

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

WEB TECHNOLOGY LAB

BCSL504

EXPERIMENT – 10

AJAX USING JAVASCRIPT AND jQuery

Text File Loading and JSON Processing

5th Semester

AJAX without jQuery • AJAX with jQuery

getJSON() • JSON Parsing

HTML • CSS • JavaScript • AJAX

Prepared by Search Creators

Learn • Practice • Execute

1. Experiment 10

Aim

Develop JavaScript AJAX programs using HTML and CSS to:

- Add text content from a text file by sending an AJAX request without using jQuery.
- Add text content from a text file by sending an AJAX request using jQuery.
- Demonstrate the use of the jQuery `getJSON()` method.
- Parse JSON data and display JSON values.

2. Requirements

- HTML5
- CSS
- JavaScript
- AJAX
- jQuery
- Modern web browser
- Internet connection when jQuery is loaded from a CDN
- A local or hosted web server for reliable AJAX file requests
- **Search Creators Compiler / Code Studio**

<https://searchcreators.org/compiler/>

3. Search Creators Compiler

Practice with Search Creators

Practice Web Technology Lab programs using the **Search Creators Compiler / Code Studio**.

Open Search Creators Compiler

<https://searchcreators.org/compiler/>

Experiment 10 uses external text and JSON files.

Therefore, the execution environment must support multiple project files and HTTP requests between those files.

If the selected environment does not support such file requests, execute the same files through a suitable local or hosted web server.

4. Introduction to AJAX

AJAX stands for:

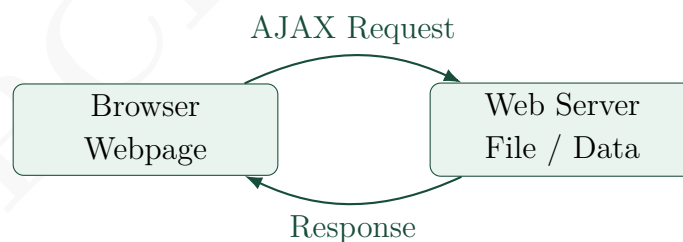
Asynchronous JavaScript and XML

AJAX allows a webpage to communicate with a server and update part of the webpage without reloading the entire page.

AJAX can exchange different types of data, including:

- Plain text
- JSON
- XML
- HTML

5. Basic Working of AJAX



The webpage sends a request.

The server returns the required data.

JavaScript then updates only the required portion of the webpage.

6. Files Used in Experiment 10

The following files can be used:

File	Purpose
data.txt	Text content used in Parts (a) and (b).
student.json	JSON data used in Part (c).
withoutjquery.html	AJAX without jQuery.
withjquery.html	AJAX using jQuery.
getjson.html	Demonstrates jQuery <code>getJSON()</code> .
parsejson.html	Parses JSON text and displays values.

7. Create data.txt

Create a file named:

data.txt

Add:

```
1 Welcome to Web Technology Lab.  
2 This content is loaded using AJAX.  
3 BCSL504 - Experiment 10.
```

8. Part (a): AJAX without jQuery

8.1 Aim

Add text content from a text file by sending an AJAX request without using jQuery.

8.2 Program – withoutjquery.html

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3  
4 <head>  
5  
6 <meta charset="UTF-8">  
7  
8 <title>AJAX without jQuery</title>  
9  
10 <style>  
11  
12 body {  
13     font-family: Arial, sans-serif;  
14     background-color: #eaf4ef;  
15     text-align: center;  
16 }
```

```
17
18 .container {
19     width: 600px;
20     margin: 60px auto;
21     padding: 25px;
22     background-color: white;
23     border: 2px solid darkgreen;
24 }
25
26 h2 {
27     color: darkgreen;
28 }
29
30 button {
31     padding: 10px 20px;
32     font-size: 16px;
33     cursor: pointer;
34 }
35
36 #output {
37     margin-top: 20px;
38     padding: 15px;
39     background-color: lightyellow;
40     white-space: pre-line;
41     font-size: 18px;
42 }
43
44 </style>
45
46 </head>
47
48 <body>
49
50 <div class="container">
51
52 <h2>AJAX without jQuery</h2>
53
54 <button onclick="loadText()">
55     Load Text File
56 </button>
57
58 <div id="output">
59     Text will appear here
60 </div>
61
62 </div>
63
64 <script>
```

