

- * Connexion : <http://localhost:8080/api/auth/login>
- * Obtenir tous les utilisateurs : <http://localhost:8080/api/admin/users>
- * Obtenir le profil de l'utilisateur connecté : <http://localhost:8080/api/users/me>
- * Obtenir les offres de l'agriculteur : <http://localhost:8080/api/farmer/offers>
- * Obtenir toutes les offres disponibles : <http://localhost:8080/api/buyer/offers>

6. Gestion des produits (Admin)

- * Créer un nouveau produit : <http://localhost:8080/api/admin/products>

7. Gestion des offres (Admin)

- * Obtenir toutes les offres : <http://localhost:8080/api/admin/offers>
- * Obtenir toutes les commandes : <http://localhost:8080/api/admin/orders>
- * Créer un reçu d'entrepôt : <http://localhost:8080/api/admin/tokenization/warehouse-receipts>
- * Obtenir les notifications de l'utilisateur : <http://localhost:8080/api/notifications>
- * Obtenir les statistiques du dashboard : <http://localhost:8080/api/admin/dashboard>

| > merde , met chaque lien au dessus de son json |

◆ Mes excuses pour la confusion. Je vais reformater la liste pour que chaque JSON soit immédiatement précédé de son endpoint complet

(méthode HTTP et URL), comme vous l'avez demandé.

1. Authentification

1.1. Connexion utilisateur

JSON de Requête (LoginRequest):

```
`json
```

```
{  
  "email": "utilisateur@example.com",  
  "password": "motdepasse"  
}
```

JSON de Réponse (JwtAuthenticationResponse):

```
`json  
{  
  "success": true,  
  "message": "Connexion réussie",  
  "data": {  
    "accessToken": "eyJhbGciOiJIUzUxMi...",  
    "refreshToken": "eyJhbGciOiJIUzUxMi...",  
    "tokenType": "Bearer"  
  },  
  "timestamp": "2025-10-06T..."  
}
```

1.2. Inscription utilisateur

```
`json  
{  
  "firstName": "Jean",  
  "lastName": "Dupont",  
  "email": "jean.dupont@example.com",  
  "password": "securePassword123",  
  "phone": "0123456789",  
  "role": "FARMER",  
  "farmName": "La Ferme Joyeuse",  
}
```

```
"farmLocation": "Campagne",  
"farmSize": "100 hectares"  
}
```

,

JSON de Requête (SignupRequest - Exemple Acheteur):

```
`json  
{  
  "firstName": "Sophie",  
  "lastName": "Martin",  
  "email": "sophie.martin@company.com",  
  "password": "securePassword123",  
  "phone": "0987654321",  
  "role": "BUYER",  
  "companyName": "Agro Distribution SARL",  
  "activityType": "Grossiste",  
  "companyAddress": "10 Rue de la Commerce"  
}
```

,

JSON de Réponse (ApiResponse<String>):

```
`json  
{  
  "success": true,  
  "message": "Utilisateur enregistré avec succès. ID: 123",  
  "data": null,  
  "timestamp": "2025-10-06T..."  
}
```

,

1.3. Rafraîchir le token

1.4. Déconnexion

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<String>):

```
`json
{
  "success": true,
  "message": "Déconnexion réussie",
  "data": null,
  "timestamp": "2025-10-06T..."
}
`
```

2. Gestion des utilisateurs (AdminController)

2.1. Obtenir tous les utilisateurs

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<UserProfileDto>>):

```
`json
{
  "success": true,
  "message": "Utilisateurs récupérés avec succès",
  "data": {
    "content": [
      {
        "id": 1,
```

```
"firstName": "Admin",
"lastName": "User",
"email": "admin@agrilend.com",
"phone": "0102030405",
"role": "ADMIN",
"hederaAccountId": "0.0.12345",
"createdAt": "2025-09-23T...",
"updatedAt": "2025-09-24T...",
"farmName": null, "farmLocation": null, "farmSize": null,
"companyName": null, "activityType": null, "companyAddress": null,
"active": true
},
{
  "id": 2,
  "firstName": "Sy",
  "lastName": "Modibo",
  "email": "artifiquea@gmail.com",
  "phone": "98765432",
  "role": "FARMER",
  "hederaAccountId": "0.0.67890",
  "createdAt": "2025-09-23T...",
  "updatedAt": "2025-09-24T...",
  "farmName": "Foro", "farmLocation": "San", "farmSize": "50 hectare",
  "companyName": null, "activityType": null, "companyAddress": null,
  "active": true
}
],
"page": 0, "size": 10, "totalElements": 2, "totalPages": 1,
"first": true, "last": true, "empty": false
},
"timestamp": "2025-10-06T..."
```

```
}
```

```
,
```

