

LATIHAN PERTEMUAN 8
PRAKTIKUM PEMOGRAMAN BERBASIS WEB
Untuk Memenuhi Praktikum Pemograman Berbasis Web



Oleh:

Nama : Siti Ghumaisa
NPM : 4522210131
Kelas : A
Semester : 4 (Genap)

Dosen :

ADI WAHYU PRIBADI ,S.SI.,M.KOM
S1-Teknik Informatika
Fakultas Teknik Universitas Pancasila
2023/2024

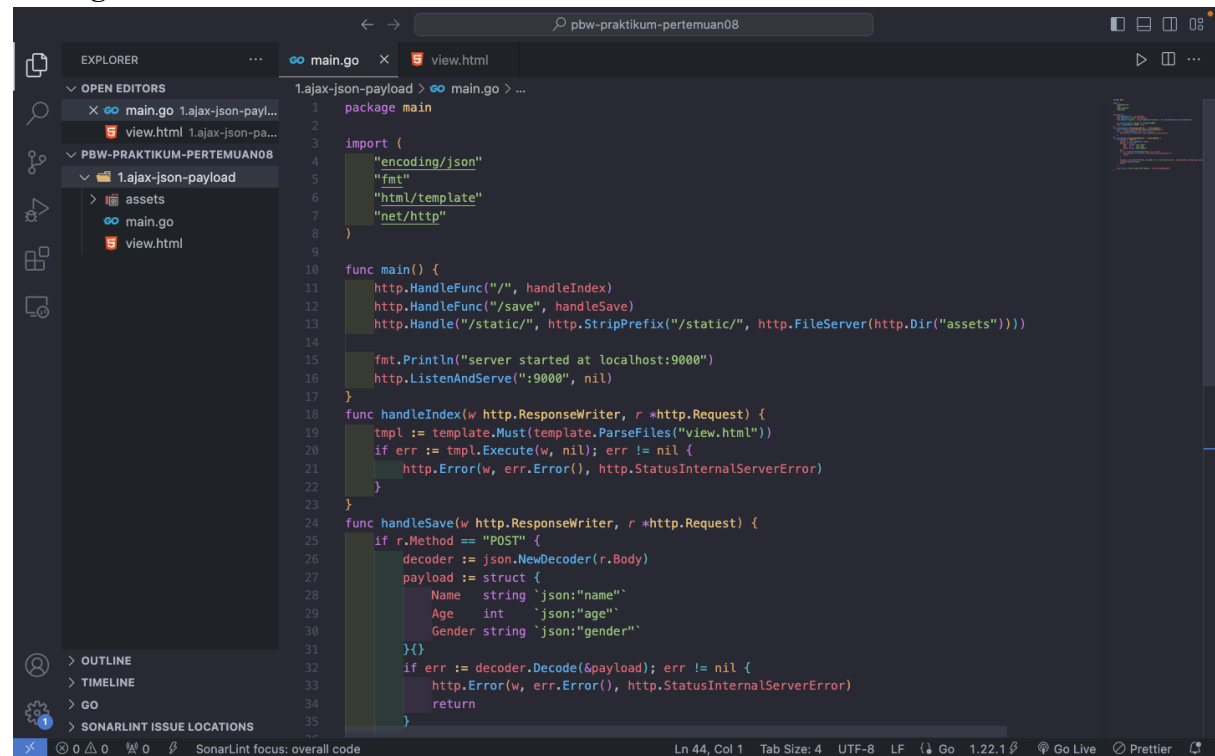
Link git-hub <https://github.com/Ghumaisa/PBW>

MATERI PRAKTIKUM :