```
65
66 function loadText() {
67
68     const xhr =
69         new XMLHttpRequest();
70
71     xhr.open(
72         "GET",
73         "data.txt",
74         true
75     );
76
77     xhr.onload = function() {
78
79         if (
80             xhr.status >= 200 &&
81             xhr.status < 300
82         ) {
83
84             document.getElementById(
85                 "output"
86             ).textContent =
87                 xhr.responseText;
88
89         } else {
90
91             document.getElementById(
92                 "output"
93             ).textContent =
94                 "Unable to load data.";
95         }
96     };
97
98     xhr.onerror = function() {
99
100         document.getElementById(
101             "output"
102         ).textContent =
103             "Request failed.";
104     };
105
106     xhr.send();
107 }
108
109 </script>
110
111 </body>
112 </html>
```

8.3 Explanation

The AJAX request object is created using:

```
1 const xhr =  
2   new XMLHttpRequest();
```

The request is prepared using:

```
1 xhr.open(  
2   "GET",  
3   "data.txt",  
4   true  
5 );
```

Here:

- GET is the HTTP method.
- data.txt is the requested resource.
- true specifies an asynchronous request.

The request is sent using:

```
1 xhr.send();
```

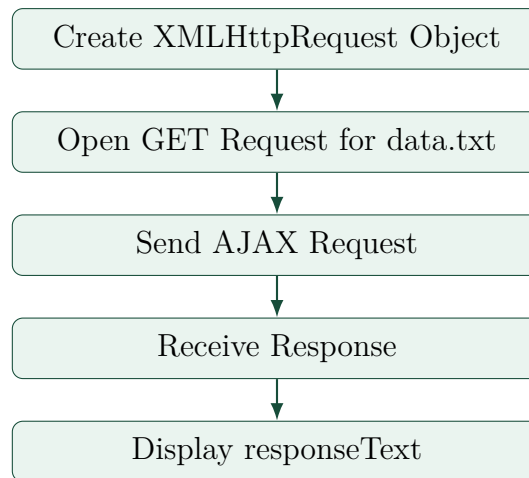
The returned text is available in:

```
1 xhr.responseText
```

and displayed using:

```
1 document.getElementById(  
2   "output"  
3 ).textContent =  
4   xhr.responseText;
```

9. XMLHttpRequest Flow



10. Part (b): AJAX with jQuery

10.1 Aim

Add text content from a text file by sending an AJAX request using jQuery.

10.2 Program – withjquery.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5
6 <meta charset="UTF-8">
7
8 <title>AJAX with jQuery</title>
9
10 <script
11 src="https://code.jquery.com/jquery-3.7.1.min.js">
12 </script>
13
14 <style>
15
16 body {
17     font-family: Arial, sans-serif;
18     background-color: #f2f2f2;
19     text-align: center;
20 }
21
22 .container {
```

```
23     width: 600px;
24     margin: 60px auto;
25     padding: 25px;
26     background-color: white;
27     border: 2px solid darkgreen;
28 }
29
30 h2 {
31     color: darkgreen;
32 }
33
34 button {
35     padding: 10px 20px;
36     font-size: 16px;
37 }
38
39 #output {
40     margin-top: 20px;
41     padding: 15px;
42     background-color: lightyellow;
43     white-space: pre-line;
44     font-size: 18px;
45 }
46
47 </style>
48
49 </head>
50
51 <body>
52
53 <div class="container">
54
55 <h2>AJAX with jQuery</h2>
56
57 <button id="loadButton">
58     Load Text File
59 </button>
60
61 <div id="output">
62     Text will appear here
63 </div>
64
65 </div>
66
67 <script>
68
69 $(document).ready(function() {
70
```

```
71     $("#loadButton").click(function() {  
72  
73         $.ajax({  
74  
75             url: "data.txt",  
76  
77             type: "GET",  
78  
79             dataType: "text",  
80  
81             success: function(data) {  
82  
83                 $("#output").text(data);  
84             },  
85  
86             error: function() {  
87  
88                 $("#output").text(  
89                     "Unable to load data."  
90                 );  
91             }  
92  
93         });  
94  
95     });  
96  
97 });  
98  
99 </script>  
100  
101 </body>  
102 </html>
```

10.3 Explanation

The jQuery AJAX request is created using:

```
1 $.ajax({  
2     ...  
3 });
```

The resource is specified using:

```
1 url: "data.txt"
```

The HTTP method is:

```
1 type: "GET"
```

The expected response is text:

```
1 dataType: "text"
```

When the request succeeds:

```
1 success: function(data) {
2
3     $("#output").text(data);
4 }
```

The returned text is displayed on the webpage.