2.2. Obtenir un utilisateur par ID

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<UserProfileDto>): (Similaire à un élément du content ci-dessus)

2.3. Activer un utilisateur

```
`json
```

```
{
```

```
  "success": true,
```

```
  "message": "Utilisateur activé avec succès",
```

```
  "data": null,
```

```
  "timestamp": "2025-10-06T..."
```

```
}
```

```
,
```

2.4. Désactiver un utilisateur

```
`json
```

```
{
```

```
  "success": true,
```

```
  "message": "Utilisateur désactivé avec succès",
```

```
  "data": null,
```

```
  "timestamp": "2025-10-06T..."
```

```
}
```

```
,
```

2.5. Mettre à jour le profil d'un utilisateur (Admin)

JSON de Requête (UserProfileDto - Exemple Agriculteur):

```
`json
{
  "id": 2,
  "firstName": "Sy",
  "lastName": "Modibo",
  "email": "artifiquea@gmail.com",
  "phone": "98765432",
  "role": "FARMER",
  "hederaAccountId": "0.0.67890",
  "isActive": true,
  "farmName": "Foro",
  "farmLocation": "San",
  "farmSize": "55 hectares"
},
```

JSON de Réponse (ApiResponse<UserProfileDto>): (Similaire au JSON de requête, avec updatedAt mis à jour)

3. Profil utilisateur (UserController)

3.1. Obtenir le profil de l'utilisateur connecté

3.2. Mettre à jour le profil de l'utilisateur connecté

```
`json
{
  "firstName": "MonNouveauPrénom",
```

```
"lastName": "MonNouveauNom",
"phone": "0000000000",
"hederaAccountId": "0.0.12345"
// N'incluez pas l'ID, le rôle ou l'email ici, ils ne sont généralement pas modifiables via cet endpoint
}
`
```

JSON de Réponse (ApiResponse<UserProfileDto>): (Similaire au JSON de requête, avec updatedAt mis à jour)

4. Fonctionnalités Agriculteur (FarmerController)

4.1. Obtenir les offres de l'agriculteur

```
`json
{
  "success": true,
  "message": "Offres de l'agriculteur récupérées avec succès",
  "data": {
    "content": [
      {
        "id": 101,
        "productId": 1,
        "productName": "Tomates",
        "quantity": 500.0,
        "quantityUnit": "KG",
        "suggestedPrice": 1.50,
        "status": "ACTIVE",
        "availabilityDate": "2025-11-01",
        "farmerId": 2,
```



```

        "farmerName": "Sy Modibo"
    }
],
"page": 0, "size": 10, "totalElements": 1, "totalPages": 1,
"first": true, "last": true, "empty": false
},
"timestamp": "2025-10-06T..."
}
`

```

4.2. Créer une offre

```

`json
{
    "productId": 1,
    "quantity": 1000.0,
    "quantityUnit": "KG",
    "suggestedPrice": 1.20,
    "harvestDate": "2025-10-20",
    "availabilityDate": "2025-10-25",
    "expiryDate": "2025-11-30",
    "qualityGrade": "A",
    "originLocation": "Ma Ferme",
    "productionMethod": "ORGANIC"
}
`

```

JSON de Réponse (ApiResponse<OfferDto>): (Similaire au JSON de requête, avec id et createdAt)

4.3. Mettre à jour une offre

JSON de Requête (OfferDto): (Similaire au JSON de création, avec l'id de l'offre)

```
`json
{
  "id": 101,
  "productId": 1,
  "quantity": 1200.0, // Quantité modifiée
  "quantityUnit": "KG",
  "suggestedPrice": 1.25, // Prix modifié
  "harvestDate": "2025-10-20",
  "availabilityDate": "2025-10-25",
  "expiryDate": "2025-11-30",
  "qualityGrade": "A",
  "originLocation": "Ma Ferme",
  "productionMethod": "ORGANIC"
}
`
```

JSON de Réponse (ApiResponse<OfferDto>): (Similaire au JSON de requête, avec updatedAt mis à jour)

4.4. Supprimer une offre

```
`json
{
  "success": true,
  "message": "Offre supprimée avec succès",
  "data": null,
  "timestamp": "2025-10-06T..."
}
`
```