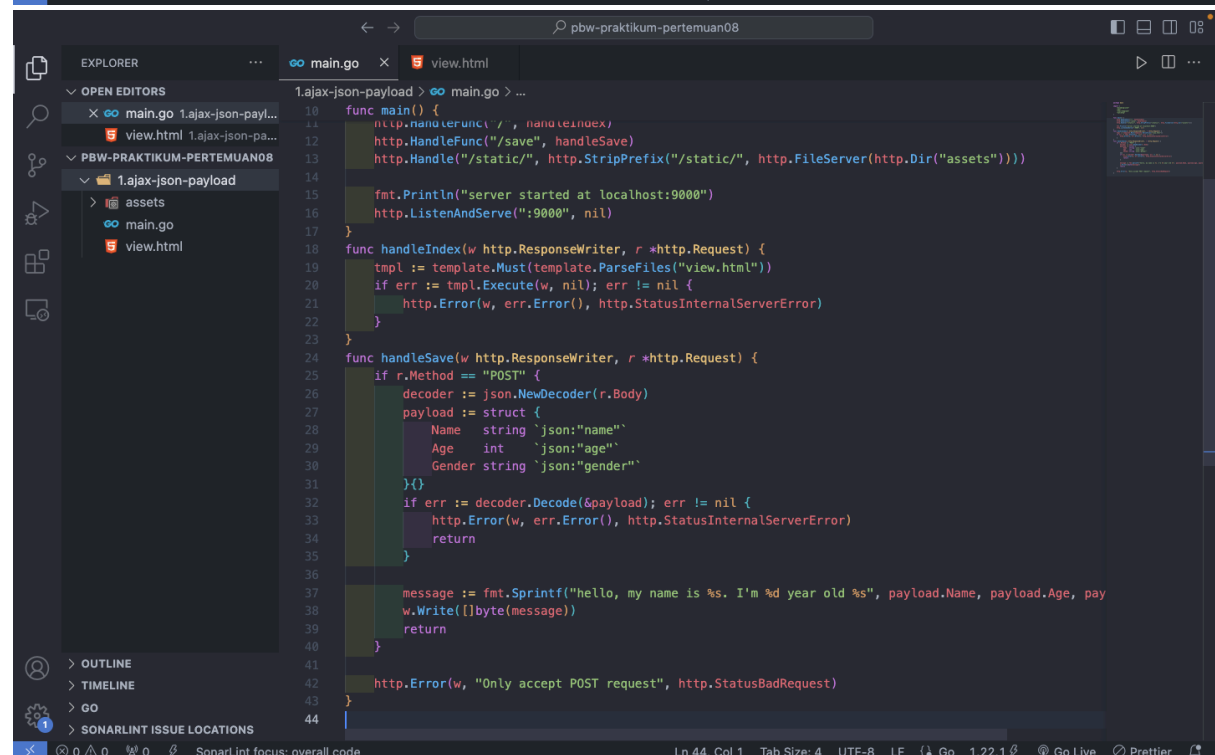
1. AJAX JSON Payload

Screenshot Source Code

main.go

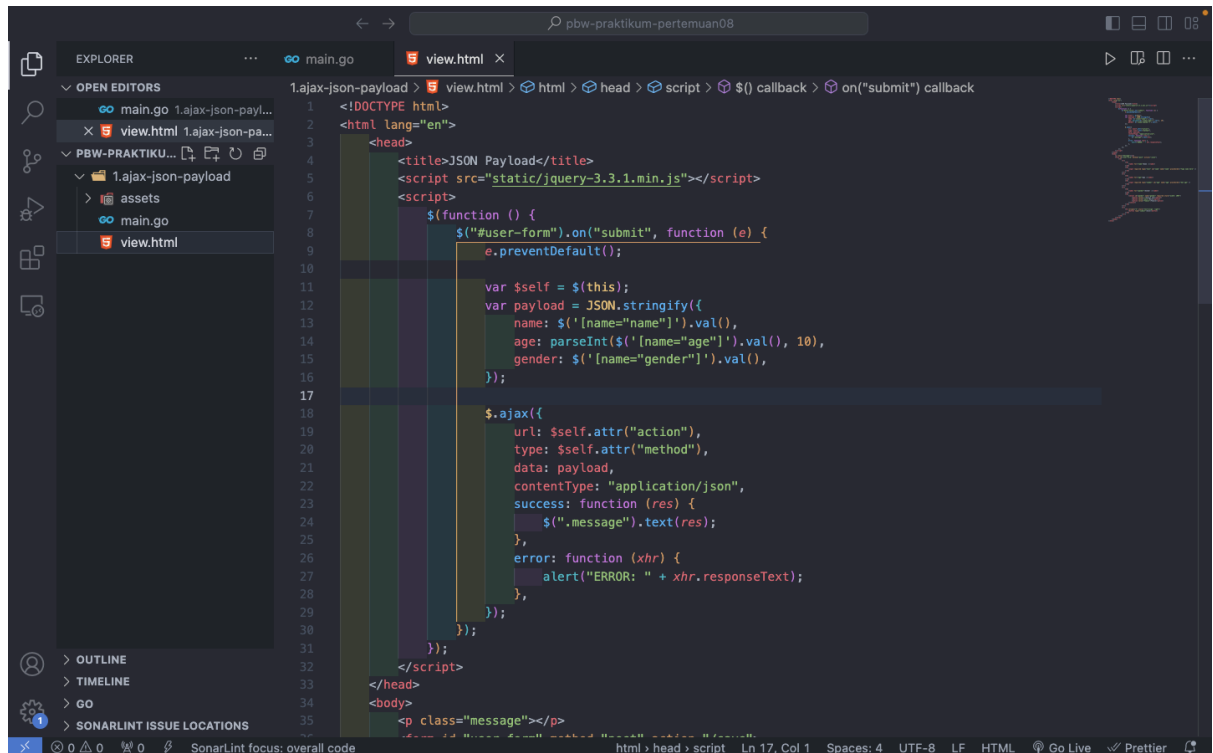


```
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6     "html/template"
7     "net/http"
8 )
9
10 func main() {
11     http.HandleFunc("/", handleIndex)
12     http.HandleFunc("/save", handleSave)
13     http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("assets"))))
14
15     fmt.Println("server started at localhost:9000")
16     http.ListenAndServe(":9000", nil)
17 }
18
19 func handleIndex(w http.ResponseWriter, r *http.Request) {
20     tmpl := template.Must(template.ParseFiles("view.html"))
21     if err := tmpl.Execute(w, nil); err != nil {
22         http.Error(w, err.Error(), http.StatusInternalServerError)
23     }
24 }
25
26 func handleSave(w http.ResponseWriter, r *http.Request) {
27     if r.Method == "POST" {
28         decoder := json.NewDecoder(r.Body)
29         payload := struct {
30             Name string `json:"name"`
31             Age int `json:"age"`
32             Gender string `json:"gender"`
33         }{}
34         if err := decoder.Decode(&payload); err != nil {
35             http.Error(w, err.Error(), http.StatusInternalServerError)
36             return
37         }
38     }
39 }
```

