

Trust-as-a-Service & Escrow Platform API

Comprehensive RESTful API Documentation, Architectural Source Code Review, Financial Ledger Mechanics & Integration Guidance

74 DOCUMENTED ENDPOINTS	13 FUNCTIONAL MODULES	100% ENDPOINT COVERAGE	SAR NATIVE CURRENCY
---	---	--	---

Platform: DAF3 Enterprise Engine (Laravel 11 / PHP 8.2+)
Security: Dual-Ledger Double-Entry & Hardware License Protection

Document Version: 3.8.0 Enterprise Release
Generated: October 2026 | Riyadh, Saudi Arabia

1. Executive Summary & Source Code Engineering Review

This section presents a comprehensive technical audit of the **DAF3 (دفع)** codebase. The review assesses architecture, data persistence models, financial transaction integrity, API design patterns, security postures, and scalability.

Platform Mission & Scope

DAF3 is a specialized Saudi Arabian (SAR currency) **Trust-as-a-Service (TaaS)** and Milestone-based Escrow Fintech platform. It bridges trust between buyers and sellers across digital services, hardware delivery, contracting, cafeteria campus ecosystems, and desktop software licensing.

1.1 Codebase Structure & Technology Stack

Layer / Component	Current Implementation	Architectural Assessment
Core Framework	Laravel 11.x on PHP 8.2+	Modern foundation utilizing service providers, facade architecture, and Eloquent ORM. Excellent LTS support and performance.
Database & ORM	MySQL 8.0+ / MariaDB via Eloquent	Normalized schema across 42 database models. Foreign key associations and indexed relations for transactions, listings, and milestones.
Payment Gateways	HyperPay (REST Checkouts API)	Supports Saudi Mada, Visa, and Mastercard with asynchronous webhook/status callback settlement into local wallet accounts.
Export Engines	Barryvdh DomPDF & Maatwebsite Excel	Native generation of PDF and XLSX invoices, financial statements, and milestone audit sheets.
In-App Communication	Database-driven real-time chat & notifications	Synchronous message polling, online presence management, read/seen receipts, and multi-tenant notification counting.
Desktop Licensing	Hardware-bound UUID / Machine Hash validation	Secures on-premise desktop POS and client applications with hardware seat limits and dynamic activation key validation.

1.2 Architectural Strengths

- Dual-Ledger Accounting:** The system maintains strict segregation between financial double-entry bookkeeping (`finance_sheets` table with credit/debit/commission breakdown) and user-facing transaction histories (`transactions` table with audit types 1, 2, 4, 5).
- Milestone State Machine:** Escrow funds cannot be released unilaterally. Release flows require mutual state synchronization (Milestone creation → Acceptance → Work in progress → Full or partial release → Seller wallet credit).
- Tiered Subscription Quotas:** Built-in subscription controls (`userServices::checkSubscriptionLimit`) calculate projected transaction values against monthly spend caps, enforcing plan upgrades when limits are approached.
- Hardware-Locked Desktop Licensing:** Desktop licenses evaluate machine fingerprints before registration, preventing unauthorized duplicate activations across multiple terminals.
- Dynamic Invoice Generation with SVG QR Codes:** Automated SVG QR code generation pointing directly to authenticated invoice payment pages simplifies mobile and web invoice fulfillment.

1.3 Technical Observations & Recommended Roadmap

⚠️ Key Architecture Recommendations for Production Scale

1. **Token-Based Authentication:** Migrate endpoints that currently accept `user_id` in the request body to authenticated sessions using **Laravel Sanctum Bearer Tokens** (`Authorization: Bearer <token>`). This prevents user impersonation on mobile and REST clients.
2. **Transactional Atomicity (ACID):** Wrap wallet mutations, milestone status updates, and ledger inserts within `DB::transaction(function() { ... })` blocks to prevent partial balance updates during network or database dropouts.
3. **Idempotency Keys:** Introduce Idempotency-Key HTTP headers on critical financial routes (`/wallet-to-wallet-transfer`, `/withdraw`, `/deposit`) to guarantee that retried requests do not result in double deductions.
4. **Response Envelope Standardization:** Standardize JSON outputs to follow a single schema across all 74 endpoints (e.g. `{"success": true, "code": 200, "message": "...", "data": { ... } }`).

1.4 Escrow & Milestone Financial State Machine

[Listing Created (Status 0: Pending)]

↓ Buyer accepts link via `/api/acceptedBuyerListing`

[Listing Active (Status 1: Active)]

↓ Milestone proposed via `/api/milestone/store` (Status 0: Pending)

↓ Seller accepts milestone via `/api/milestone/{id}/status` (Status 1: In Progress)

- **Option A: Full Release** → `/api/milestone/release` → Status 3: Released (Funds → Seller Wallet)

- **Option B: Partial Release** → `/api/milestone/partial-release` → Released amount to seller; new

Milestone created for remainder

- **Option C: Cancellation / Dispute** → `/api/milestones/cancel` → Status 4: Cancelled (Funds refunded or held for dispute)

↓ Final delivery complete via `/api/listing/markAsComplete`

[Listing Completed (Status 3: Finished)]

2. Core Concepts, Conventions & Integration Guidelines

This section outlines system conventions, base endpoints, protocol headers, financial rounding rules, and error handling behaviors implemented across the DAF3 API ecosystem.

2.1 Environment Base URLs

Environment	Base URI	Description
Local Development	http://127.0.0.1:8000/api	Active Artisan serve local environment for development and debugging.
Staging / UAT	https://staging.daf3.com/api	Sandbox environment connected to HyperPay test entity gateways.
Production	https://daf3.com/api	Live production server with active Saudi payment settlement.

2.2 Standard Request Headers

Header Key	Expected Value	Required For
Accept	application/json	All requests to enforce JSON responses.
Content-Type	application/json or multipart/form-data	Required on all POST requests containing body payloads or binary files.
Authorization	Bearer {api_token}	Required on protected routes where Sanctum token verification is enforced.

2.3 Global HTTP Response & Error Codes

HTTP Code	Status Label	Meaning & Scenario
200 OK	Success	Request completed successfully. Data payload returned in body.
201 CREATED	Resource Created	Entity created successfully (e.g. Invoice generated, User registered).
400 BAD REQUEST	Business Logic Error	Invalid transaction attempt (e.g. Canceling already cancelled listing, duplicate acceptance).
401 UNAUTHORIZED	Unauthorized	Unauthenticated attempt to access chat room or listing not yet accepted.
403 FORBIDDEN	Quota / Limit Exceeded	Monthly subscription spend limit reached. User must upgrade package via pricing.
404 NOT FOUND	Resource Not Found	Requested user, milestone, listing, invoice, or bank account record does not exist.
422 UNPROCESSABLE	Validation Failure	Input validation failed (missing required parameter, invalid email, password mismatch).

HTTP Code	Status Label	Meaning & Scenario
500 SERVER ERROR	Internal Error	Unhandled exception, database query error, or external gateway failure.

2.4 Financial Rounding & Commission Model

- **Currency:** All monetary values are processed in **Saudi Riyal (SAR)** and must be passed as numeric decimals with up to 2 decimal places (e.g. 150.00).
- **Commission-Based Subscriptions:** If the seller is subscribed to a commission-based package (`userServices::isCommissionBasedPackage($sellerId) == true`), the platform deducts a standard **3.5% commission fee** (`round($amount * 0.035, 2)`) upon milestone release.
- **Fixed Tiers:** Sellers on flat-fee subscription tiers incur **0% transaction commission** on released milestones.

2.5 Storage Paths & Binary Asset Management

Asset Category	Storage Path	Supported Formats
User Profile Avatars	public/profileImages/	JPEG, PNG, WEBP (Max 4MB)
Listing Product Images	public/seller/listings/images/	JPEG, PNG, WEBP (Multiple allowed)
Listing Video Walkthroughs	public/seller/listings/videos/	MP4, MOV, WEBM (Max 50MB)
Listing PDF / Contracts	public/seller/listings/documents/	PDF, DOCX, ZIP
Chat Attachments	public/chatuploads/	JPEG, PNG, PDF
Manual Top-up Slips	public/transferrSlips/	JPEG, PNG, PDF
Generated Invoices & QR	public/qrcodes/ & storage/app/public/exports/	SVG (QR codes), PDF, XLSX

Table of Contents

Complete directory of all 74 API endpoints across 13 core architectural domains.

1. Authentication & Account Management

POST	/api/register	#1
POST	/api/login	#2
POST	/api/google-login	#3
POST	/api/forget-password	#4
POST	/api/update-password/{id}	#5
POST	/api/delete-account	#6

2. User Profiles & Geography / Metadata

GET	/api/get-user/{id}	#7
POST	/api/update-profile/{id}	#8
GET	/api/get-countries	#9
GET	/api/get-cities/{id}	#10

3. Trust-as-a-Service (TaaS) Listings & Deals

POST	/api/create-listing/{id}	#11
GET	/api/get-user-listing/{id}	#12
POST	/api/share-link	#13
GET	/api/buyer/{id}	#14
GET	/api/seller/{id}	#15
POST	/api/acceptedBuyerListing	#16
POST	/api/listing/markAsComplete	#17
POST	/api/cancel-listing	#18

4. Milestone & Escrow Management

POST	/api/milestone/store	#19
GET	/api/buyer/milestones/{listing_id}	#20
POST	/api/milestone/release	#21
POST	/api/milestone/partial-release	#22
POST	/api/milestone/{id}/status	#23
POST	/api/milestones/cancel	#24
GET	/api/milestone/{milestone_id}	#25

5. Digital Wallet, Ledger & Banking

GET	/api/transactions/{user_Id}	#26
GET	/api/wallet-receipt/{user_id}	#27
POST	/api/bank-details	#28
POST	/api/user-wallet-update	#29
GET	/api/get-bank-details	#30
POST	/api/wallet-to-wallet-transfer	#31
GET	/api/withdraw/{user_id}	#32
POST	/api/withdraw	#33
POST	/api/deposit	#34

6. HyperPay Payment Gateway Integration

POST	/api/payment/hyperpay	#35
GET	/api/payment/hyperpay/status/{checkoutId}/{user_id}	#36

7. Invoicing & Digital Billing

POST	/api/store-invoice	#37
POST	/api/get-invoices	#38
GET	/api/invoice/{id}	#39
GET	/api/delete/{id}	#40
POST	/api/export-invoices-pdf	#41
POST	/api/export-invoices-excel	#42
POST	/api/export-milestones-pdf	#43
POST	/api/export-milestones-excel	#44

8. Manual Top-Up Requests

POST	/api/get-manual-top-ups	#45
POST	/api/store-manual-top-up	#46

9. In-App Chat, Messaging & Online Presence

GET	/api/chats/{chat_id}/messages	#47
GET	/api/chats/{user_id}	#48
POST	/api/chat-listing	#49
POST	/api/send-message	#50
POST	/api/seen-message	#51
POST	/api/seen-general-message	#52
POST	/api/update-notification	#53
POST	/api/general-update-notification	#54
GET	/api/notifications/latest-count	#55
GET	/api/notifications/general-latest-count	#56
POST	/api/enter-chat	#57
POST	/api/leave-chat	#58

10. Packages, Subscriptions & Multi-User Escrows

GET	/api/get-packages	#59
POST	/api/subscribe-package	#60
POST	/api/get-user-current-package	#61
POST	/api/update-subscription-value	#62
POST	/api/get-package-listings	#63
POST	/api/create-package-listings	#64
POST	/api/get-package-details	#65
POST	/api/create-package-milestones	#66
GET	/api/get-users-for-package	#67
POST	/api/add-users-to-package	#68
POST	/api/get-milestone-users	#69
POST	/api/release-milestone-for-user	#70
POST	/api/get-user-milestones	#71

11. Desktop License Management & Hardware Lock

POST	/api/validate-license-key	#72
-------------	---------------------------	-----

12. Public Contact & Support

POST	/api/contact	#73
-------------	--------------	-----

13. Cafeteria & Wallet POS Module

POST	/sales/verify-wallet	#74
-------------	----------------------	-----

1. Authentication & Account Management

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 1. Authentication & Account Management.

