

UI5 Virtual Theme Switching – Integrációs Guide

Áttekintés

Ez a guide bemutatja, hogyan építhetsz be **runtime téma váltást** egy meglévő OpenUI5 / SAPUI5 alkalmazásba úgy, hogy:

- **Nincs build lépés** – minden runtime történik
- A UI5 beépített témái szolgálnak **bázisként** (`sap_horizon`, `sap_horizon_dark`, ...)
- CSS custom property-k felülírásával "patchelt" **virtuális téma**kat hozol létre
- Egyetlen JS hívással váltasz: `this._switchToVirtualTheme("alarm")`

Hogyan működik?



1. `Theming.setTheme(base)` – betölti a tényleges UI5 téma CSS-eket
2. `document.documentElement.style.setProperty(...)` – felülírja a szükséges CSS változókat a `:root`-on
3. A téma váltáskor előbb töröljük a korábbi patch-et, majd alkalmazzuk az újat

Integráció lépésről lépésre

1. lépés: Hozd létre a VirtualThemeManager-t

Készíts egy **önálló modult**, amit bármelyik controller/component használhat.

Fájl: `webapp/util/VirtualThemeManager.js`

```

sap.ui.define([
    "sap/ui/core/Theming"
], function (Theming) {
    "use strict";

    // =====
    // TÉMA REGISTRY - ezt szerkeszd a saját igényeidre!
    //
    // Struktúra:
    //   key: {
    //     base: "sap_horizon",           // valódi UI5 téma ID
    //     label: "Emberi olvasható név",
    //     patch: { ... }                // CSS változó felülírások
    //   }
    //   // patch: null = nincs módosítás
    // =====

    var THEMES = {
        normal: {
            base: "sap_horizon",
            label: "Normal",
            patch: null
        },
        warning: {
            base: "sap_horizon",
            label: "Warning",
            patch: {
                "--sapBackgroundColor": "#fff8e1",
                "--sapShell_Background": "#f9a825",
                "--sapPageHeader_Background": "#fff3cd",
                "--sapBrandColor": "#ff8f00",
                "--sapTile_Background": "#fff8e1",
                "--sapList_Background": "#ffdde7",
                "--sapButton_Background": "#fff3cd"
            }
        },
        alarm: {
            base: "sap_horizon_dark",
            label: "Alarm",
            patch: {
                "--sapBackgroundColor": "#1a0000",
                "--sapShell_Background": "#4a0e0e",
                "--sapPageHeader_Background": "#330000",
                "--sapBrandColor": "#ff1744",
                "--sapErrorColor": "#ff5252",
                "--sapTile_Background": "#2d0a0a",
                "--sapList_Background": "#1f0000",
                "--sapButton_Reject_Background": "#b71c1c"
            }
        },
        nightshift: {
            base: "sap_horizon_dark",
            label: "Night Shift",
            patch: null
        }
    };

    // =====
    // MANAGER SINGLETON
    // =====

    var _sActiveKey = null;
    var _oPendingPatch = null;
    var _fnOnApplied = null;

    var VirtualThemeManager = {

        /**
         * Kötelező: hívda meg egyszer az app indulásakor (pl. Component.init)
         * Feliratkozik a Theming "applied" eseményre.
         */
        init: function () {

```

```

        _fnOnApplied = this._onBaseThemeApplied.bind(this);
        Theming.attachApplied(_fnOnApplied);
    },

    /**
     * Virtuális téma váltás - EZ AZ EGYETLEN PUBLIKUS METÓDUS
     *
     * @param {string} sThemeKey - kulcs a THEMES registry-ből
     * @returns {boolean} sikeres volt-e
     */
    switchTo: function (sThemeKey) {
        var oThemeDef = THEMES[sThemeKey];
        if (!oThemeDef) {
            console.error("[VirtualTheme] Unknown theme:", sThemeKey);
            return false;
        }
        if (_sActiveKey === sThemeKey) {
            return true; // már aktív
        }

        // 1. Korábbi patch törlése
        this._removePatch();

        var sCurrentBase = Theming.getTheme();

        if (sCurrentBase !== oThemeDef.base) {
            // Bázis téma váltás kell → patch-et a queue-be tesszük
            _oPendingPatch = oThemeDef.patch;
            Theming.setTheme(oThemeDef.base);
        } else {
            // Ugyanaz a bázis → patch azonnal megy
            this._applyPatch(oThemeDef.patch);
        }

        _sActiveKey = sThemeKey;
        return true;
    },

    /**
     * Aktuális virtuális téma kulcsa
     * @returns {string|null}
     */
    getActiveKey: function () {
        return _sActiveKey;
    },

    /**
     * Aktuális virtuális téma definíció
     * @returns {object|null}
     */
    getActiveTheme: function () {
        return _sActiveKey ? THEMES[_sActiveKey] : null;
    },

    /**
     * Az összes regisztrált téma (pl. dropdown feltöltéshez)
     * @returns {Array<{key: string, label: string}>}
     */
    getThemeList: function () {
        return Object.keys(THEMES).map(function (sKey) {
            return { key: sKey, label: THEMES[sKey].label };
        });
    },

    /**
     * Új téma hozzáadása runtime-ban
     *
     * @param {string} sKey - egyedi kulcs
     * @param {object} oThemeDef - { base, label, patch }
     */
    register: function (sKey, oThemeDef) {
        THEMES[sKey] = oThemeDef;
    }
}

