API Mobile Burkina - Documentation

Informations Générales

Version: 1.0.0
Auteur: mamadou-yeo
Base URL: /api/carteproductions
Base de données: Microsoft SQL Server

Vue d'ensemble

Cette API permet :

1. La gestion des **cartes de production** avec :
    - Recherche par critères
    - Récupération de toutes les cartes (avec pagination)
    - Mise à jour et distribution des cartes
2. L'**authentification des utilisateurs** avec :
    - Inscription (/auth/register)
    - Connexion (/auth/login) avec génération de JWT
    - Récupération du profil utilisateur connecté (/auth/profile)

Endpoints Authentification

```
User{

nom: string;
prenom: string;
login: string;
mdp: string;
idrole: number;
}
```

Inscription d'un utilisateur

Méthode : POST
URL : http://localhost:2025/api/v1/idcapture/user/register

Description :
Crée un nouvel utilisateur avec un rôle spécifique.

Body (JSON) :


```
{
  "nom": "Yeo",
  "prenom": "Mamadou",
  "login": "mamayeo",
  "mdp": "MonMotDePasse123!",
  "idrole": 1
}
```


Réponse succès (201) :


```
{
  "message": "Utilisateur créé avec succès",
  "user": {
    "idutilisateur": 1,
    "nom": "Yeo",
    "prenom": "Mamadou",
    "login": "mamayeo",
    "uestactif": true,
    "idrole": 1
  }
}
```


Erreurs :

- 500 : Erreur serveur ou login déjà existant

Connexion utilisateur

Méthode : POST
URL : http://localhost:2025/api/v1/idcapture/user/login

Description :
Authentifie un utilisateur et retourne un **JWT**.

Body (JSON) :


```
{
```

```
  "login": "mamayeo",
  "mdp": "MonMotDePasse123!"
}
```

Réponse succès (200) :

```
{
  "message": "Connexion réussie",
  "token": "eyJhbGciOiJIUzI1NiIsInR...",
  "user": {
    "idutilisateur": 1,
    "nom": "Yeo",
    "prenom": "Mamadou",
    "login": "mamayeo",
    "uestactif": true,
    "idrole": 1
  }
}
```

Erreurs :

- 401 : Identifiants invalides ou utilisateur inactif
- 500 : Erreur serveur

Exemple de route protégée avec JWT

Méthode : GET
URL :http://localhost:2025/api/v1/idcapture/user/profile

Description :
Retourne les informations de l’utilisateur connecté.
Headers :

Authorization: Bearer <TOKEN_OBTENUE_LORS_DU_LOGIN>

Réponse succès (200) :

```
{
  "idutilisateur": 1,
```

```
  "nom": "Yeo",
  "prenom": "Mamadou",
  "login": "mamayeo",
  "uestactif": true,
  "idrole": 1
}
```

Erreurs :

- 403 : Token manquant
- 401 : Token invalide ou expiré

Si tu veux, je peux **réorganiser toute ta doc finale** pour avoir **toutes les routes** CarteProduction **+ Authentification + Pagination + Recherche + JWT** dans **un seul document clair prêt à partager**, un peu comme un Swagger JSON simplifié.

Veux-tu que je fasse ça ?

Modèle de Données

Interface CarteProduction

```
interface CarteProduction {
  idcarteprod: number;
  nip?: string;
  unique_code: string;
  nom?: string;
  prenoms?: string;
  nomjeunefille?: string;
  datenaissance?: Date;
  datenaissancestring?: string;
  lieunaissance?: string;
  sexe?: string; // char(1)
  taille?: string;
  datedelivrance?: Date;
  datedelivrancestring?: string;
  lieudelivrance?: string;
  urlphoto?: string;
  photo?: Buffer;
  urlsignature?: string;
```

```
  signature?: Buffer;
  codeadministratif?: string;
  codecentreenrolement?: string;
  codecarte?: string;
  idcentreenrolement?: number;
  dateenregistrement?: Date;
  idlieuresidence?: number;
  lieuresidence?: string;
  profession?: string;
  dateexpiration?: Date;
  dateexpirationstring?: string;
  nomcontact?: string;
  prenomcontact?: string;
  telephonecontact?: string;
  mrzligne1?: string;
  mrzligne2?: string;
  isselected?: boolean;
  isonfile?: boolean;
  isproduced?: boolean;
  dateproduction?: Date;
  isdistributed?: boolean;
  datedistribution?: Date;
  compteur: number;
}
```

Endpoints

1. Récupérer toutes les cartes (avec pagination)

GET http://localhost:2025/api/v1/idcapture/carte/gestall

Description

Récupère la liste de toutes les cartes de production avec pagination.