11. Without jQuery vs With jQuery

Feature	Without jQuery	With jQuery
Request API	XMLHttpRequest	\$.ajax()
Library Required	No	Yes, jQuery
Request Method	xhr.open()	AJAX configuration
Send Request	xhr.send()	Handled by \$.ajax()
Success Handling	onload	success callback
Output	DOM API	jQuery selector/method

12. Part (c): jQuery getJSON()

12.1 Aim

Demonstrate the use of the jQuery `getJSON()` method to retrieve JSON data.

13. Create student.json

Create:

```
student.json
```

```
1 {
2     "name": "Ravi Kumar",
3     "usn": "1XX22CS001",
4     "semester": 5,
5     "course": "BCSL504",
6     "subject": "Web Technology Lab"
7 }
```

13.1 Program – getjson.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5
6 <meta charset="UTF-8">
7
8 <title>jQuery getJSON Example</title>
9
10 <script
11 src="https://code.jquery.com/jquery-3.7.1.min.js">
12 </script>
13
14 <style>
15
16 body {
17     font-family: Arial, sans-serif;
18     background-color: #eaf4ef;
19 }
20
21 .container {
22     width: 600px;
23     margin: 50px auto;
24     padding: 25px;
25     background-color: white;
26     border: 2px solid darkgreen;
27 }
28
29 h2 {
30     color: darkgreen;
31     text-align: center;
32 }
33
34 button {
35     display: block;
36     margin: 20px auto;
37     padding: 10px 20px;
38 }
39
40 #output {
41     background-color: lightyellow;
42     padding: 15px;
43     font-size: 18px;
44 }
45
46 </style>
```

```
47
48 </head>
49
50 <body>
51
52 <div class="container">
53
54 <h2>Student Details using getJSON()</h2>
55
56 <button id="loadJSON">
57     Load Student Details
58 </button>
59
60 <div id="output">
61     Student details will appear here.
62 </div>
63
64 </div>
65
66 <script>
67
68 $(document).ready(function() {
69
70     $("#loadJSON").click(function() {
71
72         $.getJSON(
73             "student.json",
74             function(data) {
75
76                 $("#output").html(
77                     "<b>Name:</b> " +
78                     data.name +
79
80                     "<br><b>USN:</b> " +
81                     data.usn +
82
83                     "<br><b>Semester:</b> " +
84                     data.semester +
85
86                     "<br><b>Course:</b> " +
87                     data.course +
88
89                     "<br><b>Subject:</b> " +
90                     data.subject
91                 );
92             }
93         ).fail(function() {
94
```

```
95         $("#output").text(  
96             "Unable to load JSON data."  
97         );  
98     });  
99  
100 });  
101  
102 });  
103  
104 </script>  
105  
106 </body>  
107 </html>
```

13.2 Explanation

The JSON file is loaded using:

```
1 $.getJSON(  
2     "student.json",  
3     function(data) {  
4         ...  
5     }  
6 );
```

The returned JSON values can be accessed using:

