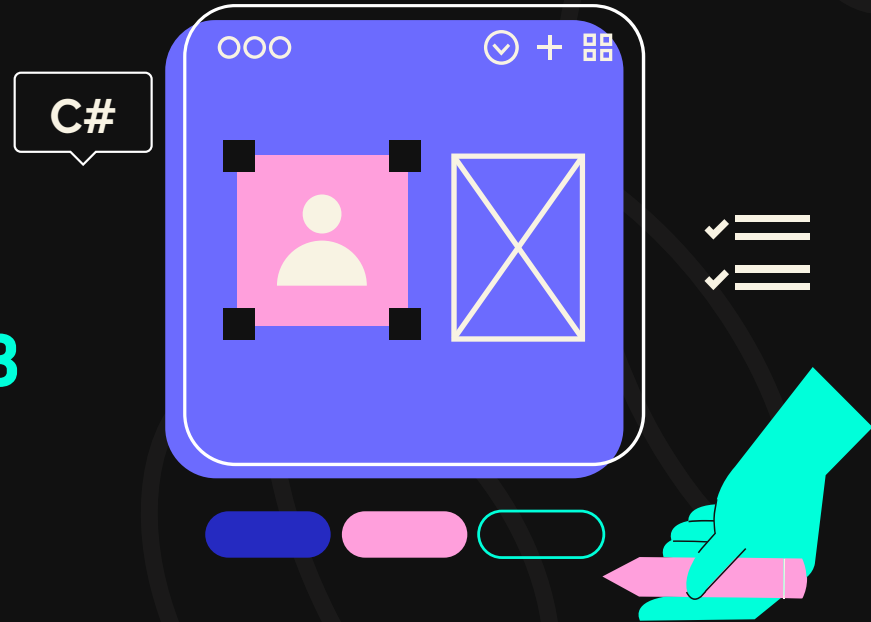


Jetstream-Backend WEB-API | Stand 17.11.23

By Arda Dursun | Template aus «[SlidesGo](#)»



Inhaltsverzeichnis – Seite 1

01

Projektüberblick

02

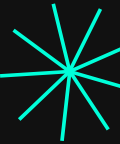
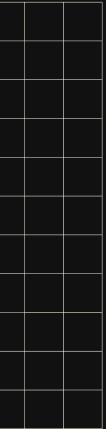
Anforderungsanalyse
Was wurde umgesetzt?

