Contents

Modules

app

Configure les middlewares, les routes, la gestion des erreurs et les options CORS pour l'API Ticket Bus CM.

controllers/adminController

controllers/authController

authController.js

controllers/busController

controllers/paymentController

controllers/reservationController

Functions

Typedefs

app

Configure les middlewares, les routes, la gestion des erreurs et les options CORS pour l'API Ticket Bus CM.

Brief: Fichier principal de configuration de l'application Express pour Ticket Bus CM.
Routes: - GET / : Test API - /api/auth : Authentification - /api/bus : Gestion des bus - /api/terminaux : Gestion des terminaux - /api/payments : Paiements - /api/tickets : Tickets - /api/admin : Administration - /api/user : utilisateur
Date: 2024-06-01
See

Version: 1.0
Author: UV PROJET CODE Team

controllers/adminController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/adminController

Kind: inner method of controllers/adminController
Route: GET /api/admin/dashboard

Liste tous les bus.

Kind: inner method of controllers/adminController
Returns: Array - Liste des bus
Route: GET /api/admin/buses

Kind: inner method of controllers/adminController
Route: POST /api/admin/buses

Param	Type	Description
compagnie	string	Compagnie du bus
type_bus	string	Type de bus

Kind: inner method of controllers/adminController
Route: PUT /api/admin/buses/:id

Param	Type	Description
id	string	Identifiant du bus

Supprime un bus.

Kind: inner method of controllers/adminController
Route: DELETE /api/admin/buses/:id

Param	Type	Description
id	string	Identifiant du bus

Liste tous les terminaux.

Kind: inner method of controllers/adminController
Returns: Array - Liste des terminaux
Route: GET /api/admin/terminals

Kind: inner method of controllers/adminController
Route: POST /api/admin/terminals

Param	Type	Description
nom	string	Nom du terminal
ville	string	Ville du terminal
adresse	string	Adresse du terminal

Kind: inner method of controllers/adminController
Route: PUT /api/admin/terminals/:id

Param	Type	Description
id	string	Identifiant du terminal

Supprime un terminal.

Kind: inner method of controllers/adminController
Route: DELETE /api/admin/terminals/:id

Param	Type	Description
id	string	Identifiant du terminal

Liste tous les horaires.

Kind: inner method of controllers/adminController
Returns: Array - Liste des horaires

Kind: inner method of controllers/adminController

Param	Type	Description
prix	number	Prix du trajet
places_disponibles	number	Nombre de places disponibles

Kind: inner method of controllers/adminController

Param	Type	Description
id	string	Identifiant de l'horaire

Supprime un horaire.

Kind: inner method of controllers/adminController

Param	Type	Description
id	string	Identifiant de l'horaire

Kind: inner method of controllers/adminController
Route: GET /api/admin/reservations

Kind: inner method of controllers/adminController
Route: PUT /api/admin/reservations/:id

Param	Type	Description
statut	string	Nouveau statut

Kind: inner method of controllers/adminController
Returns: Array - Liste des paiements
Route: GET /api/admin/payments

Liste tous les tickets.

Kind: inner method of controllers/adminController
Returns: Array - Liste des tickets

Route: GET /api/admin/tickets

Liste tous les utilisateurs (clients).

Kind: inner method of controllers/adminController
Returns: Array - Liste des utilisateurs
Route: GET /api/admin/users

Kind: inner method of controllers/adminController
Route: PUT /api/admin/users/:id

Param	Type	Description
id	string	Identifiant de l'utilisateur
nom	string	Nom (optionnel)
email	string	Email (optionnel)
type	string	Type d'utilisateur (optionnel)

Kind: inner method of controllers/adminController
Route: DELETE /api/admin/users/:id

Param	Type	Description
id	string	Identifiant de l'utilisateur

Kind: inner method of controllers/adminController
Route: POST /api/admin/messages

Param	Type	Description
to	Array.<string>	Destinataires
subject	string	Sujet du message
body	string	Contenu du message

Kind: inner method of controllers/adminController
Returns: Array - Liste des messages
Route: GET /api/admin/messages

Kind: inner method of controllers/adminController
Returns: Array - Liste des messages
Route: GET /api/admin/messages/user/:userId

Param	Type	Description
userId	string	Identifiant de l'utilisateur

Marque un message comme lu pour un utilisateur.