1. User Registration

Registers a new user account with role buyer. Generates a unique username with timestamp suffix and a 6-digit random code. Dispatches an admin notification email and increments monthly spend limits.

Content-Type: application/json

Controller Action: ApiController@register

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
name	string	REQUIRED	required	Full legal name or business representative name.
email	string	REQUIRED	required unique:users,email	Unique user email address for credentials and notifications.
password	string	REQUIRED	required	Plaintext password to be hashed using Bcrypt.
confirm_password	string	REQUIRED	required same:password	Confirmation password matching password field.
type	string	REQUIRED	required	Registration credential type (e.g. "email").
phone	string	OPTIONAL	nullable	Mobile contact number (defaults to "123456789" if omitted).

EXAMPLE REQUEST BODY

```
{
  "name": "Ahmed Al-Mansoor",
  "email": "ahmed.mansoor@example.sa",
  "password": "SecurePassword123!",
  "confirm_password": "SecurePassword123!",
  "type": "email",
  "phone": "+966501234567"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/register" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"name":"Ahmed Al-Mansoor","email":"ahmed@example.sa","pa
```

SUCCESS RESPONSE (200 OK)

User successfully created and returned with related transactions and geographic info.

```
{
  "message": "Registered successfully !",
  "user": {
    "id": 104,
    "name": "Ahmed Al-Mansoor",
    "username": "Ahmed1042",
    "email": "ahmed.mansoor@example.sa",
    "phone_number": "+966501234567",
    "role": "buyer",
    "type": "email",
    "code": 839201,
    "wallet": "0.00",
    "created_at": "2026-10-19T10:00:00.000000Z",
    "transactions": [],
    "countries_info": null,
    "cities_info": null
  }
}
```

```
}  
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Validation Error

```
{  
  "email": [  
    "The email has already been taken."  
  ],  
  "confirm_password": [  
    "The confirm password and password must match."  
  ]  
}
```

HTTP 500 - Duplicate Email and Type

```
{  
  "message": "Email already existed !"  
}
```

Database & Financial State Impact: Inserts into `users` table with role='buyer'. Triggers `userService::incrementMonthlySpend()`. Dispatches Mail `AdminNotifications` to admin.

POST /api/login

PUBLIC

2. User Login Authentication

Authenticates a user via email and password credentials where registration type is email. Validates password hash and returns user profile with relational models.

Content-Type: application/json Controller Action: ApiController@login

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
email	string	REQUIRED	required email	Registered user email address.
password	string	REQUIRED	required	Account password.

EXAMPLE REQUEST BODY

```
{
  "email": "ahmed.mansoor@example.sa",
  "password": "SecurePassword123!"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/login" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"email":"ahmed.mansoor@example.sa","password":"SecurePassword123!"}'
```

SUCCESS RESPONSE (200 OK)

Returns authenticated user object with financial transactions and localization data.

```
{
  "message": "Logged in successfully !",
  "user": {
    "id": 104,
    "name": "Ahmed Al-Mansoor",
    "username": "Ahmed1042",
    "email": "ahmed.mansoor@example.sa",
    "role": "buyer",
    "wallet": "1250.00",
    "transactions": [],
    "countries_info": {
      "id": 1,
      "name": "Saudi Arabia"
    },
    "cities_info": {
      "id": 12,
      "name": "Riyadh"
    }
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Validation Error

```
{
  "email": [
    "The email field is required."
  ]
}
```

HTTP 500 - User Not Found or Invalid Password

```
{
  "message": "Password doesn't match..",
  "user": []
}
```

Database & Financial State Impact: Performs query on `users` table. Verifies Bcrypt hash against `password` column.

3. Google OAuth Social Login

Authenticates a user via Google OAuth identity. Looks up user by email and social type to return account information.

Content-Type: application/json

Controller Action: ApiController@googleLogin

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
email	string	REQUIRED	required email	Google authenticated email address.
type	string	REQUIRED	required	OAuth identifier (e.g. "google").

EXAMPLE REQUEST BODY

```
{
  "email": "user@gmail.com",
  "type": "google"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/google-login" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"email":"user@gmail.com","type":"google"}'
```

SUCCESS RESPONSE (200 OK)

Returns user profile if social account exists.

```
{
  "message": "Logged In Successfully..",
  "user": {
    "id": 88,
    "name": "Salem Al-Dawsari",
    "email": "salem@gmail.com",
    "type": "google",
    "role": "seller"
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 200 - Account Not Found

```
{
  "message": "No user found..",
  "user": []
}
```

HTTP 422 - Validation Failure

```
{
  "type": [
    "The type field is required."
  ]
}
```

Database & Financial State Impact: Queries `users` table for matching email and type records.

POST /api/forget-password

PUBLIC

4. Forget Password OTP Request

Generates a secure 6-digit numeric OTP, writes the code to the user database record, and dispatches an OTP email to the registered address.

Content-Type: application/json Controller Action: ApiController@forgetPassword

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
email	string	REQUIRED	required email	Registered account email address.

EXAMPLE REQUEST BODY

```
{
  "email": "ahmed.mansoor@example.sa"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/forget-password" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"email":"ahmed.mansoor@example.sa"}'
```

SUCCESS RESPONSE (200 OK)

OTP generated and sent to email successfully.

```
{
  "message": "OTP has been emailed to you successfully !",
  "user": {
    "id": 104,
    "email": "ahmed.mansoor@example.sa",
    "code": 749210
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Email Not Found

```
{
  "message": "Email does not exist"
}
```

HTTP 422 - Invalid Email Format

```
{
  "email": [
    "The email must be a valid email address."
  ]
}
```

Database & Financial State Impact: Updates `users.code` with new 6-digit random integer. Dispatches Mail `forgetemail`.

5. Reset / Update Password

Updates user password by user ID. Validates confirmation match and updates the encrypted password in the database.

Content-Type: application/json Controller Action: ApiController@updatePassword

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	ID of the user resetting the password.

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
password	string	REQUIRED	required	New password.
confirm_password	string	REQUIRED	required same:password	Password confirmation.

EXAMPLE REQUEST BODY

```
{
  "password": "NewSecurePassword456!",
  "confirm_password": "NewSecurePassword456!"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/update-password/104" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"password":"NewSecurePassword456!","confirm_password":
```

SUCCESS RESPONSE (200 OK)

Password updated and refreshed user profile returned.

```
{
  "message": "Password has been updated successfully !",
  "user": {
    "id": 104,
    "name": "Ahmed Al-Mansoor",
    "email": "ahmed.mansoor@example.sa"
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Validation Error

```
{
  "confirm_password": [
    "The confirm password and password must match."
  ]
}
```

Database & Financial State Impact: Updates `users.password` with `Hash::make(password)`.

POST

/api/delete-account

AUTHENTICATED

6. Delete User Account

Permanently removes a user account record from the users table by user ID.

Content-Type: application/json Controller Action: ApiController@deleteAccount

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	Unique ID of user account to be deleted.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/delete-account" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104}'
```

SUCCESS RESPONSE (200 OK)

Account successfully deleted.

```
{
  "message": "User Account has been deleted successfully"
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Missing User ID

```
{
  "user_id": [
    "The user id field is required."
  ]
}
```

Database & Financial State Impact: Deletes record from `users` table where `id = user\_id`.

2. User Profiles & Geography / Metadata

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 2. User Profiles & Geography / Metadata.

7. Get User Profile & Relations

Retrieves complete user record including transactions history, associated country model, and city model.

Content-Type: None Controller Action: ApiController@getUser

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-user/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Returns user profile with eager-loaded relations.

```
{
  "user": {
    "id": 104,
    "name": "Ahmed Al-Mansoor",
    "username": "Ahmed1042",
    "email": "ahmed.mansoor@example.sa",
    "phone_number": "+966501234567",
    "role": "buyer",
    "wallet": "2450.00",
    "profile_picture": "104ab8dfa7c8d9e.jpg",
    "country": 1,
    "city": 12,
    "transactions": [
      {
        "id": 201,
        "user_id": 104,
        "type": 5,
        "amount": "500.00",
        "created_at": "2026-10-18T14:20:00.000000Z"
      }
    ],
    "countries_info": {
      "id": 1,
      "name": "Saudi Arabia"
    },
    "cities_info": {
      "id": 12,
      "name": "Riyadh"
    }
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 200 - User Not Found (null returned)

```
{
  "user": null
}
```

Database & Financial State Impact: Queries `users` with `transactions`, `countriesInfo`, `citiesInfo`.

8. Update User Profile & Avatar

Updates personal information, localization settings, and profile picture avatar. Handles image uploads to public/profileImages/.

Content-Type: multipart/form-data Controller Action: ApiController@updateProfile

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	User ID to update.

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
name	string	REQUIRED	required	Updated user name.
email	string	REQUIRED	required email	Updated email address.
phone	string	REQUIRED	required	Primary phone number.
country	integer	OPTIONAL	nullable	Country ID.
city	integer	OPTIONAL	nullable	City ID.
type	string	OPTIONAL	nullable	Account type tag.
profile	file	OPTIONAL	nullable image	Avatar image file (JPEG, PNG, WEBP).

EXAMPLE REQUEST BODY

```
Form-Data: name='Ahmed Al-Mansoor', email='ahmed@example.sa', phone='+966501234567', country=1, city=12, profile=104randomString.jpg
```

SUCCESS RESPONSE (200 OK)

Profile updated successfully and refreshed model returned.

```
{
  "message": "Profile has been updated successfully !",
  "user": {
    "id": 104,
    "name": "Ahmed Al-Mansoor",
    "email": "ahmed@example.sa",
    "phone_number": "+966501234567",
    "profile_picture": "104randomString.jpg",
    "country": 1,
    "city": 12
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Validation Error

```
{
  "email": [
    "The email must be a valid email address."
  ],
  "phone": [
    "The phone field is required."
  ]
}
```

Database & Financial State Impact: Updates `users` row. Saves avatar image file into `public/profileImages/`.

GET `/api/get-countries`

PUBLIC

9. List All Countries

Returns alphabetical list of supported countries with their system IDs.

Content-Type: None **Controller Action:** ApiController@getAllCountries

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-countries" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of countries ordered by name.

```
{
  "list": [
    {
      "id": 1,
      "name": "Saudi Arabia",
      "code": "SA",
      "phone_code": "+966"
    },
    {
      "id": 2,
      "name": "United Arab Emirates",
      "code": "AE",
      "phone_code": "+971"
    },
    {
      "id": 3,
      "name": "Kuwait",
      "code": "KW",
      "phone_code": "+965"
    }
  ]
}
```

Database & Financial State Impact: Selects all rows from `countries` table ordered by name ASC.

10. List Cities by Country ID

Returns list of cities within a specific country identified by country ID.

Content-Type: None Controller Action: ApiController@getCities

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Country ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-cities/1" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of cities belonging to country.