```
1 data.name  
2 data.usn  
3 data.semester  
4 data.course  
5 data.subject
```

Therefore:

student.json → getJSON() → JavaScript Data

14. Part (d): Parse and Display JSON Values

14.1 Aim

Parse JSON text and display its values on a webpage.

14.2 Program – parsejson.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5
6 <meta charset="UTF-8">
7
8 <title>Parse JSON</title>
9
10 <style>
11
12 body {
13     font-family: Arial, sans-serif;
14     background-color: #f2f2f2;
15     text-align: center;
16 }
17
18 .container {
19     width: 600px;
20     margin: 50px auto;
21     padding: 25px;
22     background-color: white;
23     border: 2px solid darkgreen;
24 }
25
26 h2 {
27     color: darkgreen;
28 }
29
30 button {
31     padding: 10px 20px;
32     font-size: 16px;
33 }
34
35 #output {
36     margin-top: 20px;
37     padding: 15px;
38     background-color: lightyellow;
39     font-size: 18px;
40 }
41
42 </style>
43
44 </head>
45
46 <body>
47
```

```
48 <div class="container">
49
50 <h2>Parse JSON and Display Values</h2>
51
52 <button onclick="displayJSON()">
53     Display JSON Values
54 </button>
55
56 <div id="output">
57     JSON values will appear here.
58 </div>
59
60 </div>
61
62 <script>
63
64 function displayJSON() {
65
66     const jsonText =
67         '{"name":"Asha",' +
68         '"usn":"1XX22CS002",' +
69         '"semester":5,' +
70         '"course":"BCSL504"}';
71
72     const student =
73         JSON.parse(jsonText);
74
75     document.getElementById(
76         "output"
77     ).innerHTML =
78
79         "<b>Name:</b> " +
80         student.name +
81
82         "<br><b>USN:</b> " +
83         student.usn +
84
85         "<br><b>Semester:</b> " +
86         student.semester +
87
88         "<br><b>Course:</b> " +
89         student.course;
90 }
91
92 </script>
93
94 </body>
95 </html>
```

15. JSON Parsing Explanation

Initially, the JSON data is only text:

```
1 const jsonText =  
2   '{"name":"Asha","semester":5}';
```

It is converted into a JavaScript object using:

```
1 const student =  
2   JSON.parse(jsonText);
```

After parsing, the values can be accessed using:

```
1 student.name  
2 student.semester
```

The process is:

JSON Text $\xrightarrow{\text{JSON.parse()}}$ JavaScript Object

16. Note on parseJSON()

Important

The syllabus refers to **parseJSON()** for displaying JSON values.
Older jQuery code commonly used:

`$.parseJSON(jsonText)`

For current JavaScript code, the native method:

`JSON.parse(jsonText)`

is used in this experiment.

The objective remains the same:

JSON Text $\rightarrow$ Parsed Object $\rightarrow$ Display Values

17. Older Syllabus Syntax

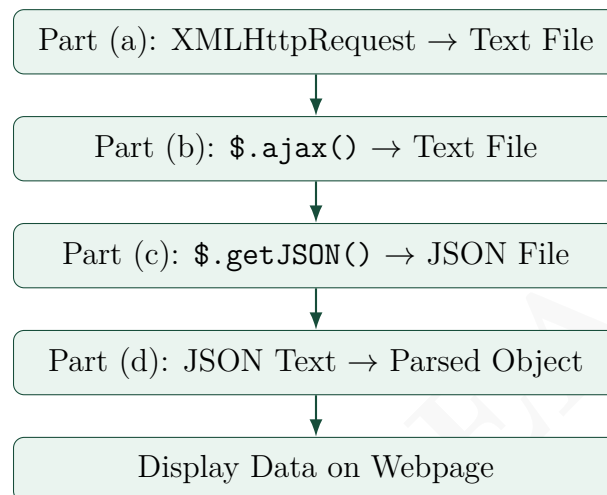
For understanding the syllabus terminology, the older style can be represented as:

```
1 // Older jQuery-oriented syntax  
2 var student =  
3   $.parseJSON(jsonText);
```

The current implementation used in this manual is:

```
1 // Current JavaScript syntax
2 const student =
3   JSON.parse(jsonText);
```

18. Overall Experiment Flow



19. Important AJAX Concepts

Concept	Purpose
AJAX	Communicates with a server without reloading the entire page.
XMLHttpRequest	Browser API for making HTTP requests.
GET	HTTP method commonly used to retrieve data.
responseText	Contains the response as text.
\$.ajax()	jQuery method for AJAX requests.
\$.getJSON()	jQuery method for retrieving JSON data.
JSON.parse()	Converts JSON text into a JavaScript value/object.

20. How to Execute

Practice with Search Creators

Start your coding practice from:

Search Creators Compiler / Code Studio

<https://searchcreators.org/compiler/>

For this AJAX experiment, make sure the execution environment supports the required project files and HTTP file requests.

Keep the required files together in the same project or server directory.