Kind: inner method of controllers/adminController
Returns: Object - Message de confirmation
Route: POST /api/admin/messages/:id/read

Param	Type	Description
id	string	Identifiant du message
userId	string	Identifiant de l'utilisateur

Kind: inner method of controllers/adminController
Route: GET /api/admin/inbox

controllers/authController

authController.js

- controllers/authController

Inscrit un nouvel utilisateur.

Kind: static method of controllers/authController
Route: POST /api/auth/register

Param	Type	Description
nom	string	Nom de l'utilisateur
email	string	Email de l'utilisateur
mot_de_passe	string	Mot de passe
type	string	Type d'utilisateur (admin, client, etc.)

Connecte un utilisateur existant.

Kind: static method of controllers/authController
Returns: Object - Token JWT et informations utilisateur ou message d'erreur
Route: POST /api/auth/login

Param	Type	Description
email	string	Email de l'utilisateur
mot_de_passe	string	Mot de passe

Kind: static method of controllers/authController
Route: PUT /api/auth/profile

Param	Type	Description
nom	string	Nouveau nom (optionnel)
email	string	Nouvel email (optionnel)
photo	string	Nouvelle photo (optionnel)

Demande l'envoi d'un OTP pour l'inscription.

Kind: static method of controllers/authController
Route: POST /api/auth/request-otp

Param	Type	Description

Kind: static method of controllers/authController
Route: POST /api/auth/verify-otp

Param	Type	Description
nom	string	Nom de l'utilisateur
mot_de_passe	string	Mot de passe

controllers/busController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/busController

Kind: static method of controllers/busController
Brief: Ajoute un nouveau bus.
Route: POST /api/bus

Param	Type	Description
compagnie	string	Compagnie du bus

Param	Type	Description
type_bus	string	Type de bus
status	string	Statut du bus (optionnel)

Kind: static method of controllers/busController
Route: PUT /api/bus/:id

Param	Type	Description

Kind: static method of controllers/busController
Returns: Array - Liste des bus
Brief: Liste tous les bus.
Route: GET /api/bus

controllers/paymentController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/paymentController

Deprecated

Kind: static method of controllers/paymentController

Param	Type	Description

Example

```
processPayment(req, res);
```

Kind: static method of controllers/paymentController

Param	Type	Description

Example

```
handleNotchPayWebhook(req, res);
```

Kind: static method of controllers/paymentController

Param	Type	Description

Example

```
initiatePayment(req, res);
```

controllers/reservationController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/reservationController

Kind: static method of controllers/reservationController

Param	Type	Description

Example

```
creerReservation(req, res);
```

Kind: static method of controllers/reservationController

Param	Type	Description

Example

```
mesReservations(req, res);
```

Kind: static method of controllers/reservationController

Param	Type	Description

Example

```
getReservedSeats(req, res);
```

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

Param	Type	Description

Example

```
creerHoraire(req, res);
```

Param	Type	Description

Example

```
listerHoraires(req, res);
```

Param	Type	Description

Example

```
recupererHoraire(req, res);
```

controllers/terminalController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/terminalController

Kind: static method of controllers/terminalController
Brief: Ajoute un nouveau terminal.
Route: POST /api/terminaux

Param	Type	Description
ville_destination	string	Ville de destination
terminal_info	string	Informations sur le terminal

Kind: static method of controllers/terminalController
Route: PUT /api/terminaux/:id

Param	Type	Description

Kind: static method of controllers/terminalController
Returns: Array - Liste des terminaux

Brief: Liste tous les terminaux.
Route: GET /api/terminaux

controllers/ticketController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/ticketController

Kind: static method of controllers/ticketController

Param	Type	Description

Example

```
getTicketByReservationId(req, res);
```

Kind: static method of controllers/ticketController

Param	Type	Description

Example

```
getReservedSeats(req, res);
```

controllers/userController

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

- controllers/userController

Kind: static method of controllers/userController
Route: GET /api/user/messages

Kind: static method of controllers/userController
Route: DELETE /api/user/messages/:id

