3

4. exam_dialog.html make it as html dialog from opener

4.1 Dialog Layer

4.2 Button OK

When this button click then try to talk to the parent that this dialog is take action OK

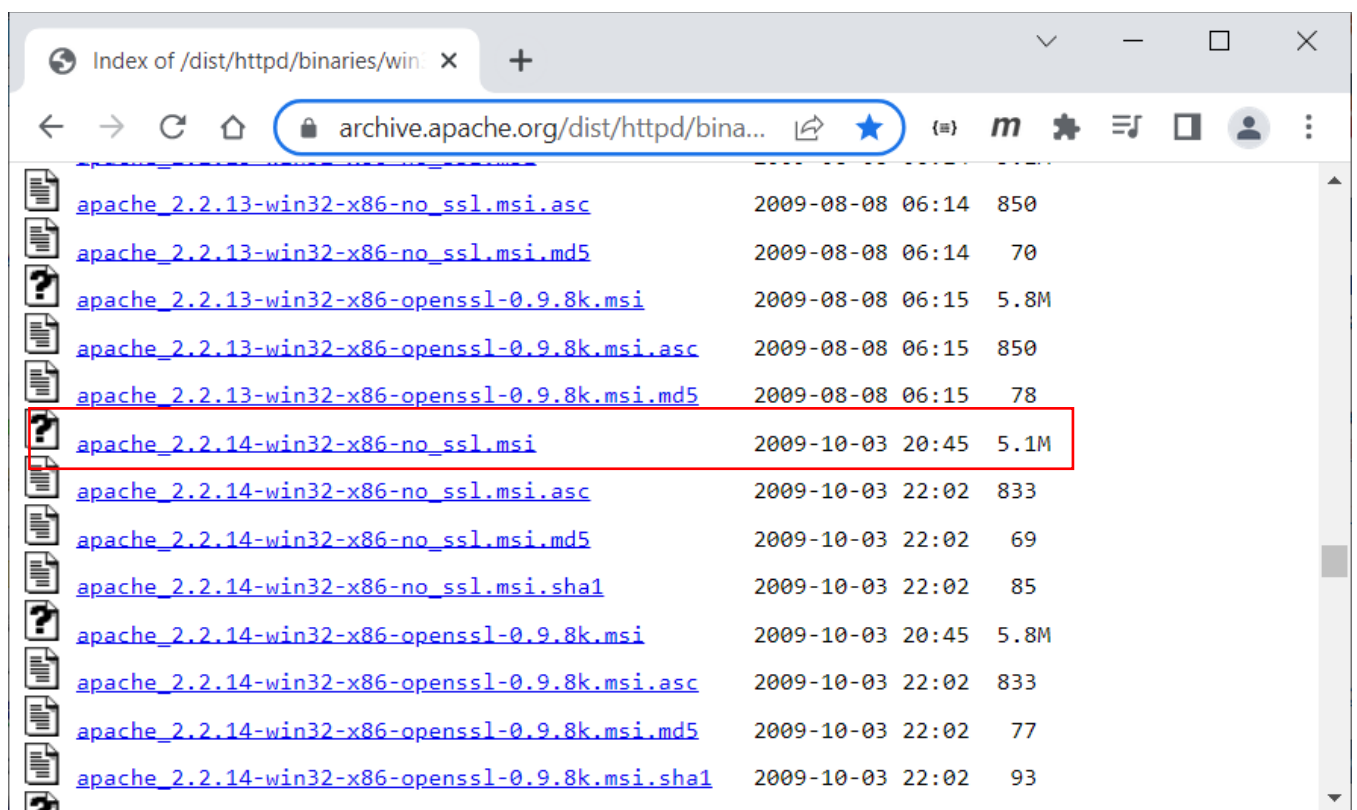
4.3 Button Cancel

When this button click then try to talk to the parent that this dialog is take action Cancel