Paramètres de requête (Query Parameters)

Paramètre	Type	Requis	Description	Défaut
page	number	Non	Numéro de page	1
limit	number	Non	Nombre d'éléments par page	10

Réponse de succès (200)

```
{
  "data": [
    {
```

```
      "idcarteprod": 1,
      "unique_code": "ABC123",
      "nom": "Doe",
      "prenoms": "John",
      // ... autres champs
    }
  ],
  "pagination": {
    "currentPage": 1,
    "totalPages": 5,
    "totalItems": 50,
    "itemsPerPage": 10,
    "hasNextPage": true,
    "hasPrevPage": false
  }
}
```


Réponses d'erreur

- **500**: Erreur serveur

2. Rechercher des cartes par critères

GET http://localhost:2025/api/v1/idcapture/carte/searchs

Description

Recherche des cartes en fonction de critères spécifiques (nom, prénoms, date de naissance).

Paramètres de requête (Query Parameters)

Paramètre	Type	Requis	Description	Format
nom	string	Non	Nom de famille	Recherche partielle
prenoms	string	Non	Prénoms	Recherche partielle
datenaissance	string	Non	Date de naissance	YYYY-MM-DD

Exemple d'appel

GET /api/carteproductions/carte/searchs?
nom=Doe&prenoms=John&datenaissance=1990-01-15

Réponse de succès (200)


```
[
  {
    "idcarteprod": 1,
    "unique_code": "ABC123",
    "nom": "Doe",
    "prenoms": "John",
```

```
    "datenaissance": "1990-01-15T00:00:00.000Z",
    // ... autres champs
  }
]
```

Réponses d'erreur

- **404**: Carte non trouvée
- **500**: Erreur serveur

3. Récupérer une carte par code unique

GET /api/carteproductions/carte/:unique_code

Description

Récupère les détails d'une carte spécifique par son code unique.

Paramètres de chemin (Path Parameters)

Paramètre	Type	Requis	Description
unique_code	string	Oui	Code unique de la carte

Exemple d'appel

GET http://localhost:2025/api/v1/idcapture/carte/ABC123

Réponse de succès (200)

```
{
  "idcarteprod": 1,
  "unique_code": "ABC123",
  "nom": "Doe",
  "prenoms": "John",
  "datenaissance": "1990-01-15T00:00:00.000Z",
  "isdistributed": false,
  "datedistribution": null,
  // ... autres champs
}
```

Réponses d'erreur

- **404**: Carte non trouvée
- **500**: Erreur serveur

4. Marquer une carte comme distribuée

PUT http://localhost:2025/api/v1/idcapture/carte/:unique_code

Description

Met à jour une carte pour la marquer comme distribuée avec la date de distribution actuelle.

Paramètres de chemin (Path Parameters)

Paramètre	Type	Requis	Description
unique_code	string	Oui	Code unique de la carte

Exemple d'appel

PUT http://localhost:2025/api/v1/idcapture/carte/ABC123

Champs mis à jour automatiquement

- isdistributed: true
- datedistribution: Date et heure actuelles

Réponse de succès (200)

```
{
  "message": "Carte mise à jour avec succès",
  "updatedCarte": {
    "idcarteprod": 1,
    "unique_code": "ABC123",
    "nom": "Doe",
    "prenoms": "John",
    "isdistributed": true,
    "datedistribution": "2025-08-27T14:30:00.000Z",
    // ... autres champs
  }
}
```

Réponses d'erreur

- **400**: Le paramètre unique_code est requis
- **404**: Carte non trouvée
- **500**: Erreur serveur

5. Mettre à jour la photo d’une carte

PUT
`http://localhost:2025/api/v1/idcapture/photo/:unique_code`

Description

Met à jour la photo d’une carte en remplaçant l’ancienne par la nouvelle image uploadée.
La nouvelle photo est stockée sur le serveur et son **URL** est enregistrée dans la base de données.

