

JenCity API Documentation

Documentation complète des API pour l'application JenCity

Généré le 15/03/2025 à 20:31:58

Information du Serveur

URL de Base : <http://localhost:3000/api>

Users

POST /api/users/login

Description: Authentifie un utilisateur existant.

Corps de la Requête:

```
{
  "email": "john@example.com",
  "password": "securepassword123"
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "user": {
    "id": 1,
    "name": "John Doe",
    "email": "john@example.com",
    "role": "user",
    "createdAt": "2023-06-15T10:30:00Z"
  }
}
```

POST /api/users/register

Description: Enregistre un nouvel utilisateur.

Corps de la Requête:

```
{
  "name": "New User",
  "email": "newuser@example.com",
  "password": "securepassword123"
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "user": {
    "id": 3,
    "name": "New User",
    "email": "newuser@example.com",
  }
}
```

```
    "role": "user",
    "createdAt": "2023-08-20T15:45:00Z"
  }
}
```

GET /api/users

Description: Récupère la liste de tous les utilisateurs avec leurs informations de base comme le nom, l'email et le rôle.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "name": "John Doe",
      "email": "john@example.com",
      "role": "user"
    },
    {
      "id": 2,
      "name": "Jane Smith",
      "email": "jane@example.com",
      "role": "admin"
    }
  ]
}
```

GET /api/users/:id

Description: Récupère les informations détaillées d'un utilisateur en fonction de son identifiant unique.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'utilisateur

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "name": "John Doe",
    "email": "john@example.com",
    "role": "user",
    "createdAt": "2023-06-15T10:30:00Z"
  }
}
```

POST /api/users

Description: Crée un nouvel utilisateur dans la base de données avec un rôle par défaut "user".

Corps de la Requête:

```
{
  "name": "New User",
  "email": "newuser@example.com",
  "password": "securepassword123"
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "name": "New User",
    "email": "newuser@example.com",
    "role": "user",
    "createdAt": "2023-08-20T15:45:00Z"
  }
}
```

PUT /api/users/:id

Description: Modifie les informations d'un utilisateur existant comme le nom et l'email.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'utilisateur

Corps de la Requête:

```
{
  "name": "Updated Name",
  "email": "updated@example.com"
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "name": "Updated Name",
    "email": "updated@example.com",
    "role": "user",
    "updatedAt": "2023-08-21T09:12:00Z"
  }
}
```

DELETE /api/users/:id

Description: Supprime définitivement un utilisateur de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'utilisateur

Réponse:

Status: 204

```
{
  "status": 204,
  "message": "User deleted successfully"
}
```

Places

GET /api/places

Description: Récupère la liste de tous les lieux touristiques disponibles incluant leur nom, type, emplacement et description.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "name": "Bulla Regia",
      "type": "historical",
      "location": {
        "latitude": 36.5594,
        "longitude": 8.7553
      },
      "description": "Ancient Roman city with unique underground villas"
    },
    {
      "id": 2,
      "name": "Ain Draham",
      "type": "natural",
      "location": {
        "latitude": 36.7828,
        "longitude": 8.6875
      },
      "description": "Mountain town known for its forests and cool climate"
    }
  ]
}
```

GET /api/places/:id

Description: Récupère les informations détaillées d'un lieu touristique incluant les images, horaires, tarifs et autres informations pratiques.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du lieu

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "name": "Bulla Regia",
    "type": "historical",
    "location": {
      "latitude": 36.5594,
      "longitude": 8.7553
    },
    "description": "Ancient Roman city with unique underground villas",
    "images": [
      "https://example.com/bulla-regia-1.jpg",
      "https://example.com/bulla-regia-2.jpg"
    ],
    "openingHours": {
      "monday": "9:00 - 17:00",
      "tuesday": "9:00 - 17:00",
      "wednesday": "9:00 - 17:00",
      "thursday": "9:00 - 17:00",
      "friday": "9:00 - 17:00",
      "saturday": "9:00 - 17:00",
      "sunday": "9:00 - 17:00"
    },
    "entranceFee": {
      "adult": 10,
      "child": 5,
      "student": 7,
      "senior": 7
    }
  }
}
```

```
}
```

POST /api/places

Description: Crée un nouveau lieu touristique dans la base de données avec ses informations de base.