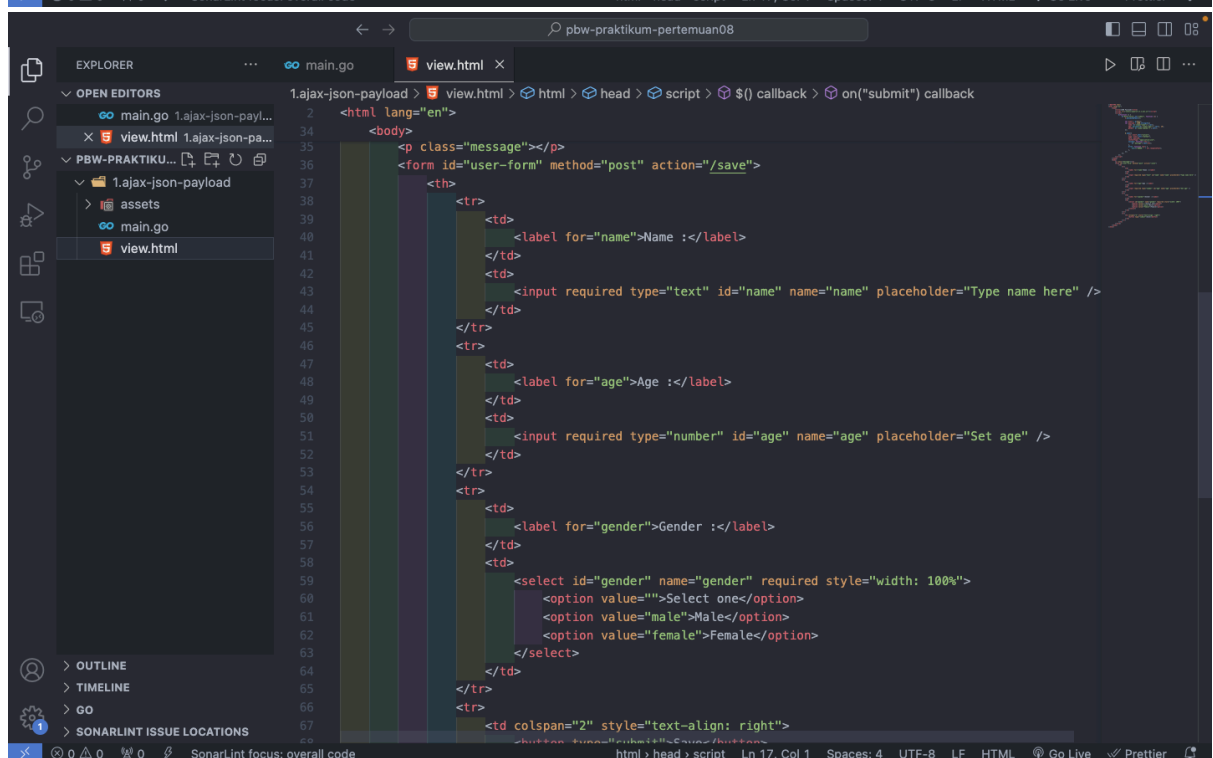


```
10 func main() {
11     http.HandleFunc("/", handleIndex)
12     http.HandleFunc("/save", handleSave)
13     http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("assets"))))
14
15     fmt.Println("server started at localhost:9000")
16     http.ListenAndServe(":9000", nil)
17 }
18
19 func handleIndex(w http.ResponseWriter, r *http.Request) {
20     tmpl := template.Must(template.ParseFiles("view.html"))
21     if err := tmpl.Execute(w, nil); err != nil {
22         http.Error(w, err.Error(), http.StatusInternalServerError)
23     }
24 }
25
26 func handleSave(w http.ResponseWriter, r *http.Request) {
27     if r.Method == "POST" {
28         decoder := json.NewDecoder(r.Body)
29         payload := struct {
30             Name string `json:"name"`
31             Age int `json:"age"`
32             Gender string `json:"gender"`
33         }{}
34         if err := decoder.Decode(&payload); err != nil {
35             http.Error(w, err.Error(), http.StatusInternalServerError)
36             return
37         }
38
39         message := fmt.Sprintf("hello, my name is %.s. I'm %d year old %.s", payload.Name, payload.Age, payload.Gender)
40         w.Write([]byte(message))
41     }
42     http.Error(w, "Only accept POST request", http.StatusBadRequest)
43 }
44 }
```

view.html



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>JSON Payload</title>
6   <script src="static/jquery-3.3.1.min.js"></script>
7   <script>
8     $(function () {
9       $("#user-form").on("submit", function (e) {
10        e.preventDefault();
11
12        var $self = $(this);
13        var payload = JSON.stringify({
14          name: $(''[name="name"]').val(),
15          age: parseInt($(''[name="age"]').val(), 10),
16          gender: $(''[name="gender"]').val(),
17        });
18
19        $.ajax({
20          url: $self.attr("action"),
21          type: $self.attr("method"),
22          data: payload,
23          contentType: "application/json",
24          success: function (res) {
25            $(".message").text(res);
26          },
27          error: function (xhr) {
28            alert("ERROR: " + xhr.responseText);
29          },
30        });
31      });
32    });
33  </script>
34 </head>
35 <body>
36   <p class="message"></p>
```



```
1 <html lang="en">
2
3 <body>
4   <p class="message"></p>
5   <form id="user-form" method="post" action="/save">
6     <table>
7       <tr>
8         <td>
9           <label for="name">Name :</label>
10        </td>
11        <td>
12          <input required type="text" id="name" name="name" placeholder="Type name here" />
13        </td>
14      </tr>
15      <tr>
16        <td>
17          <label for="age">Age :</label>
18        </td>
19        <td>
20          <input required type="number" id="age" name="age" placeholder="Set age" />
21        </td>
22      </tr>
23      <tr>
24        <td>
25          <label for="gender">Gender :</label>
26        </td>
27        <td>
28          <select id="gender" name="gender" required style="width: 100%">
29            <option value="">Select one</option>
30            <option value="male">Male</option>
31            <option value="female">Female</option>
32          </select>
33        </td>
34      </tr>
35      <tr>
36        <td colspan="2" style="text-align: right">
```