21. Procedure

1. Open the Search Creators Compiler / Code Studio or another suitable web development environment.
2. Create `data.txt`.
3. Add the sample text content.
4. Create `withoutjquery.html`.
5. Create an `XMLHttpRequest` object.
6. Open a GET request for `data.txt`.
7. Send the AJAX request.
8. Display the returned `responseText`.
9. Create `withjquery.html`.
10. Include the jQuery library.
11. Use `$.ajax()` to request `data.txt`.
12. Display the returned text.
13. Create `student.json`.
14. Enter the sample student JSON data.
15. Create `getjson.html`.
16. Use `$.getJSON()` to retrieve `student.json`.
17. Display the returned JSON values.
18. Create `parsejson.html`.
19. Create JSON text inside JavaScript.
20. Parse the JSON text using `JSON.parse()`.
21. Display the parsed values.
22. Execute the files through a suitable web server/environment and verify all outputs.

22. Expected Output – Part (a)

After clicking **Load Text File**:

Welcome to Web Technology Lab.
This content is loaded using AJAX.
BCSL504 – Experiment 10.

23. Expected Output – Part (b)

The same text content from `data.txt` is displayed using the jQuery AJAX request.

24. Expected Output – Part (c)

Name: Ravi Kumar
USN: 1XX22CS001
Semester: 5
Course: BCSL504
Subject: Web Technology Lab

25. Expected Output – Part (d)

Name: Asha
USN: 1XX22CS002
Semester: 5
Course: BCSL504

26. Important Viva Questions

Important Viva Questions

1. **What is AJAX?**

AJAX stands for Asynchronous JavaScript and XML.

2. **What is the main advantage of AJAX?**

It allows data to be exchanged with a server and parts of a webpage to be updated without reloading the entire page.

3. Does AJAX work only with XML?

No. AJAX can work with text, JSON, XML, HTML and other response formats.

4. What is XMLHttpRequest?

It is a browser API used to make HTTP requests from JavaScript.

5. Which method is used to prepare an XMLHttpRequest?

`open()`

6. Which method sends an XMLHttpRequest?

`send()`

7. What does `responseText` contain?

It contains the server response as text.

8. Which HTTP method is used in this experiment?

GET.

9. What is `$.ajax()`?

It is a jQuery method used to perform AJAX requests.

10. What is `getJSON()`?

It is a jQuery method used to retrieve JSON data.

11. What is JSON?

JSON stands for JavaScript Object Notation.

12. What does `JSON.parse()` do?

It parses valid JSON text and returns the corresponding JavaScript value.

13. What does the syllabus term `parseJSON()` refer to?

It refers to parsing JSON text so that its values can be accessed and displayed.

14. What is the current native JavaScript method used to parse JSON text?

`JSON.parse()`

15. Why may AJAX fail when an HTML file is opened directly using `file:///`?

Browsers apply security restrictions to local file requests. Running the files through a web server is more reliable.

16. What is asynchronous communication?

It allows the request to be processed without blocking the entire webpage while waiting for the response.

17. Which file is used for the text AJAX examples?

`data.txt`

18. Which file stores JSON data in this experiment?

`student.json`

19. Which Search Creators platform can be used for coding practice?

Search Creators Compiler / Code Studio

27. Quick Revision

Concept	Remember
AJAX	Asynchronous JavaScript and XML
Without jQuery	<code>XMLHttpRequest</code>
Prepare Request	<code>open()</code>
Send Request	<code>send()</code>
Text Response	<code>responseText</code>
jQuery AJAX	<code>\$.ajax()</code>
JSON Request	<code>\$.getJSON()</code>
JSON Parsing	<code>JSON.parse()</code>
HTTP Method Used	GET
Text File	<code>data.txt</code>
JSON File	<code>student.json</code>
Frontend	HTML + CSS + JavaScript
Library	jQuery
Practice Platform	Search Creators Compiler
Website	searchcreators.org/compiler/

28. Result

Result

JavaScript AJAX programs using HTML and CSS were successfully developed to:

- Load text from a text file using `XMLHttpRequest` without jQuery.
- Load text from a text file using the jQuery `$.ajax()` method.
- Retrieve JSON data using the jQuery `$.getJSON()` method.
- Parse JSON text and display its values using `JSON.parse()`.

For Search Creators coding practice:

Search Creators Compiler / Code Studio

<https://searchcreators.org/compiler/>

END OF EXPERIMENT – 10

BCSL504 – Web Technology Lab

Prepared by Search Creators

Practice Online: searchcreators.org/compiler/
