Mobile App Integration Guide: Subscriptions & Payment

Overview

The Biltix application follows a **"Payment First" Registration Flow**. A user account is ONLY created after a successful payment for a subscription plan. This guide details the API endpoints and logic required to implement this in the mobile app.

---

1. Registration Flow (Payment First)

Step 1: Fetch Plans

Display available subscription plans to the user.

- **Endpoint:** `GET /api/v1/subscriptions/plans`
- **Auth:** Public (No token required)
- **Response:** List of plans with `id`, `name`, `price`, `currency`, `features`.

Step 2: User Data Collection & Payment Initialization

Collect user details (Name, Email, Phone, Company, Password) AND the selected `plan_id`. Instead of calling a signup API, call the payment initialization API.

- **Endpoint:** `POST /api/v1/payment/init_registration`
- **Auth:** Public
- **Parameters:**

* `name`, `email`, `phone`, `password`, `company_name`, `role` (e.g., 'contractor'), `device_type` ('A' for Android, 'I' for iOS). * `plan_id` (Integer, ID of selected plan). * `redirect_url` (String, URL scheme to return to app, e.g., `biltixapp://payment/callback`).

- **Response (Success - 200):**

`json { "amount": 5000, "currency": "SAR", "description": "Subscription for Gold Plan", "payment_url": "https://api.moyasar.com/forms/...", "token": "reg_123456..." // Temporarily stores user data } `

Step 3: Process Payment (Moyasar)

Use the `payment_url` from Step 2.

- **Method A (WebView):** Open `payment_url` in a WebView. It handles 3DSecure. On success, Moyasar redirects to your `redirect_url`.
- **Method B (SDK):** If using Moyasar Mobile SDK, use the `amount` and `currency`. *Note: Validation backend expects a valid Moyasar Payment ID.*

Step 4: Complete Registration

After successful payment (or redirect), call the completion API to finalize account creation.

- **Endpoint:** `POST /api/v1/payment/complete_registration`
- **Auth:** Public
- **Parameters:**

* `id`: The Payment ID received from Moyasar (e.g., in the redirect URL query params as `id`). * `token`: The `token` received in Step 2 response (also passed back in redirect). * `status`: 'paid' (optional, validation happens on server).

- **Response (Success - 200):**

`json { "user": { "id": 1, "name": "...", "email": "..." }, "token": "auth_token_xyz...", // Login Token "message":

"Registration Successful" } `

- **Action:** Save the `token` and log the user in.

---

2. Subscription Management & access

Check Subscription Status

The user's subscription status is included in the Profile API.

- **Endpoint:** `POST /api/v1/get_user_profile`
- **Auth:** Bearer Token
- **Response:**

`json { "subscription": { "plan_name": "Gold Plan", "status": "active", // or 'expired', 'cancelled' "amount_paid": 5000, "currency": "SAR", "expires_at": "2026-05-20", "days_remaining": 360, "transaction_id": "pay_xyz..." } } `

Handling Access Limits (Critical)

If a user tries to access a feature not allowed by their plan (or if expired), the API returns a **403 Forbidden** error.

- **HTTP Code:** `403`
- **JSON Body:**

`json { "code": 403, "message": "Access denied. Subscription plan limit or expired.", "data": {} } `

- **App Logic:** If you receive this error, show a "Upgrade Required" or "Subscription Expired" popup to the user.

---

3. Handling Interruptions (Background Webhook)

If the app crashes or user closes it during payment:

- The backend listens to Moyasar Webhooks.
- If payment succeeds, the account is **auto-created** on the server.
- User can try logging in with their email/password.

4. Polling Status (Optional)

If you want to check status while user is on payment screen:

- **Endpoint:** `GET /api/v1/payment/status/{token}`
- Returns: `pending`, `completed`, or `failed`.

Technical Summary

1. **Init:** `POST /payment/init_registration` -> Get URL & Token. 2. **Pay:** WebView -> `payment_url`. 3. **Verify:** `POST /payment/complete_registration` -> Get User & Auth Token. 4. **Usage:** Handle `403` for plan limits.