

# Folder Day4 Fetch API Assignment

## 3 printable files

(file list disabled)

### Day4 Fetch API Assignment\app.js

```
1 // =====
2 // ApiModule: Handles API calls using Fetch API
3 // =====
4 const ApiModule = (function() {
5   const BASE = 'https://jsonplaceholder.typicode.com';
6   // const BASE = 'https://invalid.example';
7
8   // Generic fetch wrapper with timeout & error handling
9   async function fetchJson(endpoint, timeoutMs = 10000) {
10     const controller = new AbortController();
11     const timeout = setTimeout(() => controller.abort(), timeoutMs);
12
13     try {
14       const res = await fetch(BASE + endpoint, { signal: controller.signal });
15       clearTimeout(timeout);
16
17       // Check for HTTP errors
18       if (!res.ok) throw new Error(`Server responded with ${res.status}`);
19
20       const data = await res.json();
21
22       // Ensure valid response format
23       if (!Array.isArray(data)) throw new Error('Unexpected data format');
24
25       return data;
26     } catch (err) {
27       if (err.name === 'AbortError') throw new Error('Request timed out');
28       throw err;
29     }
30   }
31
32   // Public methods (revealing module pattern)
33   return {
34     getPosts: () => fetchJson('/posts'),
35     getTodos: () => fetchJson('/todos')
36   };
37 })();
38
39 // =====
40 // UIModule: Handles DOM manipulation and UI rendering
41 // =====
42 const UIModule = (function() {
```

```
43 // DOM elements
44 const postsList = document.getElementById('postsList');
45 const todosList = document.getElementById('todosList');
46 const postsMeta = document.getElementById('postsMeta');
47 const globalStatus = document.getElementById('globalStatus');
48
49 // Helper: clear an element's children
50 function clear(el) {
51   while (el.firstChild) el.removeChild(el.firstChild);
52 }
53
54 // Helper: escape HTML (security)
55 function escapeHtml(text) {
56   const map = { '&': '&amp;', '<': '&lt;', '>': '&gt;', '"': '&quot;', "'": '&#039;' };
57   return String(text).replace(/[&<>"]/g, m => map[m]);
58 }
59
60 // Update global status message
61 function setGlobalStatus(msg) {
62   globalStatus.textContent = msg || '';
63 }
64
65 // Show loader while fetching data
66 function showLoaderIn(el) {
67   clear(el);
68   const d = document.createElement('div');
69   d.innerHTML = '<div class="loader"></div> Loading...';
70   el.appendChild(d);
71 }
72
73 // Render fetched posts
74 function renderPosts(posts) {
75   clear(postsList);
76   posts.slice(0, 30).forEach(p => {
77     const e = document.createElement('div');
78     e.className = 'post';
79     e.innerHTML = `<div class='title'>${escapeHtml(p.title)}</div>
80                   <div class='body'>${escapeHtml(p.body)}</div>`;
81     postsList.appendChild(e);
82   });
83   postsMeta.textContent = `Showing ${Math.min(posts.length, 30)} of ${posts.length} posts`;
84 }
85
86 // Render fetched todos (with optional filter)
87 function renderTodos(todos) {
88   clear(todosList);
89   todos.slice(0, 50).forEach(t => {
90     const r = document.createElement('div');
91     r.className = 'todo' + (t.completed ? ' completed' : '');
92     r.innerHTML = `<input type='checkbox' ${t.completed ? 'checked' : ''} disabled/>`
```

```
93         <div class='label' style='flex:1'>${escapeHtml(t.title)}</div>`;
94     todosList.appendChild(r);
95 });
96 }
97
98 // Render error message
99 function renderError(el, msg) {
100     clear(el);
101     const e = document.createElement('div');
102     e.className = 'error';
103     e.textContent = msg;
104     el.appendChild(e);
105 }
106
107 // Reveal only necessary functions
108 return { showLoaderIn, renderPosts, renderTodos, renderError, setGlobalStatus };
109 })();
110
111 // =====
112 // AppController: Main controller (handles events & flow)
113 // =====
114 const AppController = (function(Api, UI) {
115     const cache = { posts: null, todos: null };
116
117     // Load and render posts
118     async function loadPosts() {
119         UI.showLoaderIn(document.getElementById('postsList'));
120         try {
121             const p = cache.posts ?? await Api.getPosts();
122             cache.posts = p;
123             UI.renderPosts(p);
124         } catch (e) {
125             UI.renderError(document.getElementById('postsList'), 'Failed: ' + e.message);
126         }
127     }
128
129     // Load and render todos
130     async function loadTodos(filter = 'all') {
131         UI.showLoaderIn(document.getElementById('todosList'));
132         try {
133             const t = cache.todos ?? await Api.getTodos();
134             cache.todos = t;
135             const filtered = filter === 'incomplete' ? t.filter(x => !x.completed) : t;
136             UI.renderTodos(filtered);
137         } catch (e) {
138             UI.renderError(document.getElementById('todosList'), 'Failed: ' + e.message);
139         }
140     }
141
142     // Wire up event listeners for buttons
```

```

143 function wireEvents() {
144     document.getElementById('refreshBtn').addEventListener('click', async () => {
145         cache.posts = null;
146         cache.todos = null;
147         await Promise.all([loadPosts(), loadTodos()]);
148     });
149
150     document.getElementById('showIncomplete').addEventListener('click', () =>
loadTodos('incomplete'));
151     document.getElementById('showAll').addEventListener('click', () => loadTodos('all'));
152 }
153
154 // Initialize the app
155 async function init() {
156     wireEvents();
157     await Promise.all([loadPosts(), loadTodos()]);
158 }
159
160 // Reveal public methods
161 return { init };
162 })(ApiModule, UIModule);
163
164 // =====
165 // App Initialization
166 // =====
167 ApplicationController.init().catch(e => {
168     document.getElementById('globalStatus').textContent = 'Error: ' + e.message;
169 });
170

