Otázky k integraci konfiguratoru BLOCKids s WooCommerce

✅ CO UŽ VÍME ZE ZDROJOVÝCH KÓDŮ

Technologie konfiguratoru:

- **Framework**: Next.js 14.2.3 + React 18 + TypeScript
- **Session management**: iron-session (server-side cookies)
- **State management**: Redux Toolkit
- **Jazyky**: cs, en, de (přes i18next)
- **Styling**: SCSS moduly

API Struktura (z analýzy kódu):

- **Base URL**: `https://phpstack-851454-4637763.cloudwaysapps.com/api/v1`
- **Token**: Posílá se v URL path, NE v headerech: `/{endpoint}/{lang}/{token}`
- **Content-Type**: `application/ld+json`
- **Jazykové mutace**: `/{lang}/` v URL path (cs, en, de)

Kompletní seznam endpointů (ze zdrojáků):

GET endpointy:

- ✅ `GET /customers/me/{token}` - validace tokenu + info o uživateli
- ✅ `GET /desks/{lang}` - seznam desek/panelů
- ✅ `GET /grips/{lang}` - seznam gripů/chytů
- ✅ `GET /mattresses/{lang}` - seznam matrací
- ✅ `GET /photos/{lang}` - inspirační fotogalerie

- ✅ `GET /faq-items/{lang}` - FAQ položky
- ✅ `GET /plans/detail/{lang}/{token}/{hash}` - detail konkrétního plánu

POST endpointy:

- ✅ `POST /plans/create/{lang}/{token}` - vytvoření nového plánu (body: `{location: "outdoor"/"indoor"}`)
- ✅ `POST /plans/update/{lang}/{token}/{hash}` - aktualizace plánu
- ✅ `POST /plans/confirm/{lang}/{token}/{hash}` - **potvrzení plánu = přidání do košíku**
- ✅ `POST /plans/clone/{lang}/{token}/{hash}` - klonování plánu
- ✅ `POST /plans/change-title/{lang}/{token}/{hash}` - změna názvu plánu (body: `{title: "..."}`)

DELETE endpointy:

- ✅ `DELETE /plans/delete/{lang}/{token}/{hash}` - smazání plánu

Autentizační flow (z analýzy kódu):

1. ✅ WooCommerce vygeneruje JWT token
2. ✅ Přesměruje na konfigurátor: `/{locale}/sso?token={JWT_TOKEN}`
3. ✅ Konfigurátor zavolá `GET /customers/me/{token}` pro validaci
4. ✅ Uloží do iron-session: userId, fullName, token, customerTypeId, accessHash
5. ✅ Přesměruje na `/detail/{accessHash}` (existující plán nebo "new")

Workflow potvrzení plánu (z analýzy kódu):

1. ✅ Uživatel dokončí konfiguraci v konfiguratoru
2. ✅ Klikne na "Přidat do košíku"
3. ✅ Zavolá se `POST /plans/update/{lang}/{token}/{hash}` (uložení)

4. ✅ Zavolá se `POST /plans/confirm/{lang}/{token}/{hash}` (potvrzení)
5. ✅ Přesměruje zpět na eshop: `NEXT_PUBLIC_URL_REDIRECT_PATH_{locale}`
6. ⚠️ **OTÁZKA**: Jak předat `accessHash` plánu zpět do eshopu pro přidání do košíku?

❓ CO JEŠTĚ POTŘEBUJEME VĚDĚT

1.1 API - Produkční prostředí

- [ ] Jaká bude **produkční base URL** API? (teď: `phpstack-851454-4637763.cloudwaysapps.com`)
- [ ] Existuje **testovací/sandbox API** pro vývoj?
- [ ] Jaké jsou **rate limity**?
- [ ] Máte **API dokumentaci** nad rámec toho, co je v kódu?