Paramètres de chemin (Path Parameters)

Paramètre	Type	Requis	Description
unique_code	string	Oui	Code unique de la carte

Corps de la requête (Request Body)

Type : `multipart/form-data`

Champ	Type	Requis	Description
`urlphoto`	File	Oui	Nouvelle photo de la carte (JPG/PNG)

Exemple d’appel

```
PUT http://localhost:2025/api/v1/idcapture/carte/ABC123/photo
Content-Type: multipart/form-data
(photo = fichier image à uploader)
```

Champs mis à jour automatiquement

- `urlphoto`: nouvel URL de la photo sur le serveur
- `updatedAt`: date et heure de la mise à jour

Réponse de succès (200)

```
{
  "message": "Photo mise à jour avec succès",
  "updatedCarte": {
    "idcarteprod": 1,
    "unique_code": "ABC123",
    "nom": "Doe",
    "prenoms": "John",
    "urlphoto": "https://mon-serveur/photos/ABC123XYZ.jpg",
    "updatedAt": "2025-09-01T12:15:00.000Z"
  }
```

```
}
```

Réponses d’erreur

- **400** : Le paramètre `unique_code` est requis
- **400** : Aucun fichier photo fourni
- **404** : Carte non trouvée
- **500** : Erreur serveur

Fonctionnalités Techniques

Recherche Intelligente

- **Recherche insensible à la casse** : Utilise COLLATE Latin1_General_CI_AI
- **Recherche partielle** : Utilise l'opérateur LIKE avec des wildcards %
- **Recherche combinée** : Possibilité de combiner plusieurs critères avec AND

Pagination

- **Page par défaut** : 1
- **Limite par défaut** : 10 éléments
- **Métadonnées complètes** : Informations sur la pagination (total, pages, navigation)

Gestion des erreurs

- **Logs détaillés** : Toutes les erreurs sont loggées dans la console
- **Réponses standardisées** : Format JSON cohérent pour toutes les réponses
- **Codes de statut appropriés** : 200, 400, 404, 500

Utilisation

Installation

npm install

Développement

npm run dev

Production

npm run build
npm start

Docker

```
# Build
npm run docker:build

# Run
npm run docker:run

# Docker Compose
npm run docker:compose
```

Exemples d'utilisation

JavaScript/Fetch

```
// Rechercher une carte
const response = await
fetch('http://localhost:2025/api/v1/idcapture/carte/ABC123');
const carte = await response.json();

// Marquer comme distribuée
const updateResponse = await
fetch('http://localhost:2025/api/v1/idcapture/carte/ABC123', {
  method: 'PUT'
});
const result = await updateResponse.json();
```

cURL

```
# Récupérer une carte
curl -X GET "http://localhost:2025/api/v1/idcapture/carte/ABC123"

# Rechercher des cartes
curl -X GET "http://localhost:2025/api/v1/idcapture/carte/searchs?
nom=Doe&prenoms=John"

# Marquer comme distribuée
curl -X PUT "http://localhost:2025/api/v1/idcapture/carte/ABC123"
```

Notes importantes

1. **Format des dates** : Les dates en entrée doivent être au format ISO (YYYY-MM-DD)
2. **Codes uniques** : Les unique_code doivent être uniques dans la base de données
3. **Recherche** : La recherche par nom/prénoms est insensible à la casse et permet la recherche partielle

4. **Distribution** : Une fois qu'une carte est marquée comme distribuée, l'opération n'est pas réversible via cette API
5. **Pagination** : Pour de meilleures performances, utilisez la pagination lors de la récupération de toutes les cartes

Contact et Support

Auteur : mamadou-yeo
Version : 1.0.0