Corps de la Requête:

```
{
  "name": "New Place",
  "type": "cultural",
  "location": {
    "latitude": 36.5,
    "longitude": 8.8
  },
  "description": "A new cultural attraction"
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "name": "New Place",
    "type": "cultural",
    "location": {
      "latitude": 36.5,
      "longitude": 8.8
    },
    "description": "A new cultural attraction",
    "createdAt": "2023-08-22T11:30:00Z"
  }
}
```

PUT /api/places/:id

Description: Met à jour les informations d'un lieu touristique existant.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du lieu

Corps de la Requête:

```
{
  "name": "Updated Place",
  "type": "historical",
  "description": "Updated description for this attraction"
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "name": "Updated Place",
    "type": "historical",
    "location": {
      "latitude": 36.5594,
      "longitude": 8.7553
    },
    "description": "Updated description for this attraction",
    "updatedAt": "2023-08-23T14:20:00Z"
  }
}
```

```
}
```

DELETE /api/places/:id

Description: Supprime un lieu touristique de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du lieu

Réponse:

Status: 204

```
{
  "status": 204,
  "message": "Place deleted successfully"
}
```

Events

GET /api/events

Description: Récupère la liste de tous les événements programmés incluant leur titre, description, dates et emplacement.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "title": "Cultural Festival",
      "description": "Annual cultural festival in Jendouba",
      "startDate": "2023-09-15T18:00:00Z",
      "endDate": "2023-09-17T22:00:00Z",
      "location": "Jendouba City Center"
    },
    {
      "id": 2,
      "title": "Hiking Trip",
      "description": "Guided hiking trip in Ain Draham mountains",
      "startDate": "2023-10-05T09:00:00Z",
      "endDate": "2023-10-05T16:00:00Z",
      "location": "Ain Draham"
    }
  ]
}
```

GET /api/events/:id

Description: Récupère les détails d'un événement spécifique.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'événement

Réponse:

Status: 200

```
{
  "status": 200,
```

```
"data": {
  "id": 1,
  "title": "Cultural Festival",
  "description": "Annual cultural festival in Jendouba with local artists and
traditional food",
  "startDate": "2023-09-15T18:00:00Z",
  "endDate": "2023-09-17T22:00:00Z",
  "location": "Jendouba City Center",
  "organizer": "Jendouba Cultural Association",
  "ticketPrice": 10,
  "capacity": 500,
  "images": [
    "https://example.com/festival1.jpg",
    "https://example.com/festival2.jpg"
  ]
}
```

POST /api/events

Description: Crée un nouvel événement dans la base de données.

Corps de la Requête:

```
{
  "title": "New Event",
  "description": "Description of the new event",
  "startDate": "2023-11-10T19:00:00Z",
  "endDate": "2023-11-10T23:00:00Z",
  "location": "Bulla Regia Site"
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "title": "New Event",
    "description": "Description of the new event",
    "startDate": "2023-11-10T19:00:00Z",
    "endDate": "2023-11-10T23:00:00Z",
    "location": "Bulla Regia Site",
    "createdAt": "2023-08-24T10:15:00Z"
  }
}
```

PUT /api/events/:id

Description: Met à jour les informations d'un événement existant.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'événement

Corps de la Requête:

```
{
  "title": "Updated Event Title",
  "description": "Updated event description",
  "location": "New location"
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
```

```
    "id": 1,
    "title": "Updated Event Title",
    "description": "Updated event description",
    "startDate": "2023-09-15T18:00:00Z",
    "endDate": "2023-09-17T22:00:00Z",
    "location": "New location",
    "updatedAt": "2023-08-25T11:20:00Z"
  }
}
```

DELETE /api/events/:id

Description: Supprime un événement de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'événement

Réponse:

Status: 204

```
{
  "status": 204,
  "message": "Event deleted successfully"
}
```

Sessions

GET /api/sessions

Description: Récupère la liste de toutes les sessions de messages d'un utilisateur.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "userId1": 1,
      "userId2": 2,
      "lastMessageAt": "2023-08-15T14:35:00Z",
      "isActive": true,
      "createdAt": "2023-08-15T14:30:00Z",
      "user": {
        "id": 2,
        "name": "Jane Smith",
        "email": "jane@example.com"
      },
      "lastMessage": {
        "content": "Bien sûr, comment puis-je vous aider?",
        "senderId": 2,
        "createdAt": "2023-08-15T14:35:00Z"
      }
    },
    {
      "id": 2,
      "userId1": 1,
      "userId2": 3,
      "lastMessageAt": "2023-08-26T16:40:00Z",
      "isActive": true,
      "createdAt": "2023-08-26T16:40:00Z",
      "user": {
        "id": 3,
        "name": "New User",
        "email": "newuser@example.com"
      }
    }
  ]
}
```



```

    "lastMessage": {
      "content": "Je voudrais plus d'informations sur l'événement du 15 septembre.",
      "senderId": 1,
      "createdAt": "2023-08-26T16:40:00Z"
    }
  ]
}

