

# API-Dokumentation für KundenAPI

Die **kundenKartenAPI** ermöglicht die Verwaltung von Kundenkarten über RESTful Endpunkte. Im Folgenden werden die Standardfunktionalitäten der API dokumentiert.

## Basis-URL

Die Basis-URL für alle Endpunkte lautet:

## Endpunkte und Methoden

### GET /api/kundenKarten

Ruft alle vorhandenen Kundenkarten ab.

- Methode: GET
- URI: /api/kundenKarten
- Erfolgreiche Antwort (200 OK):

```
{  
  "Id": 1,  
  "Nachname": "Mustermann",  
  "Vorname": "Max",  
  "Anschrift": "Musterstraße 123",  
  "Bild": "https://example.com/max_mustermann.jpg"  
}
```

- Fehlerantwort (404 Not Found):

```
{  
  "error": "Ressource nicht gefunden."  
}
```

## GET /api/kundenKarten/{id}

Ruft eine spezifische Kundenkarte anhand ihrer ID ab.

- Methode: GET
- URI: /api/kundenKarten/{id}
- Parameter:
  - {id}: Die eindeutige ID der Kundenkarte.
- Erfolgreiche Antwort (200 OK):

```
{  
  "Id": 1,  
  "Name": "Max Mustermann",  
  "Email": "max@example.com"  
}
```

- Fehlerantwort (404 Not Found):

```
{  
  "error": "Ressource nicht gefunden."  
}
```

## PUT /api/kundenKarten/{id}

Aktualisiert eine vorhandene Kundenkarte.

- Methode: PUT
- URI: /api/kundenKarten/{id}
- Parameter:
  - {id}: Die eindeutige ID der zu aktualisierenden Kundenkarte.
- Erfolgreiche Antwort (204 No Content): Keine weiteren Daten.
- Fehlerantworten:
  - 400 Bad Request: Wenn die ID in der Anfrage nicht mit der ID in der Kundenkarte übereinstimmt.
  - 404 Not Found: Wenn die Kundenkarte nicht gefunden wird.

## POST /api/kundenKarten

Erstellt eine neue Kundenkarte.

- Methode: POST
- URI: /api/kundenKarten
- Erfolgreiche Antwort (201 Created):
- 

```
{  
  "Id": 2,  
  "Name": "Neuer Kunde",  
  "Email": "neu@example.com"  
}
```

- Fehlerantwort (500 Internal Server Error):

```
{  
  "error": "Entity set 'kundenContext.kundenKarte' is  
  null."  
}
```

## DELETE /api/kundenKarten/{id}

Löscht eine Kundenkarte anhand ihrer ID.

- Methode: DELETE
- URI: /api/kundenKarten/{id}
- Parameter:
  - {id}: Die eindeutige ID der zu löschenden Kundenkarte.
- Erfolgreiche Antwort (204 No Content): Keine weiteren Daten.
- Fehlerantwort (404 Not Found): Wenn die Kundenkarte nicht gefunden wird.

# Authentifizierung

## Login-Controller

Der Login-Controller ist verantwortlich für die Authentifizierung von Benutzern und die Erstellung von JWT-Token (JSON Web Token) für autorisierte Anfragen. Der Controller ist als Web-API-Endpunkt konzipiert und folgt dem RESTful-Prinzip.

Route:

Endpunkt für die Benutzerauthentifizierung und Token-Erstellung  
HTTP-Methode

### POST /api/Login/Login

```
{  
  "UserName": "demo_user ",  
  "Password": "demo_password123"  
}
```

- Methode: POST
- URI: /api/Login/Login
- Erfolgreiche Antwort (200 OK): JWT-Token
- Fehlerantwort (400 Bad Request): Fehlermeldung
  - "Benutzerdaten nicht eingegeben.": Benutzername oder Passwort fehlen.
  - "Falsches Passwort": Das eingegebene Passwort ist falsch.
  - "Benutzer nicht gefunden": Der Benutzer mit dem angegebenen Benutzernamen wurde nicht gefunden.

## Authentifizierung und Token-Erstellung

1. Prüfung eingegebene Benutzerdaten: Überprüft, ob der Benutzername oder das Passwort leer oder null sind. Falls ja, wird eine Fehlermeldung zurückgegeben.
2. Überprüfung Benutzerdaten in der Datenbank: Sucht in der Datenbank nach einem Benutzer mit dem angegebenen Benutzernamen. Falls vorhanden, wird fortgefahren. Ansonsten wird eine Fehlermeldung zurückgegeben.
3. Überprüfe Passwort: Vergleicht das eingegebene Passwort mit dem in der Datenbank gespeicherten Passwort des Benutzers. Falls übereinstimmt, wird ein JWT-Token erstellt.
4. Token-Erstellung: Erstellt ein JWT-Sicherheitstoken mit den angegebenen Parametern, einschließlich Aussteller, erwartetem Empfänger, Ablaufzeit und Signaturinformationen.

Erfolgreiche Antwort: Gibt das erstellte JWT-Token als OK-Antwort zurück.

## Anmerkungen

Die Authentifizierung erfolgt über den Vergleich von Benutzerdaten in der Datenbank. Falls erfolgreich, wird ein JWT-Token erstellt und dem Benutzer zurückgegeben.

Alle Fehlermeldungen sind darauf ausgerichtet, klare Hinweise auf mögliche Probleme zu geben, um die Benutzerfreundlichkeit zu verbessern und sicherzustellen, dass die Benutzer korrekte Informationen eingeben.