```

```

},
// ----- belső metódusok -----
_onBaseThemeApplied: function () {
    if (_oPendingPatch) {
        this._applyPatch(_oPendingPatch);
        _oPendingPatch = null;
    }
},
_applyPatch: function (oPatch) {
    if (!oPatch) { return; }
    var oRoot = document.documentElement;
    Object.keys(oPatch).forEach(function (sVar) {
        oRoot.style.setProperty(sVar, oPatch[sVar]);
    });
},
_removePatch: function () {
    document.documentElement.removeAttribute("style");
}
};

return VirtualThemeManager;
});

```

2. lépés: Inicializálás a Component-ben

Fájl: `webapp/Component.js`

```

sap.ui.define([
    "sap/ui/core/UIComponent",
    "your/app/namespace/util/VirtualThemeManager" // ← útvonal módosítás!
], function (UIComponent, VirtualThemeManager) {
    "use strict";

    return UIComponent.extend("your.app.namespace.Component", {
        metadata: { manifest: "json" },

        init: function () {
            UIComponent.prototype.init.apply(this, arguments);
            this.getRouter().initialize();

            // ★ Virtual theme rendszer indítása
            VirtualThemeManager.init();

            // Opcionális: kezdő téma beállítása
            VirtualThemeManager.switchTo("normal");
        }
    });
});

```

3. lépés: Használat bármelyik Controller-ben

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "your/app/namespace/util/VirtualThemeManager"
], function (Controller, VirtualThemeManager) {
    "use strict";

    return Controller.extend("your.app.namespace.controller.MyView", {

        // ----- Gombra kattintás -----
        onAlarmPressed: function () {
            VirtualThemeManager.switchTo("alarm");
        },

        onNormalPressed: function () {
            VirtualThemeManager.switchTo("normal");
        },

        // ----- WebSocket üzenet -----
        onWebSocketMessage: function (oMessage) {
            switch (oMessage.type) {
                case "ALARM":
                    VirtualThemeManager.switchTo("alarm");
                    break;
                case "WARNING":
                    VirtualThemeManager.switchTo("warning");
                    break;
                case "NIGHT_SHIFT_START":
                    VirtualThemeManager.switchTo("nightshift");
                    break;
                default:
                    VirtualThemeManager.switchTo("normal");
            }
        },

        // ----- Időzített váltás -----
        onInit: function () {
            this._checkShiftTheme();
            // 5 percenként ellenőrzés
            setInterval(this._checkShiftTheme.bind(this), 5 * 60 * 1000);
        },

        _checkShiftTheme: function () {
            var iHour = new Date().getHours();
            if (iHour >= 20 || iHour < 6) {
                VirtualThemeManager.switchTo("nightshift");
            } else {
                VirtualThemeManager.switchTo("normal");
            }
        });
    });
});
```

4. lépés: Dropdown kötése a View-ban (opcionális)

Ha UI-ból is szeretnéd választhatóvá tenni:

Controller (onInit):

```

var oModel = new sap.ui.model.json.JSONModel({
    themes: VirtualThemeManager.getThemeList(),
    activeKey: "normal"
});
this.getView().setModel(oModel, "vtheme");

```

View (XML):

```

<Select selectedKey="{vtheme>/activeKey}"
    change="onThemeChange"
    items="{vtheme>/themes}">
    <core:Item key="{vtheme>key}" text="{vtheme>label}" />
</Select>

```

Controller handler:

```

onThemeChange: function (oEvent) {
    var sKey = oEvent.getParameter("selectedItem").getKey();
    VirtualThemeManager.switchTo(sKey);
    this.getView().getModel("vtheme").setProperty("/activeKey", sKey);
}

```

Elérhető CSS változók

Ezek a legfontosabb CSS custom property-k, amiket a patch-ben használhatsz. minden UI5 kontroll ezekből épül:

Alap színek

Változó	Leírás
--sapBrandColor	Elsődleges márka szín (linkek, kijelölések)
--sapHighlightColor	Kiemelés szín
--sapBaseColor	Alap szín
--sapShellColor	Shell fejléc szín
--sapBackgroundColor	Általános háttérszín

Felület elemek

Változó	Leírás
--sapShell_Background	Shell háttér
--sapPageHeader_Background	Oldal fejléc háttér
--sapTile_Background	Tile/kártya háttér
--sapList_Background	Lista háttér
--sapGroup_TitleBackground	Csoport fejléc háttér
--sapContent_LabelColor	Label szöveg szín

Szöveg színek

Változó	Leírás
--sapTextColor	Általános szöveg
--sapTitleColor	Cím szöveg
--sapContent_LabelColor	Label szöveg
--sapLink_Color	Link szín

Gombok

Változó	Leírás
--sapButton_Background	Alapértelmezett gomb háttér
--sapButton_Emphasized_Background	Kiemelt gomb háttér
--sapButton_Reject_Background	Elutasító gomb háttér
--sapButton_Accept_Background	Elfogadó gomb háttér

Szemantikus színek

Változó	Leírás
--sapPositiveColor	Sikeres/pozitív (zöld)
--sapCriticalColor	Figyelmeztető (sárga/narancs)
--sapNegativeColor	Hibás/negatív (piros)
--sapInformativeColor	Informatív (kék)
--sapNeutralColor	Semleges (szürke)
--sapErrorHandler	Hibajelző szín

Teljes lista

A böngésző DevTools-ban (F12) megtalálod az összeset:

```
// Futtatsd a konzolban
getComputedStyle(document.documentElement)
```

Vagy: `Ctrl+Shift+I` → Elements → Computed → szűrj "--sap"-ra.

Elérhető bázis téma

Téma ID	Leírás
sap_horizon	Horizon (világos) – ajánlott
sap_horizon_dark	Horizon (sötét)
sap_horizon_hcb	Horizon (high contrast fekete)
sap_horizon_hw	Horizon (high contrast fehér)
sap_fiori_3	Fiori 3 (világos)
sap_fiori_3_dark	Fiori 3 (sötét)
sap_fiori_3_hcb	Fiori 3 (high contrast fekete)
sap_fiori_3_hw	Fiori 3 (high contrast fehér)

Új téma hozzáadása

Statikusan (a THEMES objektumban)

```
var THEMES = {
    // ... meglévő téma...
    maintenance: {
        base: "sap_horizon",
        label: "Karbantartás mód",
        patch: {
            "--sapBackgroundColor": "#e3f2fd",
            "--sapBrandColor": "#1565c0",
            "--sapShell_Background": "#bbdefb"
        }
    }
};
```

Dinamikusan (runtime-ban)

```
VirtualThemeManager.register("custom_event", {
    base: "sap_horizon_dark",
    label: "Egyedi esemény",
    patch: {
        "--sapBackgroundColor": "#1a0033",
        "--sapBrandColor": "#aa00ff"
    }
});

// Ezután használható:
VirtualThemeManager.switchTo("custom_event");
```

Fontos tudnivalók és korlátok

Működési elv

- A `Theming.setTheme()` betölti a bázis téma **összes** library CSS-ét (sap.m, sap.ui.core, sap.f, stb.) – ez aszinkron, pár száz ms
- A CSS patch **szinkron** – azonnal hat, mert a `:root` inline style magasabb specificitású, mint a téma CSS custom property definíciói
- Témaváltáskor a pending patch mechanizmus biztosítja, hogy a patch csak a bázis CSS betöltése **után** kerül alkalmazásra

Korlátok

1. A `Theming.getTheme()` a **bázis témat adja vissza**, nem a virtuális téma kulcsát. Használ a `VirtualThemeManager.getActiveKey()`-t helyette.
2. **A `:root` inline style törlése:** A `_removePatch()` törli az **összes** inline style-t a `<html>` elemről. Ha más is ír oda inline style-t, az is törlődik. Szükség esetén módosítsd célzott `removeProperty()`-re.
3. **Nem minden CSS property felülírható:** Néhány UI5 kontroll belső CSS-ét közvetlenül (nem CSS változón keresztül) definiálja. Ezeket a patch nem fogja elérni. A legtöbb kontroll (`sap.m.*`, `sap.f.*`) viszont teljesen CSS változó alapú a Horizon témaban.
4. **Theming.attachApplied():** Ha a te kódod is használja ezt az eseményt más céllra, figyelj arra, hogy a `VirtualThemeManager` is feliratkozik rá. Nincs konfliktus, de a sorrend nem garantált.

Ajánlások

- A `sap_horizon` és `sap_horizon_dark` témakat használ bázisnak – ezek a legújabbak és a legjobban támogatják a CSS változókat
- Teszteld a patch-eket az összes használt UI5 kontrollal
- A DevTools-ban (`document.documentElement.style`) ellenőrizheted mely változók vannak éppen felülírva