```

#### Day4 Fetch API Assignment\index.html

```

1 <!doctype html>
2 <html lang="en">
3 <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width,initial-scale=1" />
6     <title>Fetch API Assignment - Posts & Todos</title>
7
8     <!-- Link external stylesheet -->
9     <link rel="stylesheet" href="styles.css" />
10 </head>
11 <body>
12
13     <!-- ===== Header Section ===== -->
14     <header>
15         <div>
16             <h1>Blog Posts & Todos (Fetch API)</h1>
17             <small class="msg">Data retrieved from JSONPlaceholder (sample APIs)</small>
18         </div>
19     <!-- Refresh button and global status message -->

```

```

20     <div style="margin-left:auto; display:flex; gap:8px; align-items:center">
21         <button id="refreshBtn" class="btn">Refresh</button>
22         <div id="globalStatus" aria-live="polite"></div>
23     </div>
24 </header>
25
26 <!-- ===== Main Content ===== -->
27 <main class="container">
28
29     <!-- ===== Posts Section ===== -->
30     <section class="card" id="postsCard" aria-labelledby="postsTitle">
31         <h2 id="postsTitle">Posts</h2>
32         <div id="postsList" role="list" aria-live="polite"></div>
33         <div style="margin-top:12px; display:flex; gap:8px; align-items:center;">
34             <small id="postsMeta" class="msg"></small>
35         </div>
36     </section>
37
38     <!-- ===== Todos Section ===== -->
39     <aside class="card" id="todosCard" aria-labelledby="todosTitle">
40         <h2 id="todosTitle">Todos</h2>
41         <div id="todosList" role="list" aria-live="polite"></div>
42
43         <!-- Buttons to filter todos -->
44         <div style="margin-top:12px; display:flex; gap:8px;">
45             <button id="showIncomplete" class="secondary">Show Incomplete</button>
46             <button id="showAll" class="secondary">Show All</button>
47         </div>
48     </aside>
49 </main>
50
51 <!-- ===== Footer ===== -->
52 <footer>
53     <div>Implemented using Fetch API, error handling, and Revealing Module Pattern</div>
54 </footer>
55
56 <!-- Link external JavaScript -->
57 <script src="app.js"></script>
58 </body>
59 </html>
60

```



#### Day4 Fetch API Assignment\styles.css

```

1  /* =====
2    Global Styles & Variables
3    ===== */
4  :root {
5      --bg: #f5f7fb;
6      --card: #ffffff;
7      --muted: #6b7280;

```

```
8   --accent: #2563eb;
9   --danger: #ef4444;
10  font-family: Inter, ui-sans-serif, system-ui, -apple-system, "Segoe UI", Roboto, "Helvetica
    Neue", Arial;
11 }
12
13 /* =====
14    Body & Layout
15    ===== */
16 body {
17   margin: 0;
18   padding: 24px;
19   background: var(--bg);
20   color: #111827;
21 }
22
23 /* Header Styling */
24 header {
25   display: flex;
26   gap: 16px;
27   align-items: center;
28   margin-bottom: 16px;
29 }
30 h1 { margin: 0; font-size: 1.25rem; }
31
32 /* Main container: posts + todos columns */
33 .container {
34   display: grid;
35   grid-template-columns: 1fr 360px;
36   gap: 18px;
37   align-items: start;
38 }
39
40 /* Card styling for posts/todos panels */
41 .card {
42   background: var(--card);
43   border-radius: 12px;
44   box-shadow: 0 4px 18px rgba(16,24,40,0.06);
45   padding: 16px;
46 }
47 .card h2 { margin: 0 0 8px 0; font-size: 1rem; }
48
49 /* Scrollable lists */
50 #postsList, #todosList {
51   max-height: 70vh;
52   overflow: auto;
53   padding-right: 6px;
54 }
55
56 /* =====
```