1.2 API Response formát

Ze zdrojáků vidím, že API vrací:

```json
{
  "status": 200,
  "data": { ... }
}
```

- [ ] Je to standardní formát pro všechny endpointy?
- [ ] Jak vypadají **error responses**? (kódy, struktura)

---

2. AUTENTIZACE A JWT - ✅ FLOW UŽ ZNÁME

Co víme ze zdrojových kódů:

✅ **Validace tokenu**: `GET /customers/me/{token}` vrací:

```
json

{
  "data": {
    "id": 123,
    "givenName": "Jan",
    "familyName": "Novák",
    "segment": { "id": 1 },
    "planInProgress": "abc123hash" nebo null,
    "plans": [...]
  }
}
```

✅ **SSO flow**: Přesměrování na `/{locale}/sso?token={JWT_TOKEN}`

2.1 JWT Token - co potřebujeme

☐ **Jak vygenerujeme JWT token** ve WooCommerce?

- Potřebujeme **shared secret**?
- Nebo zavoláme váš endpoint pro generování?
- Pokud shared secret - **jaký je algoritmus**? (HS256, RS256?)

☐ **Jaká je struktura JWT payload**?

```
json
```

```
{
  "user_id": 123,
  "email": "user@example.com",
  "exp": 1234567890,
  ... co dalšího?
}
```

- [ ] **Jak dlouho je token platný**? (expiration time)
  - Potřebujeme **refresh token** mechanismus?

2.2 Uživatelská synchronizace

- [ ] **Musíme vytvářet uživatele** ve vašem systému před přihlášením?
  - Nebo nás `GET /customers/me/{token}` automaticky vytvoří?
- [ ] Jak mapujete **customerTypeId** (segment)?
  - WooCommerce: customer, subscriber, shop_manager...
  - Váš systém: "Rodina" (id=1?), "Veřejný prostor" (id=2?)
- [ ] Potřebujeme endpoint pro **registraci nového zákazníka**?
  - `POST /customers/register` nebo podobně?

---

3. HOSTING A INTEGRACE KONFIGURATORU

Co víme:

- ✅ Konfigurátor je **Next.js 14 aplikace**
- ✅ Poběží na **subdoméně klienta** (např. `konfigurator.blockids.cz`)
- ✅ Používá **iron-session** (server-side cookies) - vyžaduje běh na stejné (sub)doméně
- ✅ Po potvrzení přesměruje na: `NEXT_PUBLIC_URL_REDIRECT_PATH_{locale}`

3.1 Hosting konfiguratoru

- [ ] **Kdo bude hostovat** konfigurátor?
  - My na subdoméně klienta?
  - Vy na své infrastruktuře?
- [ ] Jaké jsou **požadavky na server**?
  - Node.js verze?
  - RAM, CPU?
  - Je to **staticky exportovatelné** nebo potřebuje Node.js runtime?

3.2 Konfigurace pro nový eshop

Budeme potřebovat upravit tyto `.env` proměnné:

```bash
# API - produkční URL
API_BASE_PATH="https://???"
API_BASE_VERSION="v1"

# Session secret - nový pro produkci
SESSION_SEAL_PASSWORD="???"

# Redirect po potvrzení plánu - zpět do eshopu
NEXT_PUBLIC_URL_REDIRECT_PATH_CS="https://klient.cz/cs/kosik?plan={accessHash}"
NEXT_PUBLIC_URL_REDIRECT_PATH_EN="https://klient.cz/en/cart?plan={accessHash}"
NEXT_PUBLIC_URL_REDIRECT_PATH_DE="https://klient.cz/de/warenkorb?plan={accessHash}"
```

Otázky:

- [ ] Jak vygenerujeme nový **SESSION_SEAL_PASSWORD** pro produkci?

- [ ] Jaký formát má **accessHash** v redirect URL?
  - Query parameter `?plan=abc123`?
  - Hash `#abc123`?
  - Path `/kosik/abc123`?