5. Fonctionnalités Acheteur (BuyerController)

5.1. Obtenir toutes les offres disponibles

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<OfferDto>>): (Similaire à la réponse getAllOffers de l'Admin)

5.2. Obtenir une offre par ID

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<OfferDto>): (Similaire à un élément du content ci-dessus)

5.3. Passer une commande

`json

```
{
  "offerId": 101,
  "quantity": 200.0,
  "deliveryAddress": "20 Rue des Champs, Ville"
},
```

JSON de Réponse (ApiResponse<OrderDto>):

`json

```
{
  "success": true,
  "message": "Commande passée avec succès",
  "data": {
    "id": 201,
    "buyerId": 3,
```

```

    "offerId": 101,
    "quantity": 200.0,
    "unitPrice": 1.50,
    "totalPrice": 300.0,
    "platformFee": 15.0,
    "status": "IN_ESCROW",
    "escrowTransactionId": "0.0.123456789",
    "expectedDeliveryDate": "2025-11-05",
    "deliveryAddress": "20 Rue des Champs, Ville"
  },
  "timestamp": "2025-10-06T..."
}
`

```

5.4. Obtenir les commandes de l'acheteur

5.5. Confirmer la livraison d'une commande

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<OrderDto>): (Similaire au JSON de commande, avec status mis à jour à DELIVERED)

6. Gestion des produits (AdminController)

6.1. Créer un nouveau produit

`json

{

 "name": "Pommes Gala",

 "description": "Pommes fraîches de saison",

```
"category": "Fruits",
"subcategory": "Pommes",
"unit": "KG",
"imageUrl": "http://example.com/pommes.jpg"
}
`
```

JSON de Réponse (ApiResponse<ProductDto>): (Similaire au JSON de requête, avec id et createdAt)

6.2. Mettre à jour un produit

JSON de Requête (ProductDto):

```
`json
{
  "id": 1,
  "name": "Pommes Gala Bio", // Nom modifié
  "description": "Pommes fraîches de saison, biologiques",
  "category": "Fruits",
  "subcategory": "Pommes",
  "unit": "KG",
  "imageUrl": "http://example.com/pommes_bio.jpg"
}
`
```

JSON de Réponse (ApiResponse<ProductDto>): (Similaire au JSON de requête, avec updatedAt mis à jour)

6.3. Désactiver un produit

```
`json
{
  "success": true,

```

```
"message": "Produit désactivé avec succès",  
"data": null,  
"timestamp": "2025-10-06T..."  
}  
`
```

6.4. Obtenir un produit par ID

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<ProductDto>): (Similaire au JSON de création)

6.5. Obtenir tous les produits

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<ProductDto>>): (Similaire à la réponse getAllUsers)

6.6. Rechercher des produits

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<ProductDto>>): (Similaire à la réponse getAllProducts)

7. Gestion des offres (AdminController)

7.1. Obtenir toutes les offres

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<OfferDto>>): (Similaire à la réponse getAllOffers de l'agriculteur)

7.2. Obtenir les offres en attente de validation

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<OfferDto>>): (Similaire à la réponse getAllOffers)

7.3. Approuver une offre

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<OfferDto>): (Similaire au JSON d'offre, avec status mis à jour à ACTIVE)

7.4. Rejeter une offre

8. Gestion des commandes (AdminController)

8.1. Obtenir toutes les commandes

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<OrderDto>>): (Similaire à la réponse getAllOrders de l'acheteur)

8.2. Obtenir une commande par ID

8.3. Mettre à jour le statut d'une commande

JSON de Requête: Aucun (le statut est un paramètre de requête)

JSON de Réponse (ApiResponse<OrderDto>): (Similaire au JSON de commande, avec status mis à jour)

8.4. Libérer le séquestre d'une commande

9.1. Créer un reçu d'entrepôt

```
`json
{
  "farmerId": 2,
  "productId": 1,
  "batchNumber": "BATCH-001-TOM",
  "grossWeight": 1000.0,
  "netWeight": 980.0,
  "storageLocation": "Entrepôt Nord",
  "qualityGrade": "A",
  "hashValue": "someCryptographicHash"
}
```

JSON de Réponse (ApiResponse<WarehouseReceiptDto>): (Similaire au JSON de requête, avec id et createdAt)

9.2. Valider un reçu d'entrepôt

JSON de Requête (ValidateWarehouseReceiptRequestDto):

```
`json
{
  "auditorId": 1,
  "inspectionReport": "Rapport d'inspection détaillé...",
  "isValid": true
}
```