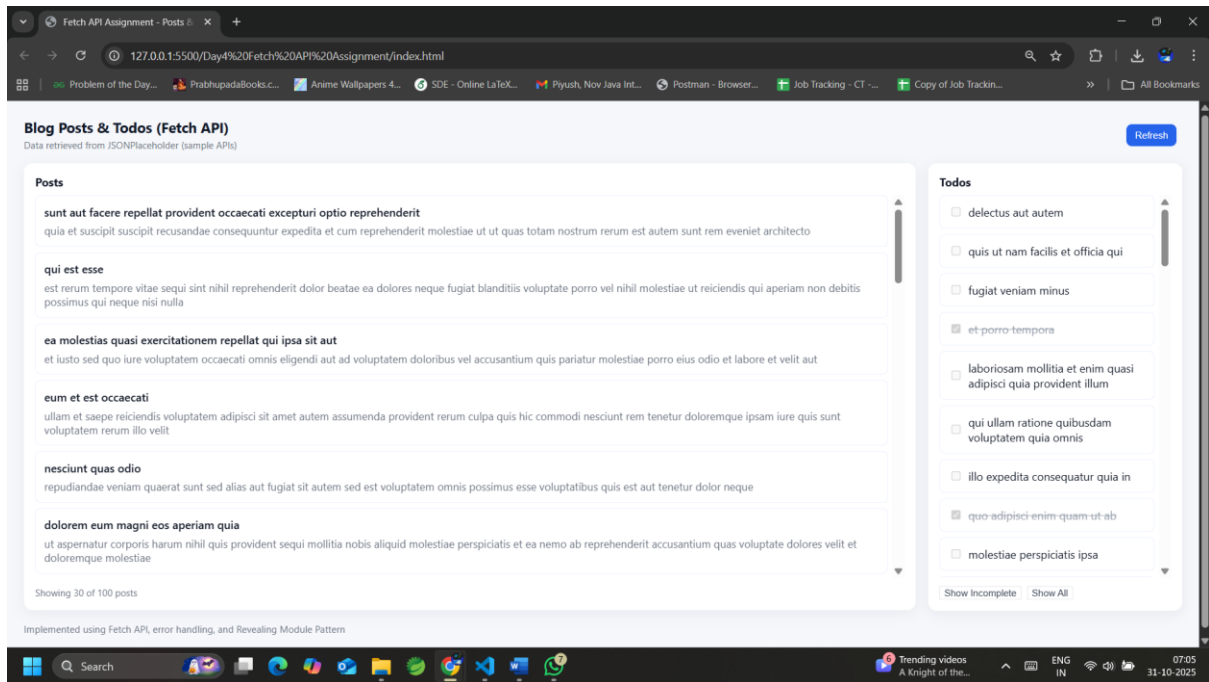
```
57     Post & Todo Items
58     ===== */
59     .post, .todo {
60         border: 1px solid #eef2ff;
61         padding: 12px;
62         border-radius: 8px;
63         margin-bottom: 8px;
64     }
65
66     /* Post title & body */
67     .post .title { font-weight:600; margin-bottom:6px; }
68     .post .body { color: var(--muted); font-size:0.95rem; }
69
70     /* Todo layout */
71     .todo {
72         display:flex;
73         align-items:center;
74         gap:8px;
75     }
76     .todo.completed .label { text-decoration: line-through; color:  #9ca3af; }
77
78     /* =====
79         Utility Classes
80         ===== */
81     small.msg { color: var(--muted); }
82     .error { color: var(--danger); font-weight:600; }
83
84     /* Loader animation */
85     .loader {
86         display:inline-block;
87         width:18px;
88         height:18px;
89         border:3px solid #e5e7eb;
90         border-top-color:var(--accent);
91         border-radius:50%;
92         animation:spin 1s linear infinite;
93     }
94     @keyframes spin { to { transform: rotate(360deg) } }
95
96     /* Footer styling */
97     footer { margin-top: 16px; font-size: 0.85rem; color: var(--muted); }
98
99     /* Buttons */
100    button.btn {
101        background: var(--accent);
102        color:  white
103        border:none;
104        padding:8px 12px;
105        border-radius:8px;
106        cursor:pointer;
```

```
107 }  
108 button.secondary {  
109   background:transparent;  
110   border:1px solid #e5e7eb;  
111   color: ■ #374151;  
112 }  
113
```



Hosted Website Link : <https://blogtrackr.netlify.app/>

## Output 1:



## Output 2: (when error occurs)