```
1 <html lang="en">
2 <body>
34 <form id="user-form" method="post" action="/save">
36 <th>
37 <tr>
45 <td>
46 <label for="age">Age :</label>
47 </td>
48 <td>
49 <input required type="number" id="age" name="age" placeholder="Set age" />
50 </td>
51 </tr>
52 <tr>
53 <td>
54 <label for="gender">Gender :</label>
55 </td>
56 <td>
57 <select id="gender" name="gender" required style="width: 100%">
58 <option value="">Select one</option>
59 <option value="male">Male</option>
60 <option value="female">Female</option>
61 </select>
62 </td>
63 </tr>
64 <tr>
65 <td colspan="2" style="text-align: right">
66 <button type="submit">Save</button>
67 </td>
68 </tr>
69 </th>
70 </tr>
71 </form>
72 </body>
73 </html>
```

Output

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <title>JSON Payload</title>
5 <script src="static/jquery-3.3.1.min.js"></script>
6 <script>
7 $(function () {
8     $('#user-form').on("submit", function (e) {
9         e.preventDefault();
10
11         var $self = $(this);
12         var payload = JSON.stringify({
13             name: $('[name="name"]').val(),
14             age: parseInt($('[name="age"]').val(), 10),
15             gender: $('[name="gender"]').val(),
16         });
17     });
18 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

1.ajax-json-payload git:(master) x go run main.go
server started at localhost:9000

go ghumaissa
zsh 1.ajax-json-p...
go 1.ajax-json-pa...

JSON Payload x +

localhost:9000

Name : Age : Gender :

Select one

Save

Memasukkan “Nama”, “Age”, dan “Gender”

JSON Payload x +

localhost:9000

Name : Age : Gender :

Female

Save

Setelah di klik save

hello, my name is Siti Ghumaisa. I'm 20 year old female

Name : Siti Ghumaisa Age : 20 Gender : Female

Save

hello, my name is Siti Ghumaisa. I'm 20 year old female

Name : Siti Ghumaisa Age : 20 Gender : Female

Save

Network panel: Request Payload