```

GET /api/sessions/:id

Description: Récupère les détails d'une session de messages spécifique.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la session

Réponse:

Status: 200

```

{
  "status": 200,
  "data": {
    "id": 1,
    "userId1": 1,
    "userId2": 2,
    "lastMessageAt": "2023-08-15T14:35:00Z",
    "isActive": true,
    "createdAt": "2023-08-15T14:30:00Z",
    "user1": {
      "id": 1,
      "name": "John Doe",
      "email": "john@example.com"
    },
    "user2": {
      "id": 2,
      "name": "Jane Smith",
      "email": "jane@example.com"
    },
    "messages": [
      {
        "id": 1,
        "sessionId": 1,
        "senderId": 1,
        "content": "Bonjour, j'ai une question sur Bulla Regia.",
        "createdAt": "2023-08-15T14:30:00Z",
        "read": true
      },
      {
        "id": 2,
        "sessionId": 1,
        "senderId": 2,
        "content": "Bien sûr, comment puis-je vous aider?",
        "createdAt": "2023-08-15T14:35:00Z",
        "read": false
      }
    ]
  }
}

```

POST /api/sessions

Description: Crée une nouvelle session de messages entre deux utilisateurs.

Corps de la Requête:

```

{
  "userId1": 1,
  "userId2": 3
}

```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "userId1": 1,
    "userId2": 3,
    "lastMessageAt": null,
    "isActive": true,
    "createdAt": "2023-08-27T09:45:00Z"
  }
}
```

PUT /api/sessions/:id

Description: Met à jour les informations d'une session de messages.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la session

Corps de la Requête:

```
{
  "isActive": false
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "userId1": 1,
    "userId2": 2,
    "lastMessageAt": "2023-08-15T14:35:00Z",
    "isActive": false,
    "updatedAt": "2023-08-27T14:10:00Z"
  }
}
```

DELETE /api/sessions/:id

Description: Supprime une session de messages de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la session

Réponse:

Status: 204

```
{
  "status": 204,
  "message": "Session deleted successfully"
}
```

Messages

GET /api/messages

Description: Récupère la liste de tous les messages envoyés dans l'application JenCity.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "sessionId": 1,
      "senderId": 1,
      "content": "Bonjour, j'ai une question sur Bulla Regia.",
      "createdAt": "2023-08-15T14:30:00Z",
      "read": true
    },
    {
      "id": 2,
      "sessionId": 1,
      "senderId": 2,
      "content": "Bien sûr, comment puis-je vous aider?",
      "createdAt": "2023-08-15T14:35:00Z",
      "read": false
    }
  ]
}
```

GET /api/messages/:id

Description: Récupère un message spécifique par son identifiant.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du message

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "sessionId": 1,
    "senderId": 1,
    "content": "Bonjour, j'ai une question sur Bulla Regia.",
    "createdAt": "2023-08-15T14:30:00Z",
    "read": true
  }
}
```

POST /api/messages

Description: Envoie un nouveau message à un utilisateur.

Corps de la Requête:

```
{
  "sessionId": 1,
  "senderId": 1,
  "content": "Je voudrais plus d'informations sur l'événement du 15 septembre."
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
```

```
    "id": 3,
    "sessionId": 1,
    "senderId": 1,
    "content": "Je voudrais plus d'informations sur l'événement du 15 septembre.",
    "createdAt": "2023-08-26T16:40:00Z",
    "read": false
  }
}
```

PUT /api/messages/:id

Description: Met à jour le statut de lecture d'un message.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du message

Corps de la Requête:

```
{
  "read": true
}
```

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 2,
    "sessionId": 1,
    "senderId": 2,
    "content": "Bien sûr, comment puis-je vous aider?",
    "createdAt": "2023-08-15T14:35:00Z",
    "read": true,
    "updatedAt": "2023-08-26T17:05:00Z"
  }
}
```

DELETE /api/messages/:id

Description: Supprime un message de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique du message

Réponse:

Status: 204

```
{
  "status": 204,
  "message": "Message deleted successfully"
}
```

Reviews

GET /api/reviews

Description: Récupère la liste de toutes les avis et évaluations sur les lieux touristiques.