```
{
  "list": [
    {
      "id": 12,
      "country_id": 1,
      "name": "Riyadh"
    },
    {
      "id": 13,
      "country_id": 1,
      "name": "Jeddah"
    },
    {
      "id": 14,
      "country_id": 1,
      "name": "Dammam"
    },
    {
      "id": 15,
      "country_id": 1,
      "name": "Al-Khobar"
    }
  ]
}
```

Database & Financial State Impact: Queries `cities` table where `country\_id = id`.

3. Trust-as-a-Service (TaaS) Listings & Deals

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 3. Trust-as-a-Service (TaaS) Listings & Deals.

11. Create Escrow Listing / Deal

Creates a new escrow deal/listing with product specifications, pricing, delivery charges, and media attachments. Evaluates user subscription limits before creation, notifies admin, and increments monthly spend.

Content-Type: multipart/form-data Controller Action: ApiController@createListing

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Seller User ID creating the listing.

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
name	string	REQUIRED	required	Listing title / Project name.
serial_number	string	REQUIRED	required	Product or contract serial identification number.
description	string	REQUIRED	required	Comprehensive description of items or milestone deliverables.
price	numeric	REQUIRED	required numeric	Base contract price in SAR.
delivery_charges	numeric	OPTIONAL	nullable numeric	Delivery or shipping fees in SAR.
cancellation_charges	numeric	OPTIONAL	nullable numeric	Cancellation fee in SAR if deal is terminated.
images[]	file[]	REQUIRED	required image	Array of gallery images uploaded to public/seller/listings/images/.
video	file	OPTIONAL	nullable	Video demonstration file uploaded to public/seller/listings/videos/.
doc	file	OPTIONAL	nullable	Contract PDF/document uploaded to public/seller/listings/documents/.

EXAMPLE REQUEST BODY

```
Form-Data: name='Web Development Contract', serial_number='SN-998822', description='Custom SaaS Development', pr
```

SUCCESS RESPONSE (200 OK)

Listing created successfully with gallery and milestones relations.

```
{
  "message": "Listing has been created successfully !",
  "listing": [
    {
```

```
{
  "id": 42,
  "user_id": 88,
  "name": "Enterprise ERP Integration",
  "serial_number": "SN-ERP-2026",
  "description": "Full ERP system development with milestone payments",
  "price": "15000.00",
  "delivery_charges": "0.00",
  "status": 0,
  "processing_fee": "0.00",
  "listing_gallery": [
    {
      "id": 91,
      "listing_id": 42,
      "image": "1729330000_spec.png"
    }
  ],
  "milestones": []
}
```

POSSIBLE ERROR RESPONSES

HTTP 403 - Subscription Limit Exceeded

```
{
  "message": "You have reached or this transaction will exceed your monthly spend limit. Please upgrade your package."
}
```

Database & Financial State Impact: Inserts into `listings` table (status 0: Pending). Inserts each file into `listingimages`. Triggers `userService::incrementMonthlySpend(id, amount)`. Sends admin notification email.

GET

/api/get-user-listing/{id}

AUTHENTICATED / USER ID

12. Get User Listings

Retrieves all listings created by a specific user with their gallery images and milestones.

Content-Type: None Controller Action: ApiController@getUserListing

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-user-listing/88" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of user listings.

```
{
  "listing": [
    {
      "id": 42,
      "user_id": 88,
      "name": "Enterprise ERP Integration",
      "price": "15000.00",
      "status": 0,
      "listing_gallery": [],
      "milestones": []
    }
  ]
}
```

Database & Financial State Impact: Selects from `listings` where `user\_id = id` with `listingGallery` and `milestones`.

POST /api/share-link

PUBLIC / AUTHENTICATED

13. Inspect & Resolve Shared Deal Link

Parses a shared deal URL (e.g. invitation acceptance link), extracts the embedded listing ID, and returns the listing details with gallery.

Content-Type: application/json Controller Action: ApiController@shareLink

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
link	string	REQUIRED	required	Full invitation acceptance link URL (e.g. "https://daf3.com/invitation-acceptance/42").

EXAMPLE REQUEST BODY

```
{
  "link": "https://daf3.com/invitation-acceptance/42"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/share-link" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"link":"https://daf3.com/invitation-acceptance/42"}'
```

SUCCESS RESPONSE (200 OK)

Listing details extracted from link.

```
{
  "success": true,
  "message": "You're watching the product!...",
  "id": {
    "id": "42"
  },
  "data": {
    "id": 42,
    "name": "Enterprise ERP Integration",
    "price": "15000.00",
    "serial_number": "SN-ERP-2026",
    "listing_gallery": []
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Invalid URL / ID Not Found

```
{
  "success": false,
  "message": "ID not found in the URL."
}
```

Database & Financial State Impact: Queries `listings` with `listingGallery` by parsed ID.

14. Get Buyer Listings

Retrieves all listings accepted by the specified buyer user ID with milestones, gallery, seller profile, and acceptance records.

Content-Type: None Controller Action: ApiController@buyerListing

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Buyer User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/buyer/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of accepted listings for buyer.

```
{
  "listing": [
    {
      "id": 42,
      "name": "Enterprise ERP Integration",
      "price": "15000.00",
      "status": 1,
      "user_info": {
        "id": 88,
        "name": "Seller Tech Corp"
      },
      "milestones": [
        {
          "id": 10,
          "title": "Milestone 1: Database Setup",
          "amount": "5000.00",
          "status": 1
        }
      ]
    }
  ]
}
```

Database & Financial State Impact: Selects `listings` having accepted relationship where `user\_id = id`.

15. Get Seller Listings

Retrieves all listings owned by the seller with their creator info, gallery, milestones, and accepted buyer info.

Content-Type: None Controller Action: ApiController@sellerListing

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Seller User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/seller/88" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of listings owned by seller.

```
{
  "listing": [
    {
      "id": 42,
      "user_id": 88,
      "name": "Enterprise ERP Integration",
      "price": "15000.00",
      "status": 1,
      "accepted_info": {
        "id": 104,
        "name": "Ahmed Al-Mansoor"
      },
      "milestones": []
    }
  ]
}
```

Database & Financial State Impact: Selects `listings` where `user\_id = id` with relations.

16. Accept Escrow Deal as Buyer

Buyer accepts a seller listing. Checks subscription spend quota, prevents self-acceptance or duplicate acceptance, transitions listing status to 1 (Active), inserts buyerlisting row, sends seller in-app notification, and increments monthly spend.

Content-Type: application/json Controller Action: ApiController@acceptedBuyerListing

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Listing ID to accept.
user_id	integer	REQUIRED	required	Buyer User ID accepting the deal.
status	integer	OPTIONAL	nullable	Optional status tag.

EXAMPLE REQUEST BODY

```
{
  "id": 42,
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/acceptedBuyerListing" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":42,"user_id":104}'
```

SUCCESS RESPONSE (200 OK)

Listing successfully accepted and activated.

```
{
  "message": "Listing added successfully",
  "acceptListing": {
    "id": 18,
    "user_id": 104,
    "listing_id": 42,
    "status": 1
  },
  "listing": [
    {
      "id": 42,
      "name": "Enterprise ERP Integration",
      "status": 1
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Self-Acceptance or Duplicate Error

```
{
  "message": "You're trying to accept your own listing."
}
```

HTTP 403 - Subscription Limit Exceeded

```
{
  "message": "You have reached or this transaction limit."
}
```

HTTP 404 - Listing Not Found

```
{
  "message": "Listing not found."
}
```

Database & Financial State Impact: Inserts into `buyerlistings`. Updates `listings.status = 1`. Creates row in `notifications` for seller. Calls `userServices::incrementMonthlySpend(buyer\_id, amount)`.

POST `/api/listing/markAsComplete`

AUTHENTICATED

17. Mark Listing as Completed

Marks an escrow deal as fully completed. Updates status on both listings and buyerlistings tables to 3 (Completed).

Content-Type: application/json **Controller Action:** ApiController@markAsCompleted

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Listing ID to complete.

EXAMPLE REQUEST BODY

```
{
  "id": 42
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/listing/markAsComplete" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":42}'
```

SUCCESS RESPONSE (200 OK)

Listing successfully marked completed.

```
{
  "success": "Listing has been marked as completed successfully",
  "updatedListing": {
    "id": 42,
    "status": 3
  },
  "updatedBuyerListing": [
    {
      "id": 18,
      "listing_id": 42,
      "status": 3
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 500 - Server Error

```
{
  "error": "Something went wrong, please try again later"
}
```

Database & Financial State Impact: Updates `listings.status = 3` and `buyerlistings.status = 3`.

18. Request Listing Cancellation

Submits a cancellation request for an active deal. Prevents duplicate requests, creates an administrative cancellation record, and flags the deal for admin mediation.

Content-Type: application/json Controller Action: ApiController@cancelBuyerListing

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Listing ID to cancel.
cancelled_by	integer	REQUIRED	required	User ID initiating cancellation.
reason	string	OPTIONAL	nullable	Reason for requesting cancellation.

EXAMPLE REQUEST BODY

```
{
  "id": 42,
  "cancelled_by": 104,
  "reason": "Deliverables did not meet technical specification agreements."
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/"
-H "Content-Type: application/json"
-H "Accept: application/json"
-d '{"id":42,"cancelled_by":104,"reason":"Deliverables did not meet technical specification agreements."}'
```

SUCCESS RESPONSE (200 OK)

Cancellation request logged for administrative review.

```
{
  "message": "Your request to cancel the listing has been sent to an admin. You will be notified soon.",
  "cancel_listing": {
    "id": 5,
    "cancelled_by": 104,
    "cancelled_to": 88,
    "listing_id": 42,
    "reason": "Deliverables did not meet technical specification agreements."
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Already Cancelled / Not Found

```
{
  "error": "Sorry, you can't cancel this listing more than once"
}
```

Database & Financial State Impact: Inserts record into `cancelllistings` table.

4. Milestone & Escrow Management

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 4. Milestone & Escrow Management.

19. Create Milestone for Listing

Creates a defined milestone deliverable with an amount, start date, and end date under an active listing. Generates an initial transaction audit row (type 5) and dispatches an in-app notification to the seller.

Content-Type: application/json Controller Action: ApiController@milestoneStore

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Listing ID.
user_id	integer	REQUIRED	required	User ID creating the milestone.
description	string	REQUIRED	required	Deliverable specification description.
start_date	date	REQUIRED	required date	Milestone commencement date (YYYY-MM-DD).
end_date	date	REQUIRED	required date after_or_equal:start_date	Milestone target completion date (YYYY-MM-DD).
amount	numeric	REQUIRED	required numeric min:0	Escrow amount allocated to this milestone in SAR.

EXAMPLE REQUEST BODY

```
{
  "id": 42,
  "user_id": 104,
  "description": "Frontend UI Components Delivery",
  "start_date": "2026-11-01",
  "end_date": "2026-11-15",
  "amount": 4500
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/milestone/store" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":42,"user_id":104,"description":"Frontend UI Components Delivery",
"start_date": "2026-11-01",
"end_date": "2026-11-15",
"amount": 4500}'
```

SUCCESS RESPONSE (200 OK)

Milestone successfully created.