JSON de Réponse (ApiResponse<WarehouseReceiptDto>): (Similaire au JSON de reçu, avec validatedBy, validatedAt mis à jour)

9.3. Préparer le minting des tokens

```
`json
```



```
{
  "success": true,
  "message": "Transaction de minting préparée",
  "data": {
    "id": 1,
    "warehouseReceiptId": 1,
    "hederaScheduledTransactionId": "0.0.SCHEDULED_TX_ID",
    "status": "PENDING_SIGNATURE"
  },
  "timestamp": "2025-10-06T..."
}
```

9.4. Signer et exécuter la transaction programmée

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<ScheduledTransactionDto>): (Similaire au JSON ci-dessus, avec status mis à jour à EXECUTED)

9.5. Distribuer les tokens

`json

```
{
  "warehouseReceiptId": 1,
  "farmerAccountId": "0.0.FARMER_ACCOUNT_ID",
  "amountToFarmer": 900.0,
  "platformAccountId": "0.0.PLATFORM_ACCOUNT_ID",
  "amountToPlatform": 80.0
}
```

JSON de Réponse (ApiResponse<String>):

```
`json
{
  "success": true,
  "message": "Tokens distribués avec succès",
  "data": null,
  "timestamp": "2025-10-06T..."
}
`
```

10. Notifications (NotificationController)

10.1. Obtenir les notifications de l'utilisateur

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<PageResponse<NotificationDto>>):

```
`json
{
  "success": true,
  "message": "Notifications récupérées avec succès",
  "data": {
    "content": [
      {
        "id": 1,
        "userId": 2,
        "type": "ORDER_CREATED",
        "title": "Nouvelle commande",
        "message": "Votre commande #201 a été passée."
      }
    ]
  }
}
```

```

    "priority": "MEDIUM",
    "sentAt": "2025-10-06T...",
    "isRead": false
  }
],
"page": 0, "size": 10, "totalElements": 1, "totalPages": 1,
"first": true, "last": true, "empty": false
},
"timestamp": "2025-10-06T..."
}
`

```

10.2. Marquer une notification comme lue

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<NotificationDto>): (Similaire au JSON de notification, avec isRead mis à jour à true)

11. Dashboard Admin (AdminController)

11.1. Obtenir les statistiques du dashboard

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<DashboardStatsDto>):

```

`json
{
  "success": true,
  "message": "Statistiques du dashboard récupérées avec succès",

```

```
"data": {
  "totalUsers": 100,
  "totalFarmers": 50,
  "totalBuyers": 49,
  "totalProducts": 20,
  "totalOffers": 30,
  "pendingOffers": 5,
  "totalOrders": 70,
  "ordersInEscrow": 15,
  "totalRevenue": 150000.00,
  "totalTokensMinted": 100000.00
},
"timestamp": "2025-10-06T..."
}
```

JSON de Requête: Aucun (la date est un paramètre de requête)

JSON de Réponse (ApiResponse<Map<LocalDate, BigDecimal>>):

```
`json
{
  "success": true,
  "message": "Revenu quotidien récupéré avec succès",
  "data": {
    "2025-10-05": 1250.50
  },
  "timestamp": "2025-10-06T..."
}
```

11.3. Revenu mensuel

JSON de Requête: Aucun (l'année et le mois sont des paramètres de requête)

JSON de Réponse (ApiResponse<Map<String, BigDecimal>>):

```
`json
{
  "success": true,
  "message": "Revenu mensuel récupéré avec succès",
  "data": {
    "October 2025": 25000.75
  },
  "timestamp": "2025-10-06T..."
}
```

11.4. Revenu annuel

JSON de Requête: Aucun (l'année est un paramètre de requête)

JSON de Réponse (ApiResponse<Map<String, BigDecimal>>):

```
`json
{
  "success": true,
  "message": "Revenu annuel récupéré avec succès",
  "data": {
    "2025": 150000.00
  },
  "timestamp": "2025-10-06T..."
}
```

11.5. Revenu par catégorie

JSON de Requête: Aucun

JSON de Réponse (ApiResponse<Map<String, BigDecimal>>):

```
`json
{
  "success": true,
  "message": "Revenu par catégorie récupéré avec succès",
  "data": {
    "Fruits": 75000.00,
    "Légumes": 50000.00,
    "Céréales": 25000.00
  },
  "timestamp": "2025-10-06T..."
}
```