Absolutely! By keeping the base CDN link static and appending the CSS file name dynamically, the code becomes cleaner and more maintainable. Here's how to implement it.

**Updated themeSwitcher.js**

Place this file in the wwwroot/js/ directory.

// themeSwitcher.js

function initializeThemeSwitcher() {

// Base CDN link

const baseCdnUrl = "https://kendo.cdn.telerik.com/2021.2.511/styles/";

// Map themes to their corresponding CSS file names

const themes = {

"default-v2": "kendo.default-v2.min.css",

"bootstrap-v5": "kendo.bootstrap-v5.min.css",

"material-v2": "kendo.material-v2.min.css",

"blueopal": "kendo.blueopal.min.css",

"black": "kendo.black.min.css",

"fiori": "kendo.fiori.min.css",

"highcontrast": "kendo.highcontrast.min.css",

"moonlight": "kendo.moonlight.min.css",

"silver": "kendo.silver.min.css",

"nova": "kendo.nova.min.css"

};

// Load the saved theme from local storage

const savedTheme = localStorage.getItem("kendoTheme") || "default-v2";

// Apply the saved theme on page load

applyTheme(savedTheme);

// Listen for changes in the dropdown

$("#theme-switcher").on("change", function () {

const selectedTheme = $(this).val();

applyTheme(selectedTheme);

// Save the selected theme in local storage

localStorage.setItem("kendoTheme", selectedTheme);

});

// Function to apply a theme by changing the stylesheet URL

function applyTheme(themeKey) {

const themeFileName = themes[themeKey];

if (themeFileName) {

const themeUrl = baseCdnUrl + themeFileName;

$("#kendo-theme").attr("href", themeUrl);

}

}

}

**Updated \_Layout.cshtml**

This file integrates the dropdown and dynamically applies themes based on the selected value.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"] - My ASP.NET Core App</title>

<!-- Kendo UI Styles (Default theme loaded by default) -->

<link id="kendo-theme" href="https://kendo.cdn.telerik.com/2021.2.511/styles/kendo.default-v2.min.css" rel="stylesheet">

<!-- Kendo UI Scripts -->

<script src="https://cdn.kendostatic.com/2023.3.1107/js/jquery.min.js"></script>

<script src="https://cdn.kendostatic.com/2023.3.1107/js/kendo.all.min.js"></script>

<!-- Custom ThemeSwitcher Script -->

<script src="/js/themeSwitcher.js"></script>

<style>

#theme-container {

margin: 10px;

padding: 10px;

display: flex;

align-items: center;

}

#theme-switcher {

width: 200px;

}

</style>

</head>

<body>

<header>

<div>

<h1>My ASP.NET Core App</h1>

</div>

<!-- Theme Switcher UI -->

<div id="theme-container">

<label for="theme-switcher" style="margin-right: 10px;">Choose Theme:</label>

<select id="theme-switcher">

<option value="default-v2">Default</option>

<option value="bootstrap-v5">Bootstrap</option>

<option value="material-v2">Material</option>

<option value="blueopal">BlueOpal</option>

<option value="black">Black</option>

<option value="fiori">Fiori</option>

<option value="highcontrast">HighContrast</option>

<option value="moonlight">Moonlight</option>

<option value="silver">Silver</option>

<option value="nova">Nova</option>

</select>

</div>

</header>

<main role="main" class="container">

@RenderBody()

</main>

<footer>

<p>&copy; 2024 - My ASP.NET Core App</p>

</footer>

<!-- Initialize ThemeSwitcher -->

<script>

$(document).ready(function () {

initializeThemeSwitcher(); // Call the function from themeSwitcher.js

});

</script>

</body>

</html>

**How It Works**

1. **Base CDN URL:**  
   The baseCdnUrl variable holds the static portion of the CDN URL. The theme file name is appended dynamically.
2. **Themes Object:**  
   The themes object maps the theme names (used in the dropdown) to their corresponding CSS file names.
3. **Dynamic Link Update:**  
   The applyTheme function constructs the full URL by appending the theme file name to the base CDN URL and updates the href attribute of the <link> tag.
4. **Persistence:**  
   The selected theme is saved to localStorage, ensuring the preference persists across page reloads.

**Testing Instructions**

1. **Run the App:** Launch your application and verify that the dropdown shows all theme options.
2. **Switch Themes:** Change the selection in the dropdown. The styles should update dynamically.
3. **Refresh:** Refresh the page and ensure the previously selected theme is applied automatically.

This implementation minimizes duplication and ensures the code is clean and maintainable. Let me know if you need further assistance! 🚀