```
{
  "success": "Milestone has been created successfully!",
  "milestone": {
    "id": 77,
    "user_id": 104,
    "listing_id": 42,
    "description": "Frontend UI Components Delivery",
    "start_date": "2026-11-01",
    "end_date": "2026-11-15",
    "amount": 4500,
    "status": 0
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Validation Error

```
{
  "messages": {
    "end_date": [
      "The end date must be a date after or equal to start date."
    ]
  }
}
```

Database & Financial State Impact: Inserts row in `milestones` (status 0: Pending). Inserts audit row in `transactions` (type 5). Inserts in-app notification in `notifications` for seller.

20. Get Listing Milestones Summary

Retrieves all milestones for a listing, filtering for accepted (1) and released (3) milestones to calculate total approved spend.

Content-Type: None Controller Action: ApiController@buyerMileStone

URI PATH PARAMETERS

Parameter	Type	Required	Description
listing_id	integer	REQUIRED	Listing ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/buyer/milestones/42" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Milestones summary and list returned.

```
{
  "listing_id": 42,
  "milestone_statuses": [
    1,
    3
  ],
  "total_filtered_milestones": 9500,
  "milestones": [
    {
      "id": 76,
      "listing_id": 42,
      "description": "Backend APIs",
      "amount": "5000.00",
      "status": 3
    },
    {
      "id": 77,
      "listing_id": 42,
      "description": "Frontend UI",
      "amount": "4500.00",
      "status": 1
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Listing Not Found

```
{
  "error": "Listing not found"
}
```

Database & Financial State Impact: Queries `listings` with `milestones`.

21. Release Full Milestone Funds

Transfers full escrow funds for a completed milestone to the seller. Sets status to 3 (Released), evaluates 3.5% commission if seller is on commission-based tier, credits net amount to seller wallet via transactionService, and logs financeSheet debit/credit rows and transaction audit log.

Content-Type: application/json Controller Action: ApiController@releaseMilestone

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Milestone ID to release.

EXAMPLE REQUEST BODY

```
{
  "id": 77
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/milestone/release" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":77}'
```

SUCCESS RESPONSE (200 OK)

Milestone funds released and seller wallet credited.

```
{
  "success": "Milestone Released Successfully"
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Milestone Not Found

```
{
  "error": "Milestone not found"
}
```

Database & Financial State Impact: Updates `milestones.status = 3`. Increments `users.wallet` for seller by net amount. Inserts row in `finance\_sheets` (type: credit, admin_commission: 3.5% or 0). Inserts row in `transactions` (type 1).

POST /api/milestone/partial-release

AUTHENTICATED

22. Release Partial Milestone Funds

Releases a portion of milestone funds to the seller while preserving the remaining amount in a new milestone record. Updates original milestone to released amount and status 3, credits seller wallet net of commission, and creates a child milestone with the remaining funds.

Content-Type: application/json **Controller Action:** ApiController@sellerPartialReleaseMilestone

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Original Milestone ID.
amount	numeric	REQUIRED	required numeric min:0	Partial amount to release (must be strictly less than current milestone amount).
user_id	integer	REQUIRED	required exists:users,id	Buyer User ID authorizing partial release.

EXAMPLE REQUEST BODY

```
{
  "id": 77,
  "amount": 2000,
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/milestone/partial-release" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":77,"amount":2000.00,"user_id":104}'
```

SUCCESS RESPONSE (200 OK)

Partial release completed.

```
{
  "success": "Milestone partially released successfully"
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Amount Exceeds Milestone Total

```
{
  "error": "Please enter a value less than 4500"
}
```

HTTP 404 - Milestone Not Found

```
{
  "error": "Milestone not found"
}
```

Database & Financial State Impact: Updates original milestone amount to released amount and status=3. Creates new milestone row for remaining balance. Credits seller wallet net of commission. Inserts row in `finance\_sheets`. Inserts row in `transactions` (type 2). Calls `userService::incrementMonthlySpend`.

23. Accept or Reject Milestone Action

Allows a seller to formally Accept (status 1) or Reject (status 2) a buyer-proposed milestone. Triggers appropriate in-app notifications to the buyer.

Content-Type: application/json Controller Action: ApiController@sellerMilestoneAction

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Milestone ID.

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
status	integer	REQUIRED	required	Action code: 1 = Accepted, 2 = Rejected.

EXAMPLE REQUEST BODY

```
{
  "status": 1
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/milestone/77/status" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"status":1}'
```

SUCCESS RESPONSE (200 OK)

Milestone status updated.

```
{
  "milestone": {
    "id": 77,
    "listing_id": 42,
    "status": 1
  }
}
```

Database & Financial State Impact: Updates `milestones.status`. Creates row in `notifications` for milestone creator.

POST

/api/milestones/cancel

AUTHENTICATED

24. Cancel Milestone

Cancels an existing milestone with an optional cancellation reason. Updates milestone status to 4 (Cancelled) and notifies the seller.

Content-Type: application/json Controller Action: ApiController@cancelMilestone

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Milestone ID to cancel.
reason	string	OPTIONAL	nullable	Explanation for cancellation.

EXAMPLE REQUEST BODY

```
{
  "id": 77,
  "reason": "Deliverable schedule renegotiated."
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/milestones/cancel" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"id":77,"reason":"Schedule changed"}'
```

SUCCESS RESPONSE (200 OK)

Milestone cancelled successfully.

```
{
  "status": "success",
  "message": "Milestone has been cancelled successfully"
}
```

Database & Financial State Impact: Updates `milestones.status = 4` and sets `reason`. Inserts notification in `notifications`.

GET

/api/milestone/{milestone_id}

AUTHENTICATED

25. Delete Milestone

Permanently deletes a milestone record from the database by ID.

Content-Type: None Controller Action: ApiController@deleteMilestone

URI PATH PARAMETERS

Parameter	Type	Required	Description
milestone_id	integer	REQUIRED	Milestone ID to delete.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/milestone/77" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Milestone deleted successfully.

```
{
  "message": "Milestone deleted successfully"
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Milestone Not Found

```
{
  "error": "Milestone not found"
}
```

Database & Financial State Impact: Deletes row from `milestones` table.

5. Digital Wallet, Ledger & Banking

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 5. Digital Wallet, Ledger & Banking.

GET

/api/transactions/{user_Id}

AUTHENTICATED / USER ID

26. Get User Transaction History

Retrieves all user transactions audit log entries from the transactions table (types: 1=milestone release, 2=partial release, 4=withdrawal, 5=deposit/milestone hold).

Content-Type: None Controller Action: ApiController@getTransaction

URI PATH PARAMETERS

Parameter	Type	Required	Description
user_Id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/transactions/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Transaction log entries returned.

```
{
  "transaction": [
    {
      "id": 101,
      "user_id": 104,
      "product_name": "Enterprise ERP Integration",
      "milestone": "UI Design",
      "type": 1,
      "amount": "5000.00",
      "created_at": "2026-10-18T10:00:00.000000Z"
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - No Transactions Found

```
{
  "message": "No transactions found for this user"
}
```

Database & Financial State Impact: Queries `transactions` where `user\_id = user\_Id`.

27. Get User Bank Account Details

Retrieves registered bank account details (IBAN, SWIFT, Account Holder, Bank Name) for a user from the userbank table.

Content-Type: None Controller Action: ApiController@userWallet

URI PATH PARAMETERS

Parameter	Type	Required	Description
user_id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/wallet-receipt/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Bank details returned.

```
{
  "status": "success",
  "userBankDetails": {
    "id": 12,
    "user_id": 104,
    "bank_name": "Al-Rajhi Bank",
    "country": "Saudi Arabia",
    "city": "Riyadh",
    "account_holder": "Ahmed Al-Mansoor",
    "swift_code": "RJHISARI",
    "iban_number": "SA0380000000608010167519"
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Bank Details Not Found

```
{
  "status": "error",
  "message": "User bank details not found"
}
```

Database & Financial State Impact: Queries `userbank` where `user\_id = user\_id`.

28. Store / Update Bank Details

Creates or updates bank account withdrawal information for a user. If bank details already exist for the user ID, it updates them; otherwise, inserts a new userbank record.

Content-Type: application/json Controller Action: ApiController@userWalletStore

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
bank_name	string	REQUIRED	required	Financial banking institution name.
country	string	REQUIRED	required	Bank registration country.
city	string	REQUIRED	required	Bank branch city.
account_holder	string	REQUIRED	required	Full legal name on bank account.
swift_code	string	REQUIRED	required	Bank SWIFT / BIC identifier.
iban_number	string	REQUIRED	required	International Bank Account Number (IBAN).

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "bank_name": "AL-Rajhi Bank",
  "country": "Saudi Arabia",
  "city": "Riyadh",
  "account_holder": "Ahmed Al-Mansoor",
  "swift_code": "RJHISARI",
  "iban_number": "SA0380000000608010167519"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/bank-details" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"bank_name":"AL-Rajhi Bank","country":"Sau
```

SUCCESS RESPONSE (200 OK)

Bank details stored successfully.

```
{
  "status": "success",
  "message": "Bank details have been added successfully"
}
```

Database & Financial State Impact: Inserts or updates row in `userbank` table.

POST /api/user-wallet-update

AUTHENTICATED

29. Update Existing Bank Details

Explicit endpoint to update existing bank details for a specified user ID.

Content-Type: application/json **Controller Action:** ApiController@userWalletUpdate

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
bank_name	string	REQUIRED	required	Bank institution name.
country	string	REQUIRED	required	Country.
city	string	REQUIRED	required	City.
account_holder	string	REQUIRED	required	Account holder legal name.
swift_code	string	REQUIRED	required	SWIFT code.
iban_number	string	REQUIRED	required	IBAN number.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "bank_name": "Saudi National Bank (SNB)",
  "country": "Saudi Arabia",
  "city": "Jeddah",
  "account_holder": "Ahmed Al-Mansoor",
  "swift_code": "NCBKSARI",
  "iban_number": "SA1210000001234567890123"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/user-wallet-update" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"bank_name":"SNB","country":"Saudi Arabia"}
```

SUCCESS RESPONSE (200 OK)

Bank details updated.

```
{
  "status": "success",
  "message": "Bank details have been updated successfully",
  "userBank": {
    "id": 12,
    "user_id": 104,
    "bank_name": "Saudi National Bank (SNB)"
  }
}
```

Database & Financial State Impact: Updates row in `userbank` table.

GET

/api/get-bank-details

PUBLIC / AUTHENTICATED

30. Get Platform Default Bank & Commission Settings

Retrieves global platform deposit bank instructions, IBAN, and default commission rate configured in the admin commision table.

Content-Type: None Controller Action: ApiController@getBankDetails

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-bank-details" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Platform banking and commission settings returned.

```
{
  "settings": {
    "id": 1,
    "commission": 3.5,
    "bank_name": "Riyad Bank",
    "account_holder": "DAF3 Trust Information Technology Est.",
    "iban_number": "SA4420000001234567890000",
    "swift_code": "RIBLSARI"
  }
}
```

Database & Financial State Impact: Selects `commision` record ID 1.

POST /api/wallet-to-wallet-transfer

AUTHENTICATED

31. Peer-to-Peer Wallet Transfer

Performs an instant peer-to-peer wallet balance transfer between two platform users using receiver email. Verifies sender balance, deducts from sender, credits receiver, and generates matching financeSheet debit and credit ledger rows.