Exercise Requirement & Setting

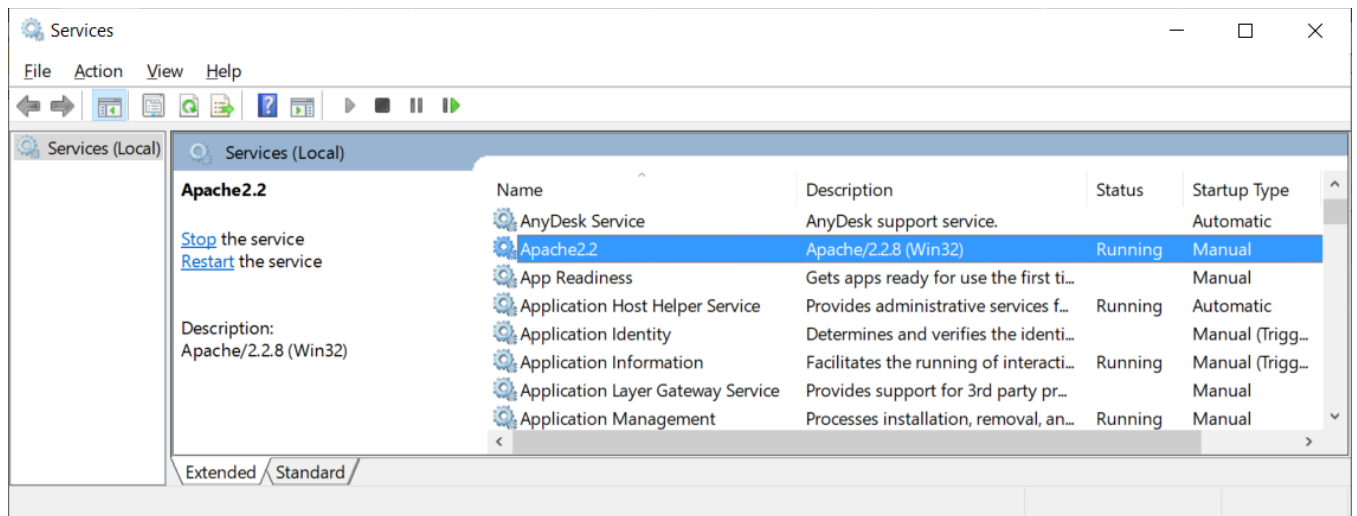
This exercise required web server to provide html page request by browser. We need to install and launch Apache web server.

1. Go to <https://httpd.apache.org/download.cgi>
(<https://archive.apache.org/dist/httpd/binaries/win32/>)



2. Download & Install

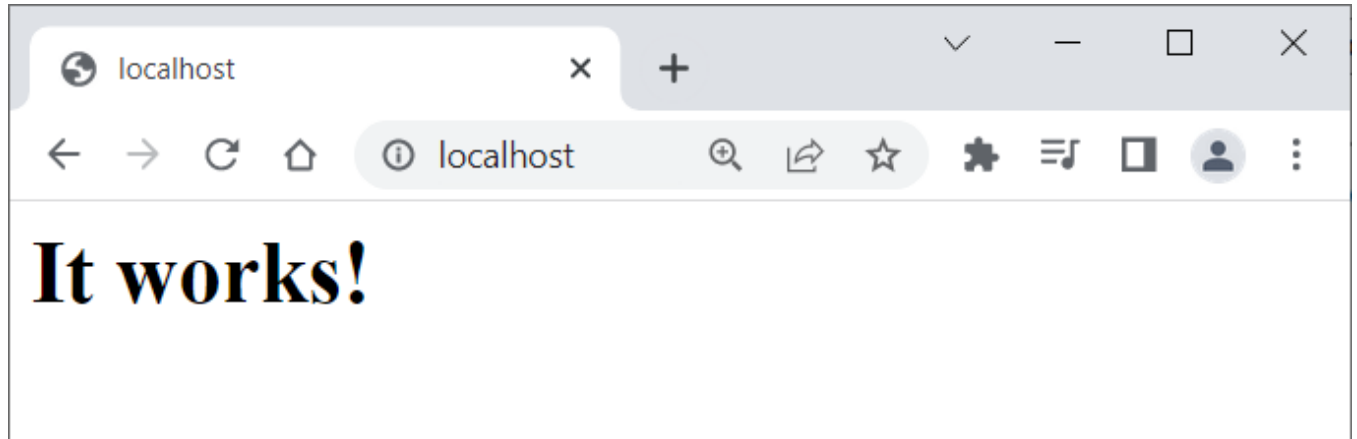
After install completed it will register as a windows service you can make it start/stop from windows service



Caution: this step may be conflict with other service like IIS then make it sure that other service stopped and this Apache service is running.