Réponse:

Status: 200

```
{
  "status": 200,
  "data": [
    {
      "id": 1,
      "userId": 1,
      "placeId": 1,
      "rating": 5,
      "comment": "Un site historique fascinant, je recommande vivement!",
      "createdAt": "2023-07-20T10:15:00Z"
    },
    {
      "id": 2,
      "userId": 2,
      "placeId": 2,
      "rating": 4,
      "comment": "Très beau paysage, mais un peu difficile d'accès.",
      "createdAt": "2023-08-05T16:30:00Z"
    }
  ]
}
```

GET /api/reviews/:id

Description: Récupère un avis spécifique par son identifiant.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'avis

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "userId": 1,
    "placeId": 1,
    "rating": 5,
    "comment": "Un site historique fascinant, je recommande vivement!",
    "createdAt": "2023-07-20T10:15:00Z"
  }
}
```

POST /api/reviews

Description: Ajoute un nouvel avis sur un lieu touristique.

Corps de la Requête:

```
{
  "userId": 3,
  "placeId": 1,
  "rating": 4,
  "comment": "Très intéressant, bonne visite guidée."
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "userId": 3,
    "placeId": 1,
    "rating": 4,
    "comment": "Très intéressant, bonne visite guidée.",
    "createdAt": "2023-08-27T09:45:00Z"
  }
}
```

```
}  
}
```

PUT /api/reviews/:id

Description: Modifie un avis existant.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'avis

Corps de la Requête:

```
{  
  "rating": 5,  
  "comment": "J'ai changé d'avis, c'était vraiment excellent!"  
}
```

Réponse:

Status: 200

```
{  
  "status": 200,  
  "data": {  
    "id": 2,  
    "userId": 2,  
    "placeId": 2,  
    "rating": 5,  
    "comment": "J'ai changé d'avis, c'était vraiment excellent!",  
    "createdAt": "2023-08-05T16:30:00Z",  
    "updatedAt": "2023-08-27T14:10:00Z"  
  }  
}
```

DELETE /api/reviews/:id

Description: Supprime un avis de la base de données.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de l'avis

Réponse:

Status: 204

```
{  
  "status": 204,  
  "message": "Review deleted successfully"  
}
```

Reservations

GET /api/reservations

Description: Récupère la liste de toutes les réservations d'événements ou de visites.

Réponse:

Status: 200

```
{  
  "status": 200,  
  "data": [  
    {  

```

```
{
  "id": 1,
  "userId": 1,
  "eventId": 1,
  "numberOfTickets": 2,
  "totalPrice": 20,
  "status": "confirmed",
  "createdAt": "2023-08-10T11:20:00Z"
},
{
  "id": 2,
  "userId": 2,
  "placeId": 1,
  "visitDate": "2023-09-20T10:00:00Z",
  "numberOfPersons": 3,
  "totalPrice": 30,
  "status": "pending",
  "createdAt": "2023-08-15T09:45:00Z"
}
]
```

GET /api/reservations/:id

Description: Récupère les détails d'une réservation spécifique.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la réservation

Réponse:

Status: 200

```
{
  "status": 200,
  "data": {
    "id": 1,
    "userId": 1,
    "eventId": 1,
    "numberOfTickets": 2,
    "totalPrice": 20,
    "status": "confirmed",
    "createdAt": "2023-08-10T11:20:00Z",
    "paymentMethod": "credit card",
    "paymentId": "pay_1234567"
  }
}
```

POST /api/reservations

Description: Crée une nouvelle réservation pour un événement ou une visite.

Corps de la Requête:

```
{
  "userId": 3,
  "eventId": 2,
  "numberOfTickets": 1,
  "paymentMethod": "paypal"
}
```

Réponse:

Status: 201

```
{
  "status": 201,
  "data": {
    "id": 3,
    "userId": 3,
    "eventId": 2,
    "numberOfTickets": 1,
  }
}
```

```
    "totalPrice": 10,  
    "status": "pending",  
    "paymentMethod": "paypal",  
    "createdAt": "2023-08-28T15:30:00Z"  
  }  
}
```

PUT /api/reservations/:id

Description: Met à jour le statut ou les détails d'une réservation.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la réservation

Corps de la Requête:

```
{  
  "status": "confirmed",  
  "paymentId": "pay_9876543"  
}
```

Réponse:

Status: 200

```
{  
  "status": 200,  
  "data": {  
    "id": 2,  
    "userId": 2,  
    "placeId": 1,  
    "visitDate": "2023-09-20T10:00:00Z",  
    "numberOfPersons": 3,  
    "totalPrice": 30,  
    "status": "confirmed",  
    "paymentId": "pay_9876543",  
    "updatedAt": "2023-08-28T16:15:00Z"  
  }  
}
```

DELETE /api/reservations/:id

Description: Annule et supprime une réservation.

Paramètres:

id (number) - Obligatoire: L'identifiant unique de la réservation

Réponse:

Status: 204

```
{  
  "status": 204,  
  "message": "Reservation cancelled and deleted successfully"  
}
```