Content-Type: application/json Controller Action: ApiController@walletToWalletTransfer

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required exists:users,id	Sender User ID.
receiver_email	string	REQUIRED	required email	Registered email address of recipient.
amount	numeric	REQUIRED	required numeric	Transfer amount in SAR.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "receiver_email": "partner@example.sa",
  "amount": 750
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/wallet-to-wallet-transfer" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"receiver_email":"partner@example.sa","amount":750}'
```

SUCCESS RESPONSE (200 OK)

Transfer completed successfully.

```
{
  "success": true,
  "message": "Wallet to wallet transfer successful",
  "data": {
    "sender": {
      "id": 104,
      "wallet": "1700.00"
    },
    "receiver": {
      "id": 110,
      "wallet": "750.00"
    }
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Insufficient Balance

```
{
  "success": false,
  "message": "Insufficient balance"
}
```

HTTP 422 - Validation Error

```
{
  "receiver_email": [
    "The receiver email field is required."
  ]
}
```

Database & Financial State Impact: Decrements `sender.wallet`. Increments `receiver.wallet`. Inserts debit row in `finance\_sheets` for sender. Inserts credit row in `finance\_sheets` for receiver.

32. Get Withdrawal Dashboard & Balances

Retrieves current wallet balance, platform commission applicability, registered bank details, and all withdrawal request records for the specified user.

Content-Type: None Controller Action: ApiController@withdraw

URI PATH PARAMETERS

Parameter	Type	Required	Description
user_id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/withdraw/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Withdrawal details and history returned.

```
{
  "bankInfo": true,
  "walletBalance": {
    "id": 104,
    "wallet": "2450.00"
  },
  "commission": {
    "commission": 0
  },
  "userBankDetails": {
    "bank_name": "Al-Rajhi Bank",
    "iban_number": "SA038000000608010167519"
  },
  "WithdrawRecord": [
    {
      "id": 1,
      "withdraw_Id": 98124,
      "withdraw_amount": "500.00",
      "status": 1
    }
  ]
}
```

Database & Financial State Impact: Selects `users`, `commision`, `userbank`, and `withdrawl` records.

POST /api/withdraw

AUTHENTICATED

33. Submit Withdrawal Request

Submits a request to withdraw wallet funds to an external bank account. Validates wallet balance, ensures no duplicate pending withdrawal request exists, immediately deducts funds from wallet, logs transaction audit row (type 4), and queues notification.

Content-Type: application/json Controller Action: ApiController@withdrawStore

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
bank_name	string	REQUIRED	required	Destination bank name.
country	string	REQUIRED	required	Bank country.
city	string	REQUIRED	required	Bank branch city.
account_holder	string	REQUIRED	required	Beneficiary account holder name.
swift_code	string	REQUIRED	required	SWIFT / BIC.
iban_number	string	REQUIRED	required	Beneficiary IBAN.
amount	numeric	REQUIRED	required numeric min:1	Amount to withdraw in SAR.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "bank_name": "AL-Rajhi Bank",
  "country": "Saudi Arabia",
  "city": "Riyadh",
  "account_holder": "Ahmed Al-Mansoor",
  "swift_code": "RJHISARI",
  "iban_number": "SA038000000608010167519",
  "amount": 1000
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/withdraw" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"bank_name":"AL-Rajhi Bank","country":"Sa
```

SUCCESS RESPONSE (200 OK)

Withdrawal request submitted successfully.

```
{
  "message": "Withdrawal request sent successfully"
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Insufficient Balance or Pending Request

```
{
  "error": "You already have a pending withdrawal request"
}
```

Database & Financial State Impact: Decrements `users.wallet` by amount. Inserts row in `withdrawal` with status 0 (Pending). Inserts row in `transactions` (type 4). Creates row in `notifications`.

POST /api/deposit

AUTHENTICATED

34. Deposit Funds to Wallet

Deposits funds into the user wallet balance. Checks monthly subscription limit quota, increments user wallet balance, records depositamount row, inserts transaction audit row (type 5), pushes notification, and increments monthly spend used.

Content-Type: application/json Controller Action: ApiController@depositStore

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
amount	numeric	REQUIRED	required numeric min:1	Deposit amount in SAR.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "amount": 5000
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/deposit" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"amount":5000.00}'
```

SUCCESS RESPONSE (200 OK)

Funds successfully deposited.

```
{
  "message": "The amount has been successfully deposited to your wallet"
}
```

POSSIBLE ERROR RESPONSES

HTTP 403 - Subscription Limit Exceeded

```
{
  "message": "You have reached or this transaction will exceed your monthly spend limit. Please upgrade your package."
}
```

Database & Financial State Impact: Increments `users.wallet`. Inserts into `depositamount`. Inserts into `transactions` (type 5). Inserts into `notifications`. Calls `userServices::incrementMonthlySpend`.

6. HyperPay Payment Gateway Integration

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 6. HyperPay Payment Gateway Integration.

POST

/api/payment/hyperpay

AUTHENTICATED

35. Initiate HyperPay Checkout Session

Creates a secure payment checkout session with HyperPay gateway using server-to-server REST communication. Generates a unique merchantTransactionId, acquires a checkoutId, and returns a redirect mobile payment URL.

Content-Type: application/json Controller Action: ApiController@initiatePayment

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
amount	numeric	REQUIRED	required numeric	Transaction amount in SAR.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "amount": 1500
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/payment/hyperpay" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"amount":1500.00}'
```

SUCCESS RESPONSE (200 OK)

Checkout session created and mobile redirect URL generated.

```
{
  "url": "https://daf3.com/mobile-app-payment/8AC7A4CA8F140A7C018F14F1C62A12DE.uat01-vm-tx02"
}
```

POSSIBLE ERROR RESPONSES

HTTP 500 - Gateway Initiation Failure

```
{
  "success": false,
  "message": "Payment initiation failed: Connection refused"
}
```

Database & Financial State Impact: Queries user localization information. Sends HTTP POST to HyperPay `/checkouts` endpoint.

GET

/api/payment/hyperpay/status/{checkoutId}/{user_id}

AUTHENTICATED / GATEWAY CALLBACK

36. Verify HyperPay Payment Status

Queries HyperPay gateway to verify the settlement status of a checkout ID. If approved (result code 000.100.110), increments user wallet balance by deposit amount, records depositamount entry, inserts transaction audit row (type 5), pushes in-app notification, and increments monthly spend.

Content-Type: None Controller Action: ApiController@checkPaymentStatus

URI PATH PARAMETERS

Parameter	Type	Required	Description
checkoutId	string	REQUIRED	HyperPay unique checkout session ID.
user_id	integer	REQUIRED	User ID receiving the deposit.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/payment/hyperpay/status/8AC7A4CA8F140A7C018F14F1C62A12DE/104" -H "Accept: appl
```

SUCCESS RESPONSE (200 OK)

Payment verified and wallet credited.

```
{
  "success": true,
  "message": "Payment was successful"
}
```

POSSIBLE ERROR RESPONSES

HTTP 200 - Payment Declined by Issuer / Gateway

```
{
  "success": false,
  "message": "Payment failed: Card expired or insufficient funds"
}
```

HTTP 500 - Status Verification Exception

```
{
  "success": false,
  "message": "Payment status check failed"
}
```

Database & Financial State Impact: Increments `users.wallet`. Inserts into `depositamount`. Inserts into `transactions` (type 5). Inserts into `notifications`. Calls `userServices::incrementMonthlySpend`.

7. Invoicing & Digital Billing

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 7. Invoicing & Digital Billing.

37. Generate Dynamic Invoice with QR Code

Generates a digital invoice record with an embedded SVG QR code pointing directly to the invoice payment web page. Checks subscription limits, generates an invoice code, saves SVG to public/qrcodes/, logs invoice record, and sends admin email alert.

Content-Type: application/json Controller Action: ApiController@storeInvoice

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required exists:users,id	Creator User ID.
amount	numeric	REQUIRED	required numeric min:0	Invoice bill amount in SAR.
type	string	REQUIRED	required	Invoice type (e.g. "credit").

EXAMPLE REQUEST BODY

```
{
  "user_id": 88,
  "amount": 3200,
  "type": "credit"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/store-invoice" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":88,"amount":3200.00,"type":"credit"}'
```

SUCCESS RESPONSE (201 OK)

Invoice created with SVG QR code URL.

```
{
  "success": true,
  "message": "Invoice created successfully",
  "data": {
    "id": 55,
    "user_id": 88,
    "code": "19102612345688",
    "amount": 3200,
    "code_image": "qrcodes/19102612345688.svg"
  },
  "qr_code_url": "https://daf3.com/qrcodes/19102612345688.svg"
}
```

POSSIBLE ERROR RESPONSES

HTTP 403 - Subscription Limit Exceeded

```
{
  "message": "You have reached or this transaction will exceed your monthly spend limit. Please upgrade your package."
}
```

Database & Financial State Impact: Inserts row into `invoices` table. Generates SVG file in `public/qrcodes/`. Dispatches admin notification email.

POST

/api/get-invoices

AUTHENTICATED

38. Retrieve User Invoices

Retrieves all generated digital invoices for the specified user ID.

Content-Type: application/json Controller Action: ApiController@getInvoices

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 88
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-invoices" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":88}'
```

SUCCESS RESPONSE (200 OK)

List of invoices returned.

```
{
  "message": "success",
  "data": [
    {
      "id": 55,
      "user_id": 88,
      "code": "19102612345688",
      "amount": "3200.00",
      "code_image": "qrcores/19102612345688.svg"
    }
  ]
}
```

Database & Financial State Impact: Selects `invoices` where `user\_id` = user_id`.

39. Get Milestone Invoice Details

Fetches detailed milestone invoice including listing info, creator info, and dynamic commission rate depending on whether seller is commission-based or flat-tier.

Content-Type: None Controller Action: ApiController@invoice

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Milestone ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/invoice/77" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Milestone invoice and commission metadata returned.

```
{
  "invoice": {
    "id": 77,
    "listing_id": 42,
    "amount": "4500.00",
    "description": "Frontend UI Components Delivery",
    "listing_info": {
      "id": 42,
      "name": "Enterprise ERP Integration"
    },
    "user_info": {
      "id": 104,
      "name": "Ahmed Al-Mansoor"
    }
  },
  "commission": {
    "id": 1,
    "commission": 0
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Invoice Not Found

```
{
  "error": "Invoice not found"
}
```

Database & Financial State Impact: Queries `milestones` with `listingInfo` and `userInfo`. Evaluates commission rates.

GET

/api/delete/{id}

AUTHENTICATED

40. Delete Invoice

Permanently deletes an invoice record by invoice ID.

Content-Type: None Controller Action: ApiController@invoiceDelete

URI PATH PARAMETERS

Parameter	Type	Required	Description
id	integer	REQUIRED	Invoice ID to delete.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/delete/55" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Invoice successfully deleted.

```
{
  "success": true,
  "message": "Invoice deleted successfully",
  "data": {
    "id": 55,
    "code": "19102612345688"
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Invoice Not Found

```
{
  "success": false,
  "message": "Invoice not found"
}
```

Database & Financial State Impact: Deletes row from `invoices` table.

POST /api/export-invoices-pdf

AUTHENTICATED

41. Export Invoices to PDF

Generates a styled A4 portrait PDF compilation of all invoices for a user using DomPDF. Stores file in storage/app/public/exports/ and returns a public download URL.

Content-Type: application/json Controller Action: ApiController@exportInvoicesPDF

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required exists:users,id	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 88
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/export-invoices-pdf" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":88}'
```

SUCCESS RESPONSE (200 OK)

PDF generated with download URL.