```
{name: "Siti Ghumaisa", age: 20, gender: "female"}
```

4 requests | 5.8 kB transferred

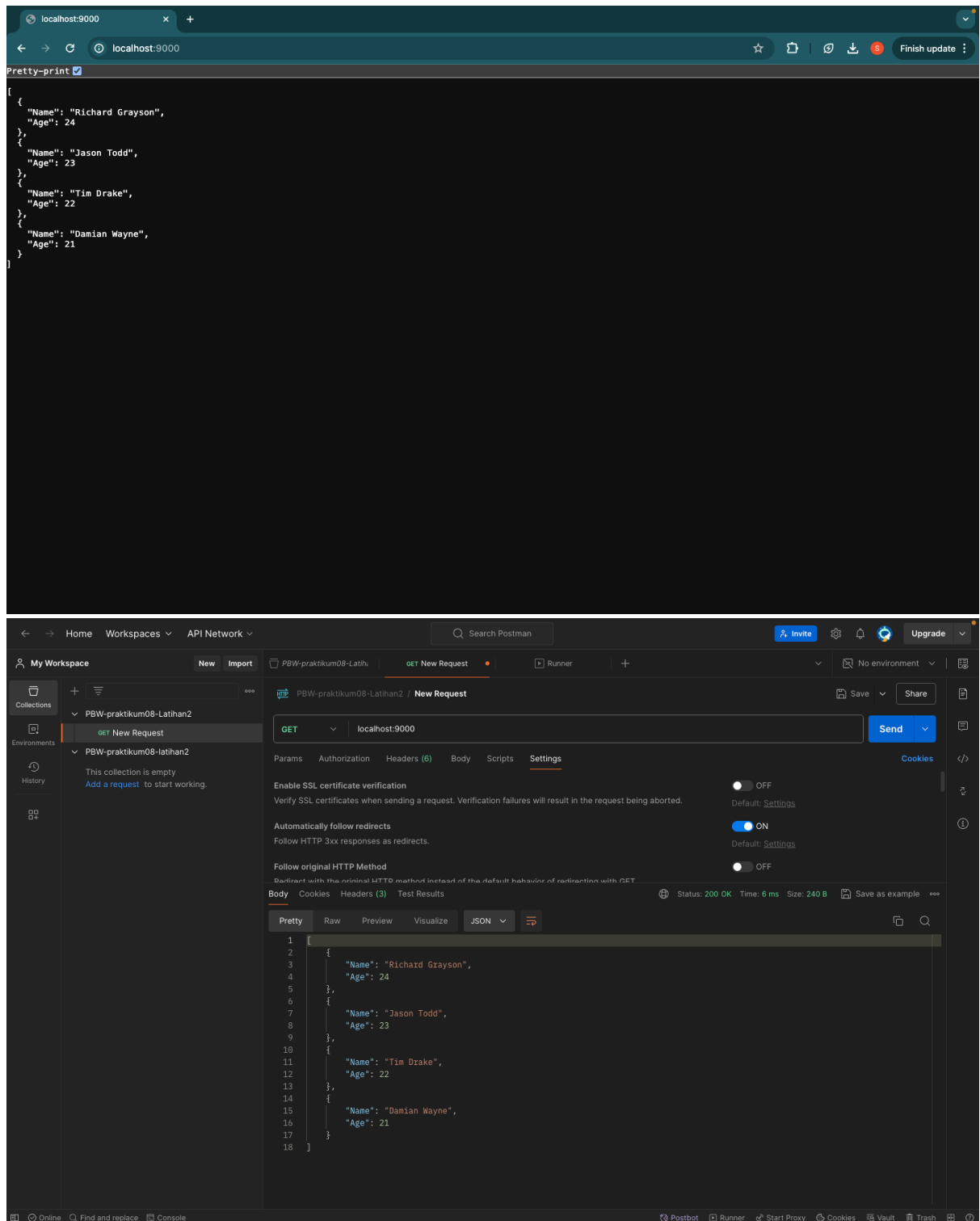
2. AJAX JSON Response

Screenshot Source Code

```
1 package main
2
3 import (
4     "encoding/json"
5     "fmt"
6     "net/http"
7 )
8
9 func main() {
10     http.HandleFunc("/", ActionIndex)
11
12     fmt.Println("server started at localhost:9000")
13     http.ListenAndServe(":9000", nil)
14 }
15
16 func ActionIndex(w http.ResponseWriter, r *http.Request) {
17     data := []struct {
18         Name string
19         Age  int
20     }{
21         {"Richard Grayson", 24},
22         {"Jason Todd", 23},
23         {"Tim Drake", 22},
24         {"Damian Wayne", 21},
25     }
26
27     jsonInBytes, err := json.Marshal(data)
28     if err != nil {
29         http.Error(w, err.Error(), http.StatusInternalServerError)
30         return
31     }
32
33     w.Header().Set("Content-Type", "application/json")
34     w.Write(jsonInBytes)
35 }
```

Output

```
2.ajax-json-response git:(master) x go run main.go
server started at localhost:9000
```



3. AJAX Multiple File Upload

Screenshot Source Code

Main.go


```
1 package main
2
3 import "fmt"
4 import "net/http"
5 import "html/template"
6 import "path/filepath"
7 import "io"
8 import "os"
9
10 func main() {
11     http.HandleFunc("/", handleIndex)
12     http.HandleFunc("/upload", handleUpload)
13     http.Handle("/static/", http.StripPrefix("/static/", http.FileServer(http.Dir("assets"))))
14
15     fmt.Println("server started at localhost:9000")
16     http.ListenAndServe(":9000", nil)
17 }
18
19 func handleIndex(w http.ResponseWriter, r *http.Request) {
20     tmpl := template.Must(template.ParseFiles("view.html"))
21     if err := tmpl.Execute(w, nil); err != nil {
22         http.Error(w, err.Error(), http.StatusInternalServerError)
23     }
24 }
25
26 func handleUpload(w http.ResponseWriter, r *http.Request) {
27     if r.Method != "POST" {
28         http.Error(w, "Only accept POST request", http.StatusBadRequest)
29         return
30     }
31
32     basePath, _ := os.Getwd()
33
34     reader, err := r.MultipartReader()
35     if err != nil {
```