3.3 Cookie a session management

- [ ] Konfigurátor používá cookie `blockids-config` (iron-session)
- [ ] Cookie musí být **sdílená mezi subdoménou a hlavní doménou**
- [ ] Nastavení: `domain: ".klient.cz"` aby fungovalo i na `konfigurator.klient.cz`
- [ ] Je to v pořádku, nebo potřebujete jiné řešení?

---

4. BUSINESS LOGIKA A PŘIDÁNÍ DO KOŠÍKU - ⚠️ KLÍČOVÁ SEKCE

4.1 Workflow ✅ co víme + ❓ co potřebujeme

Známý flow ze zdrojáků:

1. ✅ Zákazník se přihlásí v eshopu (WooCommerce)
2. ✅ Eshop vygeneruje JWT token
3. ✅ Přesměruje na: `konfigurator.klient.cz/cs/sso?token={JWT}`
4. ✅ Konfigurátor validuje token přes API: `GET /customers/me/{token}`
5. ✅ Konfigurátor vytvoří plán nebo otevře existující
6. ✅ Zákazník navrhne lezeckou stěnu
7. ✅ Klikne "Přidat do košíku"
8. ✅ Konfigurátor zavolá:
   - `POST /plans/update/{lang}/{token}/{hash}` (uloží změny)

- `POST /plans/confirm/{lang}/{token}/{hash}` (potvrdí = "přidej do košíku")

9. ✅ Přesměruje zpět na: `NEXT_PUBLIC_URL_REDIRECT_PATH`

❓ CO NENÍ JASNÉ:

☐ **Jak předat accessHash zpět do eshopu?**

- Query param: `https://klient.cz/kosik?plan=abc123hash`?
- Cookie (iron-session)?
- Jinak?

☐ **Co eshop udělá s accessHash?**

- Zavolá `GET /plans/detail/{lang}/{token}/{hash}` pro získání dat?
- Nebo je celý plán uložený už na vašem API a eshop jen vytvoří WC produkt s odkazem?

☐ **Jak se plán zobrazí v košíku WooCommerce?**

- Jako jeden produkt "Vlastní lezecká stěna" + metadata (rozměry, gripy, matrace)?
- Jako více produktů (každý grip zvlášť)?
- Jako variabilní produkt?

☐ **Uloží se plán do WC objednávky?**

- Celý JSON plánu?
- Jen accessHash a link na detail?
- Screenshot/obrázek návrhu?

4.2 Cenová kalkulace - ❗ KRITICKÉ

Ze zdrojáků vidím:

```typescript
```

```
{
  grip: number,          // ID gripu
  gripQuantity: number,  // Množství gripů
  mattress: number,      // ID matrace
  mattressQuantity: number, // Množství matrací
  calculatedWidth: number,  // Šířka v ???
  calculatedHeight: number  // Výška v ???
}
```

Otázky:

- [ ] **Ceny gripů a matrací** jsou v `GET /grips/{lang}` a `GET /mattresses/{lang}`?
  - Jsou ceny v Kč (CZK)?
  - Jsou s DPH nebo bez?
- [ ] **Jak se počítá cena za desky/panely?**
  - Cena za m²?
  - Fixní cena podle velikosti?
  - Endpoint pro výpočet: `POST /plans/calculate-price`?
- [ ] **Cena montáže/instalace?**
  - Je v ceně nebo extra?
  - Jak se určuje?
- [ ] **Slevy, kupóny?**
  - Řeší se v eshopu nebo v konfiguratoru?
- [ ] **Doporučený postup:**
  - Eshop má své produkty (gripy, matrace, desky) s cenami?
  - Nebo stahuje ceny z vašeho API?

4.3 Stock/Dostupnost

- [ ] Má API informaci o **skladové dostupnosti**?
- [ ] Musíme kontrolovat stock před přidáním do košíku?
- [ ] Co když není skladem během objednávání?