```
{
  "success": true,
  "file_name": "invoices_88_1729331000.pdf",
  "download_url": "https://daf3.com/storage/exports/invoices_88_1729331000.pdf"
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - No Invoices Found

```
{
  "success": false,
  "message": "No invoices found"
}
```

Database & Financial State Impact: Selects `invoices`. Writes PDF file into storage public exports.

POST

/api/export-invoices-excel

AUTHENTICATED

42. Export Invoices to Excel (.xlsx)

Exports all user invoices to an Excel spreadsheet (.xlsx) using Maatwebsite Excel and returns public download URL.

Content-Type: application/json Controller Action: ApiController@exportInvoicesExcel

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required exists:users,id	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 88
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/export-invoices-excel" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":88}'
```

SUCCESS RESPONSE (200 OK)

Excel spreadsheet generated.

```
{
  "success": true,
  "file_name": "invoices_88_1729331000.xlsx",
  "download_url": "https://daf3.com/storage/exports/invoices_88_1729331000.xlsx"
}
```

Database & Financial State Impact: Generates XLSX spreadsheet in storage public directory.

43. Export Milestones to PDF

Generates an official A4 milestone statement PDF for an escrow listing and returns download URL.

Content-Type: application/json Controller Action: ApiController@exportMilestonesPDF

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
listing_id	integer	REQUIRED	required exists:listings,id	Listing ID.

EXAMPLE REQUEST BODY

```
{
  "listing_id": 42
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/export-milestones-pdf" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"listing_id":42}'
```

SUCCESS RESPONSE (200 OK)

PDF report generated.

```
{
  "success": true,
  "file_name": "milestones_42_1729331000.pdf",
  "download_url": "https://daf3.com/storage/exports/milestones_42_1729331000.pdf"
}
```

Database & Financial State Impact: Loads `milestones` where `listing\_id = listing\_id`. Writes PDF file into storage.

POST

/api/export-milestones-excel

AUTHENTICATED

44. Export Milestones to Excel (.xlsx)

Exports all milestones associated with a deal into an Excel spreadsheet.

Content-Type: application/json Controller Action: ApiController@exportMilestonesExcel

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
listing_id	integer	REQUIRED	required exists:listings,id	Listing ID.

EXAMPLE REQUEST BODY

```
{
  "listing_id": 42
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/export-milestones-excel" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"listing_id":42}'
```

SUCCESS RESPONSE (200 OK)

Excel sheet generated.

```
{
  "success": true,
  "file_name": "milestones_42_1729331000.xlsx",
  "download_url": "https://daf3.com/storage/exports/milestones_42_1729331000.xlsx"
}
```

Database & Financial State Impact: Queries `milestones`. Exports spreadsheet to public storage.

8. Manual Top-Up Requests

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 8. Manual Top-Up Requests.

45. Get User Manual Top-Up Requests & Summary

Retrieves all manual bank transfer top-up requests for a user, providing computed aggregates: total top-up amount, pending amount, approved amount, total count, and individual request records.

Content-Type: application/json Controller Action: ApiController@getManualTopUps

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-manual-top-ups" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104}'
```

SUCCESS RESPONSE (200 OK)

Top-up statistics and request list returned.

```
{
  "count": 3,
  "totalTopUp": 7500,
  "totalPendingTopup": 2500,
  "totalApprovedTopup": 5000,
  "list": [
    {
      "id": 14,
      "user_id": 104,
      "amount": "2500.00",
      "image": "slip_104_1729330000.jpg",
      "status": "pending",
      "created_at": "2026-10-19T08:00:00.000000Z"
    },
    {
      "id": 9,
      "user_id": 104,
      "amount": "5000.00",
      "image": "slip_104_1728000000.jpg",
      "status": "approved",
      "created_at": "2026-10-10T12:00:00.000000Z"
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - Validation Error

```
{
  "success": false,
  "message": "Validation failed",
  "errors": {
    "user_id": [
      "The user id field is required."
    ]
  }
}
```

Database & Financial State Impact: Queries `manual\_topups` table where `user\_id = user\_id`.

POST `/api/store-manual-top-up`

AUTHENTICATED

46. Submit Manual Bank Slip Top-Up

Submits a physical or electronic bank transfer slip image for admin verification. Uploads image to public/transferrSlips/, inserts a manualTopup record with pending status, and returns updated user totals.

Content-Type: multipart/form-data **Controller Action:** ApiController@storeManualTopup

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
amount	numeric	REQUIRED	required	Transferred amount in SAR.
image	file	REQUIRED	required	Bank transfer receipt image or PDF uploaded via FileHelper.

EXAMPLE REQUEST BODY

Form-Data: user_id=104, amount=2500.00, image=[binary slip file]

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/store-m
-F "user_id=104" \
-F "amount=2500.00" \
-F "image=@/path/to/transfer_receipt.jpg"
```

SUCCESS RESPONSE (200 OK)

Transfer slip uploaded and recorded under pending status.

```
{
  "count": 4,
  "totalTopUp": 10000,
  "totalPendingTopup": 5000,
  "totalApprovedTopup": 5000,
  "list": [
    {
      "id": 15,
      "user_id": 104,
      "amount": "2500.00",
      "image": "slip_104_1729332000.jpg",
      "status": "pending"
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 400 - User Not Found or Validation Error

```
{
  "message": "No user Found !"
}
```

Database & Financial State Impact: Uploads image to `public/transferrSlips/`. Inserts row into `manual\_topups` with status='pending'.

9. In-App Chat, Messaging & Online Presence

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 9. In-App Chat, Messaging & Online Presence.

GET /api/chats/{chat_id}/messages

AUTHENTICATED

47. Fetch Messages for Chat Thread

Retrieves all historical chat messages exchanged in a specific conversation thread identified by chat_id.

Content-Type: None Controller Action: ApiController@fetchMessages

URI PATH PARAMETERS

Parameter	Type	Required	Description
chat_id	integer	REQUIRED	Chat thread ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/chats/22/messages" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

List of messages returned.

```
{
  "status": "success",
  "message": [
    {
      "id": 1,
      "chat_id": 22,
      "user_id": 104,
      "message": "Hello, have you reviewed the milestone specs?",
      "uploads": null,
      "created_at": "2026-10-19T09:00:00.000000Z"
    },
    {
      "id": 2,
      "chat_id": 22,
      "user_id": 88,
      "message": "Yes, we are commencing development today.",
      "uploads": null,
      "created_at": "2026-10-19T09:05:00.000000Z"
    }
  ]
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - No Messages Found

```
{
  "status": "error",
  "message": []
}
```

Database & Financial State Impact: Selects rows from `chats` where `chat\_id = chat\_id`.

48. Fetch User Conversations & Mark Seen

Marks all incoming chats for user as viewed and seen (view=1, seen=1), then merges all conversations where the user acts as buyer or seller.

Content-Type: None Controller Action: ApiController@chatView

URI PATH PARAMETERS

Parameter	Type	Required	Description
user_id	integer	REQUIRED	User ID.

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/chats/104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

All conversations returned.

```
{
  "allListings": [
    {
      "id": 22,
      "buyer_id": 104,
      "seller_id": 88,
      "listing_id": 42,
      "listing_info": {
        "id": 42,
        "name": "Enterprise ERP Integration"
      }
    }
  ]
}
```

Database & Financial State Impact: Updates `chats` (view=1, seen=1). Queries `chat\_created` with `listingInfo`.

49. Initiate or Retrieve Chat by Listing

Retrieves an existing conversation thread or automatically initializes a new chatCreated thread for a specific deal between buyer and seller.

Content-Type: application/json Controller Action: ApiController@chatViewByListing

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required integer	Requesting User ID.
type	string	REQUIRED	required string	User role context ("buyer" or "seller").
id	integer	REQUIRED	required integer	Listing ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "type": "buyer",
  "id": 42
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/chat-listing" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"type":"buyer","id":42}'
```

SUCCESS RESPONSE (200 OK)

Chat thread retrieved or established.

```
{
  "status": "success",
  "alllistings": {
    "id": 22,
    "buyer_id": 104,
    "seller_id": 88,
    "listing_id": 42,
    "listing_info": {
      "id": 42,
      "name": "Enterprise ERP Integration"
    }
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 401 - Listing Not Accepted Yet

```
{
  "status": "error",
  "message": "Listing not accepted yet"
}
```

Database & Financial State Impact: Checks or inserts row into 'chat_created'.

50. Send Chat Message with Optional Attachment

Posts a new message to a chat thread with optional file upload (images/PDFs saved to public/chatuploads/). Evaluates whether receiver is currently online (in online_users table) to flag notification delivery status.

Content-Type: multipart/form-data Controller Action: ApiController@sendMessage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	Chat thread ID.
user_id	integer	REQUIRED	required	Sender User ID.
message	string	OPTIONAL	nullable	Text message content.
uploads	file	OPTIONAL	nullable file	Optional image or document attachment.

EXAMPLE REQUEST BODY

Form-Data: id=22, user_id=104, message='Here is the updated design mockup', uploads=[design.png]

CURL INVOC.

```
curl -X POST \
  -F "id=22" \
  -F "user_id=104" \
  -F "message='Here is the updated design mockup'" \
  -F "uploads=@design.png" \
  http://localhost:8080/api/send-message
```

SUCCESS RESPONSE (200 OK)

Message delivered and notifications queued.

```
{
  "status": "success",
  "msg": "Message sent successfully.",
  "message": [
    {
      "id": 3,
      "chat_id": 22,
      "user_id": 104,
      "message": "Design approved.",
      "created_at": "2026-10-19T09:10:00.000000Z"
    }
  ],
  "sender": {
    "id": 104,
    "name": "Ahmed Al-Mansoor"
  },
  "Receiver": {
    "id": 22,
    "buyer_id": 104,
    "seller_id": 88
  }
}
```

Database & Financial State Impact: Inserts into `chats`. Checks `online\_users`. Inserts into `chatnotifications`.

POST

/api/seen-message

AUTHENTICATED

51. Mark Chat Messages as Seen

Updates read receipts for chat notifications and messages where user_id matches (seen=1).

Content-Type: application/json Controller Action: ApiController@seenMessage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/seen-message" -H "Content-Type: application/json" -d '{
```

SUCCESS RESPONSE (200 OK)

Seen status updated.

```
{
  "message": "Messages marked as seen"
}
```

Database & Financial State Impact: Updates `chatnotifications.seen = 1` and `chats.seen = 1`.

POST

/api/seen-general-message

AUTHENTICATED

52. Mark General Notifications as Seen

Sets seen = 1 on general system notifications for the specified user.

Content-Type: application/json Controller Action: ApiController@seenGeneralMessage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/seen-general-message" -H "Content-Type: application/json"
```

SUCCESS RESPONSE (200 OK)

Notifications updated.

```
{
  "message": "Notifications marked as seen"
}
```

Database & Financial State Impact: Updates `notifications.seen = 1`.

POST

/api/update-notification

AUTHENTICATED

53. Mark Chat Notifications as Read

Marks chat notifications status as 1 (read) and chat view status as 1.

Content-Type: application/json Controller Action: ApiController@updateNotification

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/update-notification" -H "Content-Type: application/json"
```

SUCCESS RESPONSE (200 OK)

Notifications and chats updated.