```
36     if err != nil {
37         http.Error(w, err.Error(), http.StatusInternalServerError)
38         return
39     }
40
41     for {
42         part, err := reader.NextPart()
43         if err == io.EOF {
44             break
45         }
46
47         fileLocation := filepath.Join(basePath, "files", part.FileName())
48         dst, err := os.Create(fileLocation)
49         if dst != nil {
50             defer dst.Close()
51         }
52         if err != nil {
53             http.Error(w, err.Error(), http.StatusInternalServerError)
54             return
55         }
56
57         if _, err := io.Copy(dst, part); err != nil {
58             http.Error(w, err.Error(), http.StatusInternalServerError)
59             return
60         }
61
62         w.Write([]byte('all files uploaded'))
63     }
64 }
```

View.html

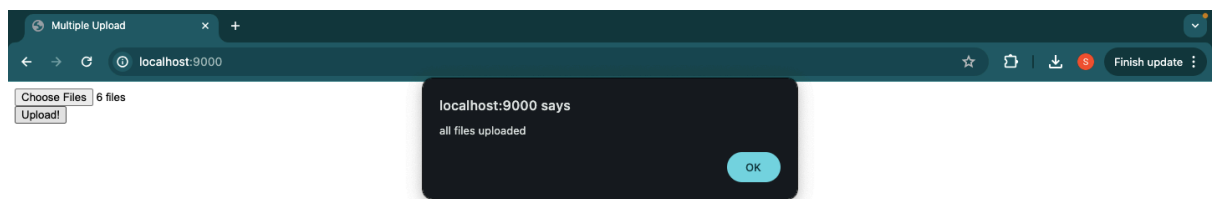
The screenshot shows the VS Code editor with the file `view.html` open. The Explorer sidebar on the left shows the project structure: `3.ajax-multi-upload` > `view.html`. The main editor area displays the following JavaScript code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Multiple Upload</title>
5 <script src="static/jquery-3.3.1.min.js"></script>
6 <script>
7     $(function () {
8         $("#user-form").on("submit", function (e) {
9             e.preventDefault();
10
11             var $self = $(this);
12             var files = $("#upload-file")[0].files;
13             var formData = new FormData();
14
15             for (var i = 0; i < files.length; i++) {
16                 formData.append("files", files[i]);
17             }
18
19             $.ajax({
20                 url: $self.attr("action"),
21                 type: $self.attr("method"),
22                 data: formData,
23                 processData: false,
24                 contentType: false,
25             }).then(function (res) {
26                 alert(res);
27                 $("#user-form").trigger("reset");
28             }).catch(function (a) {
29                 alert("ERROR: " + a.responseText);
30             });
31         });
32     });
33 </script>
34 </head>
35 <body>
```

The screenshot shows the VS Code editor with the file `view.html` open. The Explorer sidebar on the left shows the project structure: `3.ajax-multi-upload` > `view.html`. The main editor area displays the following HTML code:

```
1 <html>
2 <head>
3 <script>
4     $(function () {
5         $("#user-form").on("submit", function (e) {
6             alert(res);
7             $("#user-form").trigger("reset");
8         }).catch(function (a) {
9             alert("ERROR: " + a.responseText);
10         });
11     });
12 </script>
13 </head>
14 <body>
15 <form id="user-form" method="post" action="/upload">
16 <input required multiple id="upload-file" type="file" />
17 <br />
18 <button id="btn-upload" type="submit">Upload!</button>
19 </form>
20 </body>
21 </html>
```

