### **Spécifications Détaillées des API Backend - Africage**

### **1. Introduction**

Ce document détaille l'ensemble des points d'accès (endpoints) de l'API RESTful pour le backend d'Africage. Il a pour but de servir de référence technique pour les développeurs frontend et mobile afin d'intégrer les fonctionnalités de l'application. L'API est conçue pour être prévisible, cohérente et sécurisée.

#### **Principes Généraux**

* **URL de base :** https://api.africage.com/v1/
* **Format des données :** Toutes les requêtes et réponses seront au format application/json.
* **Authentification :** L'accès aux endpoints sécurisés se fait via un jeton JWT (JSON Web Token). Le jeton doit être inclus dans l'en-tête Authorization de chaque requête : Authorization: Bearer <votre\_jeton\_jwt>.
* **Gestion des erreurs :** L'API utilise les codes de statut HTTP standards pour indiquer le succès ou l'échec d'une requête. Un corps de réponse JSON avec un message d'erreur clair sera fourni en cas d'échec.
  + 200 OK : Requête réussie.
  + 201 Created : Ressource créée avec succès.
  + 400 Bad Request : La requête est malformée (ex: champ manquant).
  + 401 Unauthorized : Jeton d'authentification manquant ou invalide.
  + 403 Forbidden : L'utilisateur n'a pas les droits pour accéder à la ressource.
  + 404 Not Found : La ressource demandée n'existe pas.
  + 500 Internal Server Error : Erreur côté serveur.

### **2. Modèles de Données (Data Models)**

Voici les objets JSON principaux qui circulent dans l'API.

**User:** {

"id": "user\_uuid\_123",

"firstName": "John",

"lastName": "Doe",

"email": "john.doe@email.com",

"phone": "+221771234567",

"profilePictureUrl": "https://cdn.africage.com/profiles/user\_uuid\_123.jpg",

"averageRating": 4.8,

"kycStatus": "VERIFIED" // PENDING, VERIFIED, REJECTED

}

**Parcel (Colis):** {

"id": "parcel\_uuid\_456",

"senderId": "user\_uuid\_123",

"travelerId": "user\_uuid\_789", // null si non assigné

"origin": { "city": "Dakar", "country": "Sénégal" },

"destination": { "city": "Abidjan", "country": "Côte d'Ivoire" },

"description": "Documents importants",

"size": "SMALL", // SMALL, MEDIUM, LARGE

"weightKg": 2,

"photoUrl": "https://cdn.africage.com/parcels/parcel\_uuid\_456.jpg",

"proposedPrice": 15000, // En XOF ou devise locale

"status": "IN\_TRANSIT", // PENDING, ACCEPTED, IN\_TRANSIT, DELIVERED, CANCELLED

"trackingCode": "AFR789DKRABJ"

}

**Trip (Trajet):** {

"id": "trip\_uuid\_abc",

"travelerId": "user\_uuid\_789",

"origin": { "city": "Dakar", "country": "Sénégal" },

"destination": { "city": "Abidjan", "country": "Côte d'Ivoire" },

"departureDate": "2025-08-15T09:00:00Z",

"arrivalDate": "2025-08-16T18:00:00Z",

"availableSpace": "MEDIUM" // SMALL, MEDIUM, LARGE

}

### **3. Endpoints de l'API**

#### **3.1. Authentification & Gestion des Utilisateurs**

* **POST /auth/register**
  + **Description :** Crée un nouveau compte utilisateur.
  + **Auth :** Aucune.
  + **Corps (Request Body) :** { "firstName": "...", "lastName": "...", "email": "...", "phone": "...", "password": "..." }
  + **Réponse (201 Created) :** { "user": { ...User object... }, "token": "jwt\_token" }
* **POST /auth/login**
  + **Description :** Connecte un utilisateur et retourne un jeton JWT.
  + **Auth :** Aucune.
  + **Corps (Request Body) :** { "email": "...", "password": "..." }
  + **Réponse (200 OK) :** { "user": { ...User object... }, "token": "jwt\_token" }
* **GET /users/me**
  + **Description :** Récupère le profil de l'utilisateur actuellement connecté.
  + **Auth :** Requise.
  + **Réponse (200 OK) :** { ...User object... }
* **PUT /users/me**
  + **Description :** Met à jour le profil de l'utilisateur connecté.
  + **Auth :** Requise.
  + **Corps (Request Body) :** { "firstName": "...", "lastName": "...", "phone": "..." } (champs optionnels)
  + **Réponse (200 OK) :** { ...User object updated... }
* **POST /users/me/kyc**
  + **Description :** Soumet les documents pour la vérification KYC. Utilise un multipart/form-data pour l'upload de fichier.
  + **Auth :** Requise.
  + **Réponse (202 Accepted) :** { "message": "Documents submitted for verification." }