3. Test

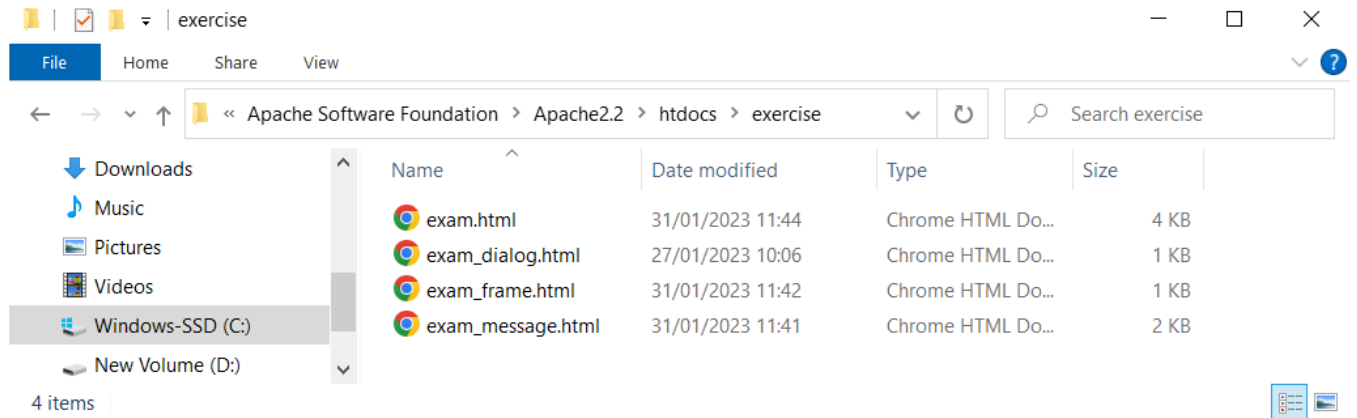
Try to go to <http://localhost/>



4. Setting Up

By the default root or working directory of Apache is htdocs folder under Apache installation path
 ex. C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs

try to copy exercise folder under training course material to htdocs folder

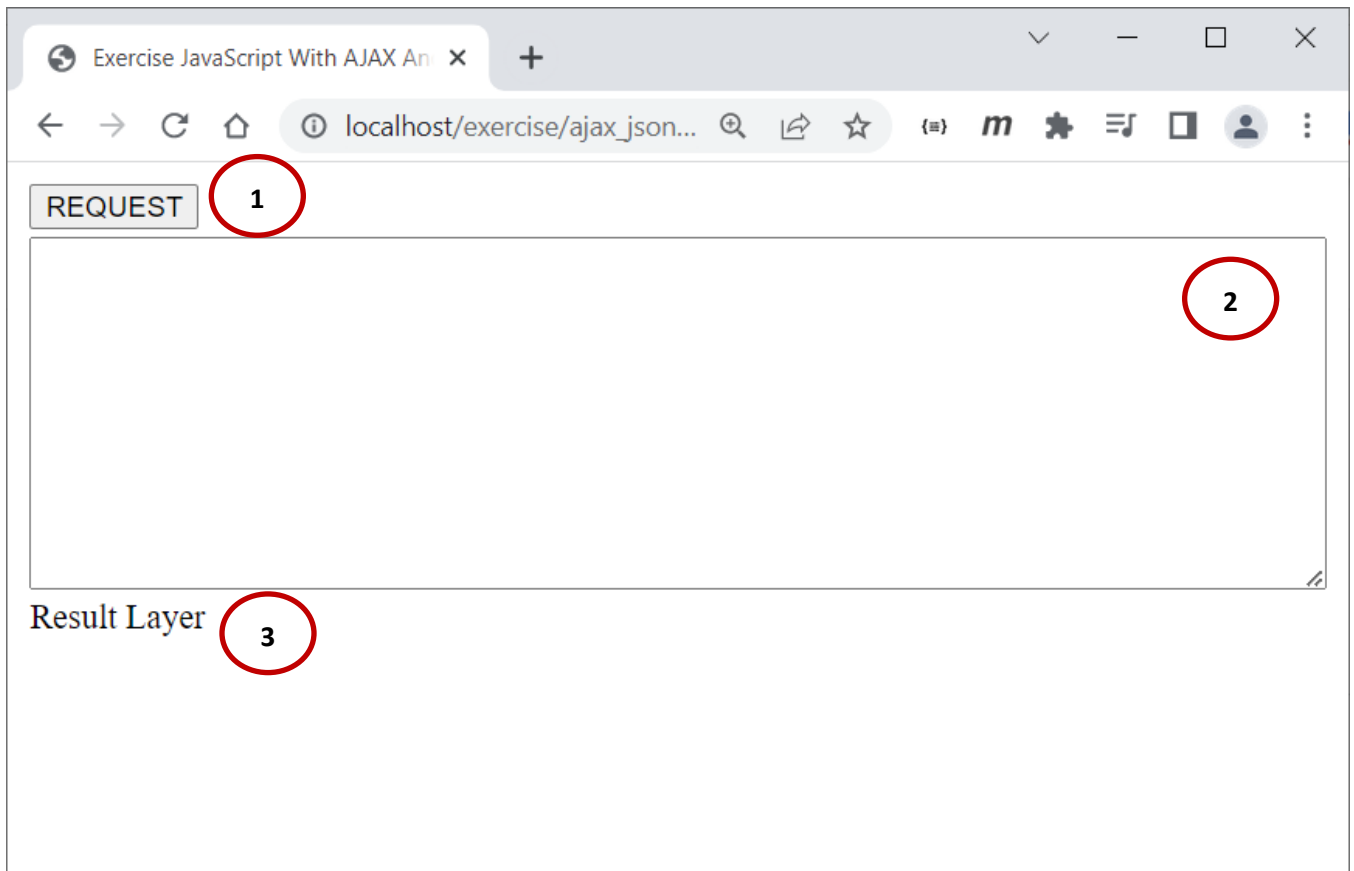


5. Launch Exercise

Try <http://localhost/exercise/exam.html>

JavaScript Exercise with AJAX & JSON

This exercise objective on using JavaScript with ajax & json data manipulate into html element by reading data in json format from ajax and then parse to json object.



Screen 3. ajax_json.html


```

D:\TrainingJavaScript\exercise\sample-datas.json - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
sample-datas.json x
1 {
2   "root": [
3     {"transno":1, "journalname":"Breakfast", "journaltype":"0", "amount": 120.0000, "remark":"Noodle + Coffee"},
4     {"transno":2, "journalname":"Lunch", "journaltype":"0", "amount": 100.0000, "remark":"Fry Rice"},
5     {"transno":3, "journalname":"Dinner", "journaltype":"0", "amount": 250.0000, "remark":"Steak"},
6     {"transno":4, "journalname":"Commission", "journaltype":"1", "amount": 1100.0000, "remark": null}
7   ]
8 }
9
JSON file length: 438 lines: 9 Ln: 1 Col: 5 Pos: 3 Windows (CR LF) UTF-8 INS

```

Screen 4. sample-datas.json.html

Exercise JavaScript With AJAX An x +

localhost/exercise/ajax_json...

REQUEST

```

{
  "root": [
    {"transno":1, "journalname":"Breakfast",
"journaltype":"0", "amount": 120.0000, "remark":"Noodle + Coffee"},
    {"transno":2, "journalname":"Lunch",
"journaltype":"0", "amount": 100.0000, "remark":"Fry Rice"},
    {"transno":3, "journalname":"Dinner",
"journaltype":"0", "amount": 250.0000, "remark":"Steak"},
    {"transno":4, "journalname":"Commission",
"journaltype":"1", "amount": 1100.0000, "remark": null}
  ]
}

```

# ¹	Transaction ²	Amount ³	Remark ⁴
1	Breakfast	120	Noodle + Coffee
2	Lunch	100	Fry Rice
3	Dinner	250	Steak
4	Commission	1100	
	Receive	5 1100	
	Payment	6 470	
	Balance	7 630	

Screen 5. ajax_json.html (Result)

Screen Definitions & Specifications

Screen 3. ajax_json.html

1. Button Request

When this button click try to ajax get sample-datas.json file with schema format json like Screen 4.

1.1 All file contents display to text area message from 2.

1.2 Try to parse and display json data into table from 3.

2. Text Area Message

3. Result Layer

Expect result by ajax from 1. with the following in screen 5.

Screen 5. ajax_json.html (Result)

1. Display transno value under root array elements

2. Display journalname value under root array elements

3. Display amount value under root array elements

4. Display remark value under root array elements (if null display blank)

5. Display receive amount value under root array elements (sum amount if journaltype='1')

6. Display pay amount value under root array elements (sum amount if journaltype='0')

7. Display balance amount = receive amount (5) – pay amount (6)