Output



4. Download File

Screenshot Source Code

```
package main

import (
    "encoding/json"
    "fmt"
    "html/template"
    "io"
    "net/http"
    "os"
    "path/filepath"
)

type M map[string]interface{}

func main() {
    http.HandleFunc("/", handleIndex)
    http.HandleFunc("/list-files", handleListFiles)
    http.HandleFunc("/download", handleDownload)

    fmt.Println("server started at localhost:9000")
    http.ListenAndServe(":9000", nil)
}

func handleIndex(w http.ResponseWriter, r *http.Request) {
    tpl := template.Must(template.ParseFiles("view.html"))
    if err := tpl.Execute(w, nil); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
    }
}

func handleListFiles(w http.ResponseWriter, r *http.Request) {
    files := []M{}
    basePath, _ := os.Getwd()
    filesLocation := filepath.Join(basePath, "files")
    // ... (lines 31-35) ...
}
```

```
func handleListFiles(w http.ResponseWriter, r *http.Request) {
    // ... (lines 31-35) ...
    if err := os.Stat(filesLocation); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }
    if !err.IsDir() {
        return nil
    }
    files = append(files, M{"filename": info.Name(), "path": path})
    // ... (lines 41-45) ...
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }
    res, err := json.Marshal(files)
    if err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }
    w.Header().Set("Content-Type", "application/json")
    w.Write(res)
}

func handleDownload(w http.ResponseWriter, r *http.Request) {
    if err := r.ParseForm(); err != nil {
        http.Error(w, err.Error(), http.StatusInternalServerError)
        return
    }
    path := r.FormValue("path")
    // ... (lines 64-70) ...
}
```

```
31 func handleListFiles(w http.ResponseWriter, r *http.Request) {
32     res, err := json.Marshal(files)
33     if err != nil {
34         http.Error(w, err.Error(), http.StatusInternalServerError)
35         return
36     }
37     w.Header().Set("Content-Type", "application/json")
38     w.Write(res)
39 }
40
41 func handleDownload(w http.ResponseWriter, r *http.Request) {
42     if err := r.ParseForm(); err != nil {
43         http.Error(w, err.Error(), http.StatusInternalServerError)
44         return
45     }
46     path := r.FormValue("path")
47     f, err := os.Open(path)
48     if f != nil {
49         defer f.Close()
50     }
51     if err != nil {
52         http.Error(w, err.Error(), http.StatusInternalServerError)
53         return
54     }
55     contentDisposition := fmt.Sprintf("attachment; filename=%s", f.Name())
56     w.Header().Set("Content-Disposition", contentDisposition)
57
58     if _, err := io.Copy(w, f); err != nil {
59         http.Error(w, err.Error(), http.StatusInternalServerError)
60         return
61     }
62 }
```

View.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Download file</title>
5     <script>
6         function Yo() {
7             var self = this;
8             var $ul = document.getElementById("list-files");
9
10            var renderData = function (res) {
11                res.forEach(function (each) {
12                    var $li = document.createElement("li");
13                    var $a = document.createElement("a");
14
15                    $li.innerText = "download ";
16                    $li.appendChild($a);
17                    $ul.appendChild($li);
18
19                    $a.href = "/download?path=" + encodeURIComponent(each.path);
20                    $a.innerText = each.filename;
21                    $a.target = "_blank";
22                });
23            };
24
25            var getAllListFiles = function () {
26                var xhr = new XMLHttpRequest();
27                xhr.open("GET", "/list-files");
28                xhr.onreadystatechange = function () {
29                    if (xhr.readyState == 4 && xhr.status == 200) {
30                        var json = JSON.parse(xhr.responseText);
31                        renderData(json);
32                    }
33                };
34                xhr.send();
35            };
36        }
37    </script>
38 </head>
39 <body>
40     <div>
41         <ul id="list-files">
42             <li>download <a href="/download?path=" + encodeURIComponent(each.path) + ">each.filename</a>
43         </ul>
44     </div>
45 </body>
46 </html>
```

```
2 <html>
3 <head>
4 <script>
5     function Yo() {
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25         var getAllListFiles = function () {
26             var xhr = new XMLHttpRequest();
27             xhr.open("GET", "/list-files");
28             xhr.onreadystatechange = function () {
29                 if (xhr.readyState == 4 && xhr.status == 200) {
30                     var json = JSON.parse(xhr.responseText);
31                     renderData(json);
32                 }
33             };
34             xhr.send();
35         };
36
37         self.init = function () {
38             getAllListFiles();
39         };
40     };
41
42     window.onload = function () {
43         new Yo().init();
44     };
45 </script>
46 </head>
47 <body>
48     <ul id="list-files"></ul>
49 </body>
50 </html>
51
```

Output