#### **3.2. Gestion des Colis (Côté Expéditeur)**

* **POST /parcels**
  + **Description :** Crée une nouvelle annonce de colis à envoyer.
  + **Auth :** Requise.
  + **Corps (Request Body) :** { "origin": {...}, "destination": {...}, "description": "...", "size": "...", "weightKg": ..., "proposedPrice": ... }
  + **Réponse (201 Created) :** { ...Parcel object... }
* **GET /parcels/my-shipments**
  + **Description :** Liste tous les colis envoyés par l'utilisateur connecté.
  + **Auth :** Requise.
  + **Réponse (200 OK) :** [ { ...Parcel object... }, { ...Parcel object... } ]

#### **3.3. Gestion des Trajets & Transport (Côté Voyageur)**

* **POST /trips**
  + **Description :** Crée une nouvelle annonce de trajet.
  + **Auth :** Requise.
  + **Corps (Request Body) :** { "origin": {...}, "destination": {...}, "departureDate": "...", "arrivalDate": "...", "availableSpace": "..." }
  + **Réponse (201 Created) :** { ...Trip object... }
* **GET /parcels/search**
  + **Description :** Recherche des colis disponibles correspondant à un trajet.
  + **Auth :** Requise.
  + **Paramètres (Query Params) :** ?originCity=Dakar&destinationCity=Abidjan&maxSize=MEDIUM
  + **Réponse (200 OK) :** [ { ...Parcel object... }, { ...Parcel object... } ]
* **POST /parcels/{parcelId}/accept**
  + **Description :** Un voyageur accepte de transporter un colis.
  + **Auth :** Requise.
  + **Paramètres (URL) :** parcelId - L'ID du colis à accepter.
  + **Réponse (200 OK) :** { ...Parcel object updated with travelerId and status ACCEPTED... }

#### **3.4. Suivi & Mise à jour du Statut**

* **GET /parcels/{parcelId}**
  + **Description :** Récupère les détails d'un colis spécifique.
  + **Auth :** Requise.
  + **Réponse (200 OK) :** { ...Parcel object... }
* **PUT /parcels/{parcelId}/status**
  + **Description :** Met à jour le statut d'un colis (utilisé par le voyageur).
  + **Auth :** Requise (seul le voyageur assigné peut le faire).
  + **Corps (Request Body) :** { "status": "IN\_TRANSIT" } ou { "status": "DELIVERED", "deliveryCode": "..." }
  + **Réponse (200 OK) :** { ...Parcel object updated... }

#### **3.5. Messagerie**

* **GET /chats**
  + **Description :** Liste toutes les conversations de l'utilisateur.
  + **Auth :** Requise.
  + **Réponse (200 OK) :** [ { "withUser": { ...User object... }, "lastMessage": "...", "timestamp": "..." }, ... ]
* **GET /chats/{userId}**
  + **Description :** Récupère l'historique des messages avec un autre utilisateur.
  + **Auth :** Requise.
  + **Réponse (200 OK) :** [ { "senderId": "...", "content": "...", "timestamp": "..." }, ... ]
* **POST /chats/{userId}**
  + **Description :** Envoie un message à un autre utilisateur.
  + **Auth :** Requise.
  + **Corps (Request Body) :** { "content": "Bonjour, je suis intéressé par votre annonce." }
  + **Réponse (201 Created) :** { "message": "Message sent successfully." }