Param	Type	Description

Kind: static method of controllers/userController
Route: POST /api/user/reply-admin

Param	Type	Description
subject	string	Sujet du message
body	string	Contenu du message

routes/admin

Date: 2024-06-01
Route: GET /api/admin/dashboard Statistiques du dashboard
Route: GET/POST/PUT/DELETE /api/admin/buses Gestion des bus
Route: GET/POST/PUT/DELETE /api/admin/terminals Gestion des terminaux
Route: GET /api/admin/payments Gestion des paiements
Route: GET /api/admin/tickets Gestion des tickets
Route: GET/PUT/DELETE /api/admin/users Gestion des utilisateurs
Route: POST/GET/PUT /api/admin/messages Gestion des messages
Version: 1.0
Author: UV PROJET CODE Team

routes/auth

Date: 2024-06-01
Route: POST /api/auth/login Connexion utilisateur
Version: 1.0
Author: UV PROJET CODE Team

routes/bus

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

routes/payment

Brief: Routes pour l'initiation des paiements et la gestion des webhooks de paiement.
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

routes/reservation

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

routes/ticket

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

routes/user

Brief: Routes utilisateur pour la gestion des messages et la communication avec l'admin.
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

server

Date: 2024-06-01
See

- app.js pour la configuration de l'application

Version: 1.0
Author: UV PROJET CODE Team

Kind: global function
Throws:

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Example

```
// Depuis la ligne de commande :
node createAdmin.js
```

Kind: global function
Returns: Promise.<void> - Passe au middleware suivant si l'utilisateur est admin
Throws:

- Error Si le token est manquant, invalide ou si l'utilisateur n'est pas admin

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

Param	Type	Description
next	function	Fonction de rappel pour passer au middleware suivant

Example

Kind: global function
Throws:

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

Param	Type	Description
next	function	Fonction de rappel pour passer au middleware suivant

Example

```
router.get('/profile', authMiddleware, controller.getProfile);
```

Kind: global function
Throws:

Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team

Param	Type	Description
email	string	Adresse email du destinataire

Example

```
await sendOtpMail('user@mail.com', '123456');
```

Bus : Object

Bus.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
compagnie	String	Compagnie du bus
status	String	Statut du bus (actif, inactif)

Example

```
const bus = new Bus({ numero: 'BUS001', capacite: 50, compagnie: 'Express', type_bus: 'VIP'
```

EmailVerification : Object

EmailVerification.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
expiresAt	Date	Date d'expiration de l'OTP

Example

```
const verif = new EmailVerification({ email: 'test@mail.com', otp: '123456', expiresAt: new
```

Message : Object

Message.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
to	Array.<ObjectId>	Destinataires du message
subject	String	Sujet du message
body	String	Contenu du message
sentAt	Date	Date d'envoi
readBy	Array.<ObjectId>	utilisateurs ayant lu le message

Example

```
const msg = new Message({ to: [userId], from: adminId, subject: 'Info', body: 'Bienvenue !'
```

Payment : Object

Payment.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
moyen	String	Moyen de paiement (Mobile Money, NotchPay, Test)
transaction_id	String	Identifiant de la transaction

Example

Reservation : Object

Reservation.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description

Example

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
prix	Number	Prix du trajet
places_disponibles	Number	Nombre de places restantes

Example

Terminal : Object

Terminal.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
nom	String	Nom du terminal
ville	String	Ville du terminal
adresse	String	Adresse du terminal

Example

Ticket : Object

Ticket.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description

Example

User : Object

User.js

Kind: global typedef
Date: 2024-06-01
Version: 1.0
Author: UV PROJET CODE Team
Properties

Name	Type	Description
nom	String	Nom de l'utilisateur
email	String	Email unique
type	String	Type d'utilisateur (client, admin)
photo	String	Photo de profil (optionnelle)

Example

```
const user = new User({ nom: 'Jean', email: 'jean@mail.com', mot_de_passe: '...', type: 'cli
```