---

5. TECHNICKÉ DETAILY - DATOVÉ STRUKTURY

5.1 Workspace - ✅ už rozumíme

Ze zdrojových kódů víme:

```typescript
workspace: {
  A1: { id: 123, rotation: 0 },  // ID gripu + rotace (0/90/180/270)
  A2: { id: 456, rotation: 90 },
  B1: { id: 123, rotation: 180 },
  ...
  F2: { id: 789, rotation: 270 }
}
```

- ✅ Grid je **6 sloupců (A-F) x 2 řady (1-2)** = 12 pozic
- ✅ Každá pozice: ID gripu + rotace ve stupních
- ✅ Horizontální vs Vertical orientace mění rozložení gridu

Doplňující otázky:

- [ ] Jak se **výpočítává calculatedWidth/Height**?
  - Automaticky podle počtu obsazených pozic v gridu?
- [ ] **customWidth/Height** - kdy se používá?

- Může zákazník zadat vlastní rozměry mimo grid?

☐ V jakých **jednotkách** jsou rozměry? (cm, mm, palce?)

5.2 Gripy (Holds/Desks)

Ze zdrojových kódů:

```typescript
{
  id: number,
  title: string,
  price: number,        // Cena v CZK?
  currency: "czk",
  order: number,
  image: string,        // URL obrázku
  overlays: [           // Různé varianty vzhledu
    {
      id: number,
      type: "rectangle" | "triangle_top" | "triangle_left",
      orientation: "horizontal" | "vertical",
      rotation: 0 | 90 | 180 | 270,
      inputs: boolean,   // Co to znamená?
      image: string
    }
  ],
  type: "rectangle" | "table" | "triangle_left"  // V plan-detail
}
```

Otázky:

☐ Co jsou **"overlays"**?

- Různé barevné varianty stejného gripu?

- Nebo různé grafické prvky na gripu?

- [ ] Co znamená **"inputs"** u overlay?
- [ ] Je `type` u gripu totéž co u overlay, nebo má jiný význam?
- [ ] **"Desks"** vs "Grips"** - je to totéž?
  - V endpoints jsou: `/desks/{lang}` i `/grips/{lang}`
  - Jaký je rozdíl?

5.3 Matrace

- [ ] Endpoint `/mattresses/{lang}` vrací jakou strukturu?
  - Podobnou jako gripy? (id, title, price, image...)
- [ ] **mattressQuantity** - počítá se automaticky podle velikosti plánu?
- [ ] Může zákazník vybrat **více typů** matrací v jednom plánu?

5.4 Location type

Ze zdrojáků:

```typescript
location: "outdoor" | "indoor"
```

- [ ] Ovlivňuje to **dostupné gripy/matrace**?
- [ ] Nebo jen **cenu** (outdoor dražší)?
- [ ] Zobrazuje se v eshopu jako **kategorie produktu**?

---

6. TESTOVÁNÍ A DEPLOYMENT

6.1 Development prostředí

Co budeme potřebovat pro vývoj:

☐ **DEV/Sandbox API**

- Testovací API URL (oddělené od produkce)
- Testovací JWT tokeny
- Testovací účty zákazníků

☐ **Testovací konfigurátor**

- Můžeme mít DEV instanci konfiguratoru?
- Nebo testujeme na produkčním?

6.2 Produkční deployment

Naše plánovaná infrastruktura:

```
DEV doména:     dev.klient.cz
                konfigurator.dev.klient.cz

PROD doména:    klient.cz
                konfigurator.klient.cz
```

Otázky:

☐ Můžeme mít **2 instance** konfiguratoru (DEV + PROD)?

☐ Potřebujeme **oddělit API keys/secrets** pro DEV vs PROD?

☐ **Migrace dat** z DEV na PROD - jak to řešíte?

- Plány zákazníků se přenesou?

- Nebo začínáme na PROD s prázdnou DB?

6.3 Session a cookies na více doménách

Potenciální problém:

- Eshop: `klient.cz`
- Konfigurátor: `konfigurator.klient.cz`
- iron-session cookie: `blockids-config`

Řešení:

- Cookie domain: `.klient.cz` (sdílená)
- SameSite: `Lax`
- Secure: `true` (jen HTTPS)

Ověření:

- [ ] Je tento přístup v pořádku pro vaši architekturu?
- [ ] Nebo preferujete jiné řešení (token v URL, separate storage...)?

