

DYNAMIC PAGINATION WITH AJAX AND JAVASCRIPT

Overview:

This guide helps you create dynamic pagination buttons using Thymeleaf, JavaScript, and Spring Boot. It assumes your AJAX method gets paginated JSON data including `totalPages`, `currentPage`, and `pageSize`.

1. Sample JSON Response (from your AJAX call)

```
{  
  "content": [ { "id": 1, "name": "Item A" }, { "id": 2, "name": "Item B" } ],  
  "totalPages": 5,  
  "currentPage": 2,  
  "pageSize": 10  
}
```

2. HTML Structure

```
<div id="itemContainer"></div>  
  
<div id="paginationContainer"></div>
```

3. JavaScript Code

```
function loadPage(pageNumber = 0) {  
  fetch(`/api/items?page=${pageNumber}`)  
    .then(response => response.json())  
    .then(data => {  
      renderItems(data.content);  
      renderPagination(data.totalPages, data.currentPage);  
    })  
}
```

```
    })  
    .catch(error => console.error("Error fetching data:", error));  
}
```

```
function renderItem(items) {  
    const container = document.getElementById("itemContainer");  
    container.innerHTML = "";  
    items.forEach(item => {  
        const div = document.createElement('div');  
        div.textContent = item.name;  
        container.appendChild(div);  
    });  
}
```

```
function renderPagination(totalPages, currentPage) {  
    const container = document.getElementById("paginationContainer");  
    container.innerHTML = "";  
  
    for (let i = 0; i < totalPages; i++) {  
        const btn = document.createElement("button");  
        btn.textContent = i + 1;  
        btn.onclick = () => loadPage(i);  
        if (i === currentPage) {  
            btn.style.fontWeight = "bold";  
        }  
        container.appendChild(btn);  
    }  
}
```

```
}
```

```
// Initial load
```

```
loadPage(0);
```

4. Spring Boot Controller (Optional)

```
@GetMapping("/api/items")
```

```
@ResponseBody
```

```
public Page<Item> getItems(@RequestParam(defaultValue = "0") int page,
```

```
                           @RequestParam(defaultValue = "10") int size) {
```

```
    return itemService.getItems(PageRequest.of(page, size));
```

```
}
```

Output:

- Renders items and dynamic pagination buttons.
- Page changes are handled by JavaScript (no reload).