```
{
  "message": "Notifications and chats updated successfully"
}
```

Database & Financial State Impact: Updates `chatnotifications.status = 1` and `chats.view = 1`.

file:///C:/Projects/Laravel/Daf3-Latest/Daf3_Complete_API_Documentation.html

73/101

POST

/api/general-update-notification

AUTHENTICATED

54. Mark General Notifications as Read

Updates view = 1 on general notifications for user.

Content-Type: application/json Controller Action: ApiController@generalUpdateNotification

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/general-update-notification" -H "Content-Type: applicat
```

SUCCESS RESPONSE (200 OK)

General notifications updated.

```
{
  "message": "Notifications updated successfully"
}
```

Database & Financial State Impact: Updates `notifications.view = 1`.

GET /api/notifications/latest-count

AUTHENTICATED

55. Get Unread Chat Notification Count

Retrieves latest 5 chat notifications and calculates count of unread (status = 0) items.

Content-Type: None **Controller Action:** ApiController@latestNotificationsCount

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/notifications/latest-count?user_id=104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Unread count and notification items returned.

```
{
  "latestChatNotificationsCount": 2,
  "notifications": [
    {
      "id": 41,
      "user_id": 104,
      "title": "New Message",
      "description": "You have a new message... ",
      "status": 0
    }
  ]
}
```

Database & Financial State Impact: Queries `chatnotifications` where `user\_id = user\_id`.

GET /api/notifications/general-latest-count

AUTHENTICATED

56. Get Unread General Notification Count

Retrieves latest 5 general notifications and calculates count of unread (status = 0) items.

Content-Type: None **Controller Action:** ApiController@generalLatestNotificationsCount

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/notifications/general-latest-count?user_id=104" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Unread general notification count returned.

```
{
  "generallatestNotificationsCount": 1,
  "notifications": [
    {
      "id": 18,
      "user_id": 104,
      "title": "Milestone Released",
      "description": "Your milestone has been released ... ",
      "status": 0
    }
  ]
}
```

Database & Financial State Impact: Queries `notifications` where `user\_id = user\_id`.

POST

/api/enter-chat

AUTHENTICATED

57. Register Online Chat Room Presence

Registers user as active and online in the online_users table with a secure random session cookie key.

Content-Type: application/json Controller Action: ApiController@enterChat

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	User ID entering chat room.

EXAMPLE REQUEST BODY

```
{
  "id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/enter-chat" -H "Content-Type: application/json" -d '{"id":104'}
```

SUCCESS RESPONSE (200 OK)

User registered online.

```
{
  "message": "User entered the chat successfully"
}
```

Database & Financial State Impact: Inserts into `online\_users` if user not already present.

POST

/api/leave-chat

AUTHENTICATED

58. Deregister Chat Room Presence

Removes user presence entry from online_users table upon leaving the conversation view.

Content-Type: application/json Controller Action: ApiController@leaveChat

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
id	integer	REQUIRED	required	User ID leaving chat room.

EXAMPLE REQUEST BODY

```
{
  "id": 104
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/leave-chat" -H "Content-Type: application/json" -d '{"id":104}'
```

SUCCESS RESPONSE (200 OK)

User deregistered from active chat presence.

```
{
  "message": "User left the chat successfully"
}
```

Database & Financial State Impact: Deletes record from `online\_users` where `user\_id = id`.

10. Packages, Subscriptions & Multi-User Escrows

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 10. Packages, Subscriptions & Multi-User Escrows.

GET

/api/get-packages

PUBLIC / AUTHENTICATED

59. List Subscription Packages

Retrieves all available subscription tiers (Free, Pro, Enterprise, Commission-based) with their pricing, monthly spend limits, processing fees, and durations.

Content-Type: None Controller Action: ApiController@getPackages

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-packages" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Packages list returned.

```
{
  "list": [
    {
      "id": 1,
      "package_name": "Starter Escrow",
      "package_price": "0.00",
      "duration": 30,
      "monthly_spend": 10000,
      "processing_fee": "15.00"
    },
    {
      "id": 2,
      "package_name": "Enterprise TaaS",
      "package_price": "499.00",
      "duration": 30,
      "monthly_spend": 100000,
      "processing_fee": "0.00"
    }
  ]
}
```

Database & Financial State Impact: Queries all rows from `packages` table.

60. Subscribe to Tiered Package

Enrolls user into a subscription package. Automatically expires any active existing subscription, sets new start and end dates based on package duration, initializes monthly_spend_used to 0, and activates the plan.

Content-Type: application/json Controller Action: ApiController@subscribePackage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID subscribing.
package_id	integer	REQUIRED	required	Selected Package ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "package_id": 2
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/subscribe-package" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"package_id":2}'
```

SUCCESS RESPONSE (200 OK)

Subscription activated.

```
{
  "item": {
    "id": 19,
    "user_id": 104,
    "pkg_id": 2,
    "package_name": "Enterprise TaaS",
    "price": "499.00",
    "duration": 30,
    "monthly_spend": 100000,
    "monthly_spend_used": 0,
    "start_date": "2026-10-19",
    "expires_at": "2026-11-18",
    "status": "active"
  }
}
```

Database & Financial State Impact: Updates prior active subscriptions to status='expired'. Inserts new row in `subscriptions` table.

POST

/api/get-user-current-package

AUTHENTICATED

61. Get User Current Subscription

Checks if user has an active subscription, returning subscription status flag and plan details.

Content-Type: application/json Controller Action: ApiController@getUsercurrentPackage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{  "user_id": 104}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-user-current-package" -H "Content-Type: application"
```

SUCCESS RESPONSE (200 OK)

Active subscription status returned.

```
{  "subscribed": true,  "item": {    "id": 19,    "package_name": "Enterprise TaaS",    "monthly_spend": 100000,    "monthly_spend_used": 15000,    "status": "active",    "expires_at": "2026-11-18"  }}
```

Database & Financial State Impact: Queries `subscriptions` where `user\_id = user\_id` and `status = 'active'`.

62. Increment / Decrement Subscription Usage

Directly adjusts subscription usage metrics (validated for monthly_spend_used) with increment or decrement actions.

Content-Type: application/json Controller Action: ApiController@updateSubscriptionValues

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
subscription_id	integer	REQUIRED	required	Subscription ID.
column	string	REQUIRED	required	Column to update (restricted to "monthly_spend_used").
value	numeric	REQUIRED	required	Adjustment value.
type	string	REQUIRED	required	Action type ("increment" or "decrement").

EXAMPLE REQUEST BODY

```
{
  "subscription_id": 19,
  "column": "monthly_spend_used",
  "value": 5000,
  "type": "increment"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/update-subscription-value" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"subscription_id":19,"column":"monthly_spend_used","value":5000,"t
```

SUCCESS RESPONSE (200 OK)

Subscription usage updated.

```
{
  "message": "Subscription updated successfully !",
  "item": {
    "id": 19,
    "monthly_spend_used": 20000
  }
}
```

Database & Financial State Impact: Updates `subscriptions` column based on math operation.

POST

/api/get-package-listings

AUTHENTICATED

63. Get User Package Listings

Retrieves all package-based escrow contract listings owned by user with their milestones.

Content-Type: application/json Controller Action: ApiController@getPackageListings

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{  "user_id": 104}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-package-listings" -H "Content-Type: application/json"
```

SUCCESS RESPONSE (200 OK)

Package listings returned.

```
{  "count": 1,  "list": [    {      "id": 8,      "user_id": 104,      "title": "Q4 Vendor Sourcing Package",      "code": "202610199",      "milestones": []    }  ]}
```

Database & Financial State Impact: Queries `package\_listings` where `user\_id = user\_id` with `milestones`.

64. Create Package Listing Contract

Creates a master package listing grouping multiple milestones and assigned users. Validates subscription limits and generates a unique package code.

Content-Type: application/json Controller Action: ApiController@createPackageListings

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.
title	string	REQUIRED	required	Contract / Package title.

EXAMPLE REQUEST BODY

```
{
  "user_id": 104,
  "title": "Logistics Fleet Maintenance Package"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/create-package-listings" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"user_id":104,"title":"Logistics Fleet Maintenance Package"}
```

SUCCESS RESPONSE (200 OK)

Package listing created.

```
{
  "count": 2,
  "list": [
    {
      "id": 9,
      "user_id": 104,
      "title": "Logistics Fleet Maintenance Package",
      "code": "2026101910"
    }
  ]
}
```

Database & Financial State Impact: Inserts into `package\_listings`. Calls `userService::incrementMonthlySpend`.

65. Get Package Listing Details

Retrieves complete details of a package listing including all child milestones.

Content-Type: application/json Controller Action: ApiController@getPackageDetails

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
package_id	integer	REQUIRED	required	Package listing ID.

EXAMPLE REQUEST BODY

```
{
  "package_id": 9
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-package-details" -H "Content-Type: application/json"
```

SUCCESS RESPONSE (200 OK)

Package details with milestones returned.

```
{
  "item": {
    "id": 9,
    "title": "Logistics Fleet Maintenance Package",
    "code": "2026101910",
    "milestones": [
      {
        "id": 14,
        "title": "Vehicle Inspections",
        "amount": "3000.00"
      }
    ]
  }
}
```

Database & Financial State Impact: Queries `package\_listings` with `milestones`.

66. Add Milestone to Package Listing

Creates a child milestone deliverable under an existing package listing.

Content-Type: application/json Controller Action: ApiController@createPackageMilestones

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
package_id	integer	REQUIRED	required	Package listing ID.
user_id	integer	REQUIRED	required	User ID.
name	string	REQUIRED	required	Milestone name.
description	string	REQUIRED	required	Deliverable description.
amount	numeric	REQUIRED	required	Milestone amount in SAR.

EXAMPLE REQUEST BODY

```
{
  "package_id": 9,
  "user_id": 104,
  "name": "Tire Replacements",
  "description": "Supply of commercial tires",
  "amount": 4500
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/create-package-milestones"
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"package_id":9,"user_id":104,"name":"Tire Replacement
```

SUCCESS RESPONSE (200 OK)

Milestone created under package.

```
{
  "item": {
    "id": 9,
    "title": "Logistics Fleet Maintenance Package",
    "milestones": [
      {
        "id": 15,
        "title": "Tire Replacements",
        "amount": "4500.00"
      }
    ]
  }
}
```

Database & Financial State Impact: Inserts row into `package\_milestones`.

GET

/api/get-users-for-package

AUTHENTICATED

67. Get Eligible Users for Package Allocation

Returns list of active platform users (status = 1) who can be linked to package milestones.

Content-Type: None

Controller Action: ApiController@getUsersForPackage

CURL INVOCATION

```
curl -X GET "https://daf3.com/api/get-users-for-package" -H "Accept: application/json"
```

SUCCESS RESPONSE (200 OK)

Active users returned.

```
{
  "count": 24,
  "listUsers": [
    {
      "id": 104,
      "name": "Ahmed Al-Mansoor",
      "email": "ahmed@example.sa",
      "status": 1
    }
  ]
}
```

Database & Financial State Impact: Selects `users` where `status = 1`.

POST /api/add-users-to-package

AUTHENTICATED

68. Associate Users to Package Milestones

Links multiple users to all milestones existing under a package listing, creating individual packageUsers allocations.

Content-Type: application/json **Controller Action:** ApiController@addUsersToPackage

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
package_id	integer	REQUIRED	required	Package listing ID.
user_id	integer	REQUIRED	required	Creator User ID.
users	array	REQUIRED	required	Array of user IDs to associate.

EXAMPLE REQUEST BODY

```
{
  "package_id": 9,
  "user_id": 104,
  "users": [
    110,
    112
  ]
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/add-users-to-package" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"package_id":9,"user_id":104,"users":[110,112]}'
```

SUCCESS RESPONSE (200 OK)

Users linked to package milestones.