---

7. TIMELINE A PRIORITIES

7.1 Harmonogram

- [ ] **Kdy budete mít připravené:**
  - Finální API dokumentaci?
  - Testovací prostředí?
  - Produkční API credentials?
- [ ] **Deadline** pro spuštění integrace?

☐ **Plánujete změny v API** v dohledné době?

- Breaking changes?
- Nové endpointy?

7.2 Naše odhady (po analýze zdrojáků)

Fáze 1: WordPress Plugin (WC integrace)

- Generování JWT tokenů: **1 den**
- Přesměrování do konfiguratoru: **0.5 dne**
- Příjem dat zpět z konfiguratoru: **1 den**
- Vytvoření/aktualizace WC produktu: **1-2 dny**
- **Celkem: 3.5-4.5 dne**

Fáze 2: Konfigurátor deployment

- Setup na subdoméně: **0.5 dne**
- Konfigurace .env (API URLs, redirects): **0.5 dne**
- Testování session/cookies: **0.5 dne**
- **Celkem: 1.5 dne**

Fáze 3: Testování & debugging

- End-to-end testing: **1-2 dny**
- Opravy bugů: **1-2 dny**
- **Celkem: 2-4 dny**

📊 **CELKOVÝ ODHAD: 7-10 pracovních dnů**

(Závisí na odpovědích níže - pokud budeme potřebovat vlastní autentizační systém nebo složitější integrace, může to být více)

⭐ TOP 10 PRIORITNÍCH OTÁZEK - ODPOVĚZTE PROSÍM JAKO PRVNÍ

Bez odpovědí na tyto otázky nemůžeme začít s implementací:

🔴 KRITICKÉ (nutné hned):

1. **JWT generování** - Jak vygenerujeme JWT tokeny ve WooCommerce?
   - Shared secret + algoritmus?
   - Nebo váš endpoint pro generování?
2. **AccessHash flow** - Jak předat `accessHash` zpět do eshopu po potvrzení plánu?
   - Query param `?plan=abc123`?
   - Cookie?
   - Jinak?
3. **Cenová kalkulace** - Jak spočítat celkovou cenu plánu?
   - Ceny jsou v API responses?
   - Existuje endpoint `/plans/calculate-price`?
   - Jak se počítá cena desek/panelů?
4. **API prostředí** - Jaké budou produkční/testovací URL?
   - DEV API URL?
   - PROD API URL?
   - Rate limity?
5. **Konfigurátor hosting** - Kdo bude hostovat konfigurátor?

- My na subdoméně klienta?
- Vy na své infrastruktuře?

🟡 DŮLEŽITÉ (potřebujeme brzy):

6. **WC Product mapping** - Jak uložit plán do WooCommerce objednávky?
   - Jeden produkt + metadata?
   - Více produktů (gripy, matrace zvlášť)?
   - Celý JSON plánu nebo jen accessHash?
7. **Session management** - Je OK shared cookie `.klient.cz`?
   - Nebo preferujete jiné řešení?
8. **Uživatelská synchronizace** - Musíme vytvářet účty ve vašem systému?
   - Automaticky při registraci v WC?
   - Nebo endpoint `/customers/register`?
9. **Error handling** - Jak vypadají error responses z API?
   - HTTP status codes?
   - JSON struktura errors?
10. **Timeline** - Kdy budete mít připravené testovací prostředí a dokumentaci?

📧 KONTAKT PRO ODPOVĚDI

Prosím pošlete odpovědi na: **[váš email]**

Nebo zavolejte call pro probrání technických detailů: **[váš telefon]**

Dokument připravil: Aleš
Datum: 10. 2. 2026
Verze: 2.0 (aktualizováno po analýze zdrojových kódů konfiguratoru)