```
{
  "item": {
    "id": 9,
    "title": "Logistics Fleet Maintenance Package"
  }
}
```

Database & Financial State Impact: Inserts rows into `package\_users` for each user and milestone pair.

POST

/api/get-milestone-users

AUTHENTICATED

69. Get Users Assigned to Package Milestone

Retrieves all users assigned to a specific package milestone along with milestone specs.

Content-Type: application/json Controller Action: ApiController@getMilestoneUsers

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
milestone_id	integer	REQUIRED	required	Package Milestone ID.

EXAMPLE REQUEST BODY

```
{  "milestone_id": 15}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-milestone-users" -H "Content-Type: application/"
```

SUCCESS RESPONSE (200 OK)

Assigned users returned.

```
{  "item": {    "id": 15,    "title": "Tire Replacements",    "amount": "4500.00"  },  "users": [    {      "id": 31,      "package_milestone_id": 15,      "user_id": 110,      "amount": "4500.00",      "status": "pending"    }  ]}
```

Database & Financial State Impact: Queries `package\_milestones` with `packageUsers`.

70. Release Package Milestone to User

Releases milestone funds to an assigned user. Deducts commission if on a commission-based plan, credits the user wallet via transactionService, and updates allocation status to released.

Content-Type: application/json

Controller Action: ApiController@releaseMilestoneForUser

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
milestone_id	integer	REQUIRED	required	Package user allocation ID (package_users.id).

EXAMPLE REQUEST BODY

```
{
  "milestone_id": 31
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/release-milestone-for-user" -H "Content-Type: appli"
```

SUCCESS RESPONSE (200 OK)

Milestone funds credited to user wallet.

```
{
  "message": "Milestone released successfully",
  "item": {
    "id": 15,
    "title": "Tire Replacements"
  },
  "users": [
    {
      "id": 31,
      "status": "released"
    }
  ]
}
```

Database & Financial State Impact: Credits `users.wallet` net of commission. Inserts row into `finance\_sheets`. Updates `package\_users.status = 'released'`.

71. Get User Assigned Package Milestones

Retrieves all package milestones assigned to a specific user with package and milestone details.

Content-Type: application/json Controller Action: ApiController@getUserMilestones

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
user_id	integer	REQUIRED	required	User ID.

EXAMPLE REQUEST BODY

```
{
  "user_id": 110
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/get-user-milestones" -H "Content-Type: application/json"
```

SUCCESS RESPONSE (200 OK)

User package milestones returned.

```
{
  "count": 1,
  "list": [
    {
      "id": 31,
      "user_id": 110,
      "amount": "4500.00",
      "status": "released",
      "package_milestone": {
        "id": 15,
        "title": "Tire Replacements"
      },
      "package_listing": {
        "id": 9,
        "title": "Logistics Fleet Maintenance Package"
      }
    }
  ]
}
```

Database & Financial State Impact: Queries `package\_users` with `packageMilestone` and `packageListing`.

11. Desktop License Management & Hardware Lock

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 11. Desktop License Management & Hardware Lock.

POST /api/validate-license-key

PUBLIC / CLIENT APP

72. Validate Desktop License & Register Hardware Device

Validates a commercial desktop software license key against userDesktopLicense records. Verifies active status, ensures activated terminal count is within allowed seat limits (license.users), and binds the machine fingerprint (device_data) into userDesktopLicenseMeta.

Content-Type: application/json Controller Action: LicenseController@validateLicenseKey

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
license_key	string	REQUIRED	required	32-character formatted desktop license key (e.g. "ABCD-EFGH-JKLM-NPQR-STUV-WXYZ-2345-6789").
device_data	object string	REQUIRED	required	Machine hardware fingerprint object (CPU ID, BIOS UUID, MAC address, Machine GUID, Hostname).

EXAMPLE REQUEST BODY

```
{
  "license_key": "DAF3-PRO-8899-7711-2233-4455",
  "device_data": {
    "machine_guid": "e4eaaaf2-d142-11e1-b3e4-080027620c41",
    "cpu_id": "BFEBFBFF000906EA",
    "os": "Windows 11 Pro 64-bit",
    "hostname": "POS-TERMINAL-01"
  }
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/api/validate-license-key" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"license_key": "DAF3-PRO-8899", "device_data": { "machine_guid": "e4eaaaf2-d142-11e1-b3e4-080027620c41", "cpu_id": "BFEBFBFF000906EA", "os": "Windows 11 Pro 64-bit", "hostname": "POS-TERMINAL-01" }}
```

SUCCESS RESPONSE (200 OK)

License key is valid and hardware device registered.

```
{
  "status": "success",
  "message": "License key is valid"
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Invalid License Key

```
{
  "status": "error",
  "message": "License key is not valid"
}
```

HTTP 422 - License Expired

```
{
  "status": "error",
  "message": "License is expired"
}
```

HTTP 422 - Device Seat Quota Full

```
{
  "status": "error",
  "message": "License is full"
}
```

Database & Financial State Impact: Queries `user\_desktop\_licenses`. Inserts or updates hardware fingerprint in `user\_desktop\_license metas`.

12. Public Contact & Support

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 12. Public Contact & Support.

73. Submit Contact / Support Inquiry

Receives user inquiries, bug reports, and partnership messages. Validates required details and logs the message into the contacts table.

Content-Type: application/json Controller Action: ApiController@storeContact

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
name	string	REQUIRED	required	Sender full name.
email	string	REQUIRED	required email	Sender reply email address.
subject	string	REQUIRED	required	Inquiry subject line.
message	string	REQUIRED	required	Detailed message body.

EXAMPLE REQUEST BODY

```
{
  "name": "Tariq Al-Harbi",
  "email": "tariq@business.sa",
  "subject": "API Integration Inquiry for Enterprise ERP",
  "message": "We are integrating DAF3 Escrow into our B2B procurement platform and require custom webhook config"
}
```

SUCCESS RESPONSE (200 OK)

Support message submitted.

```
{
  "message": "Your query has been submitted successfully. We will contact you as soon as possible",
  "contactInfo": {
    "id": 8,
    "name": "Tariq Al-Harbi",
    "email": "tariq@business.sa",
    "subject": "API Integration Inquiry for Enterprise ERP"
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 422 - Validation Error

```
{
  "email": [
    "The email must be a valid email address."
  ]
}
```

Database & Financial State Impact: Inserts record into `contacts` table.

13. Cafeteria & Wallet POS Module

Detailed endpoint specifications, validation parameters, curl examples, and database state transitions for 13. Cafeteria & Wallet POS Module.

74. Verify Customer / Student Wallet & Daily Limit (POS)

Specialized AJAX/API POS endpoint for cafeteria, retail, and campus terminals. Verifies customer wallet identity via QR code, NFC string, student ID, or username. Validates positive wallet balance, enforces daily spending caps (daily_limit), aggregates today's spending from sales_receipts, and computes real-time remaining daily allowance.

Content-Type: application/json Controller Action: SalesController@verifyWalletCode

REQUEST BODY PARAMETERS

Parameter	Type	Requirement	Validation	Description
wallet_code	string	REQUIRED	required string	Raw scanned wallet code, QR payload, RFID tag, student ID, username, phone, or email.

EXAMPLE REQUEST BODY

```
{
  "wallet_code": "DAF03-STU-88912"
}
```

CURL INVOCATION

```
curl -X POST "https://daf3.com/sales/verify-wallet" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{"wallet_code":"DAF03-STU-88912"}'
```

SUCCESS RESPONSE (200 OK)

Customer wallet verified with balance and daily limits.

```
{
  "status": 200,
  "success": true,
  "message": "Customer wallet verified successfully.",
  "data": {
    "customer": {
      "id": 205,
      "name": "Zaid Al-Ghamdi",
      "username": "ZaidGhamdi",
      "code": "STU-88912",
      "role": "buyer",
      "daily_limit": 50,
      "wallet": 320
    },
    "wallet_balance": 320,
    "daily_limit": 50,
    "today_spent": 15,
    "remaining_daily_limit": 35
  }
}
```

POSSIBLE ERROR RESPONSES

HTTP 404 - Customer Not Found

```
{
  "status": 404,
  "success": false,
  "message": "Invalid Wallet Code! Customer not found."
}
```

HTTP

```
{
}
```

HTTP 422 - Daily Limit Exhausted

```
{
  "status": 422,
  "success": false,
  "message": "Daily spending limit reached! Customer has already spent 50.00 of 50.00 SAR daily limit."
}
```

Database & Financial State Impact: Queries `users` table across multiple code identifiers. Aggregates `sales\_receipts` sum where `student\_id = customer\_id` and `date = today`.

3. Developer Integration SDKs & Quickstart Guide

This guide provides battle-tested implementation code snippets in JavaScript (Axios) and Flutter/Dart for communicating with DAF3 Escrow endpoints.

3.1 JavaScript / TypeScript SDK Client (Axios)

```
import axios from 'axios';

const daf3Client = axios.create({
  baseURL: 'https://daf3.com/api',
  timeout: 15000,
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json',
  }
});

// Example: Authenticate and Fetch Listings
export async function authenticateAndLoad(email, password) {
  try {
    const loginRes = await daf3Client.post('/login', { email, password });
    const user = loginRes.data.user;
    console.log(`Logged in as: ${user.name} (Wallet: ${user.wallet} SAR)`);

    const listingsRes = await daf3Client.get(`/get-user-listing/${user.id}`);
    return { user, listings: listingsRes.data.listing };
  } catch (error) {
    console.error('API Error:', error.response?.data || error.message);
    throw error;
  }
}
```

3.2 Mobile SDK Client (Flutter / Dart)

```
import 'dart:convert';
import 'package:http/http.dart' as http;

class Daf3ApiClient {
  final String baseUrl = 'https://daf3.com/api';

  Future<Map<String, dynamic>> releaseMilestone(int milestoneId) async {
    final response = await http.post(
      Uri.parse('$baseUrl/milestone/release'),
      headers: {'Accept': 'application/json', 'Content-Type': 'application/json'},
      body: jsonEncode({'id': milestoneId}),
    );

    if (response.statusCode == 200) {
      return jsonDecode(response.body);
    } else {
      throw Exception('Failed to release milestone: ${response.body}');
    }
  }
}
```

4. Database Schema Dictionary & Status Enums

4.1 Milestone Status Codes

Status Code	State Label	Description & Financial State
STATUS 0	Pending / Created	Milestone has been proposed and created. Awaiting seller confirmation.
STATUS 1	Accepted / Active	Milestone accepted by seller. Work in progress. Escrow funds secured.
STATUS 2	Rejected	Milestone deliverable or terms rejected by seller.
STATUS 3	Released / Settled	Work verified and accepted by buyer. Funds disbursed to seller wallet net of commission.
STATUS 4	Cancelled / Terminated	Milestone cancelled. Reason stored in database; funds refunded or subjected to mediation.

4.2 Transaction Audit Types (transactions.type)

Type ID	Audit Classification	Direction & Behavior
1	Full Milestone Release	Credit to seller wallet upon full milestone approval.
2	Partial Milestone Release	Credit to seller wallet for partial release amount.
4	Wallet Withdrawal	Debit from user wallet balance to external bank account.
5	Deposit / Milestone Hold	Credit to user wallet balance via HyperPay or manual slip.