03

Technologiestack

04

Implementierung



Inhaltsverzeichnis – Seite 2

05

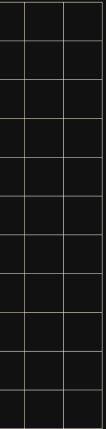
Testing

06

Ergebnis inkl. Code
Stand 17.11.2023

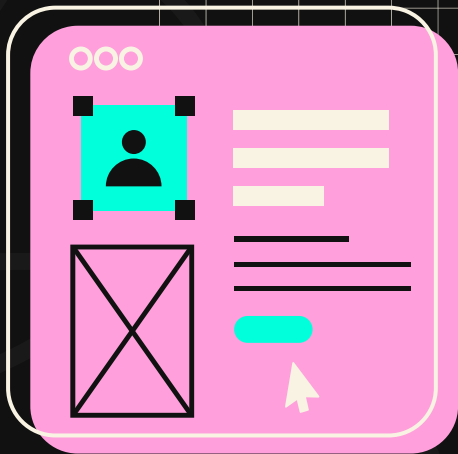
07

Fazit + Fragen



01

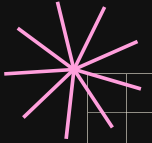
Projektüber- sicht



Projektübersicht

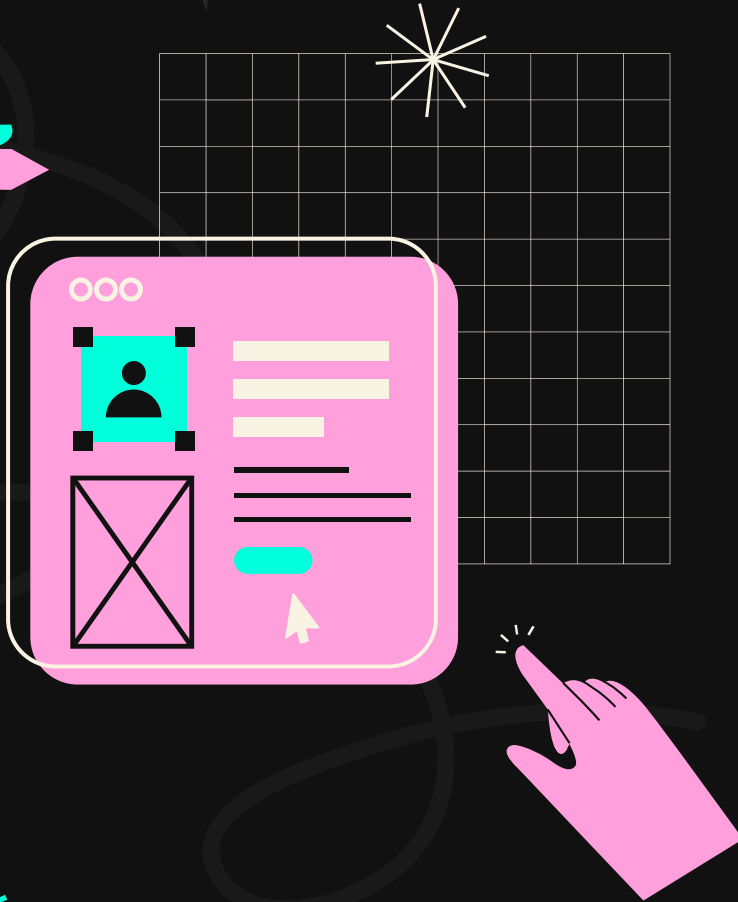


- **Thema:** Entwicklung einer Web-API für das Ski-Service Management
- **Ziel:** Digitalisierung und Optimierung des Serviceprozesses
- **Bezug:** Modul 294



02

Anforderungen

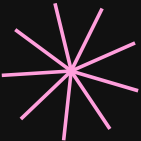


Nicht-Optionale Anforderungen

2.1 Zusammenfassung der Anforderungen

Nr.	Beschreibung
A1	Login Dialog mit Passwort für den autorisierten Zugang der Mitarbeiter (Datenänderungen).
A2	In der Datenbank müssen die Informationen des Serviceauftrags und die Login Daten der Mitarbeiter verwaltet sein.
A3	Erfasste Serviceaufträge abrufbar sein.
A4	Die erfassten Serviceaufträge müssen selektiv nach Priorität abrufbar sein.
A5	Mitarbeiter können eine Statusänderung eines Auftrages vornehmen.
A6	Mitarbeiter können Aufträge löschen (z.B. bei Stornierungen)
A7	Die aufgerufenen API-Endpoints müssen zwecks Fehlerlokalisierung protokolliert sein (DB oder Protokolldatei).
A8	Datenbankstruktur muss normalisiert in der 3.NF sein inkl. referenzieller Integrität
A9	Für die Web-API Applikation muss ein eigener Datenbankbenutzerzugang mit eingeschränkter Berechtigung (DML) zur Verfügung gestellt werden (Benutzer root bzw. sa ist verboten).
A10	Das Web-API muss vollständig nach Open-API (Swagger) dokumentiert sein.
A11	Das Softwareprojekt ist über ein Git-Repository zu verwalten.
A12	Ganzes Projektmanagement muss nach IPERKA dokumentiert sein

</>



Optionale Anforderungen

</>

3 Zusätzliche Anforderungen

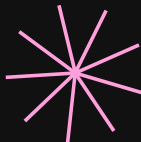
Zusatzpunkte für optionale Erweiterungen. Zur Erreichung der max. Punktzahl müssen zwei optionale Anforderungen umgesetzt werden. Es werden nur zwei zusätzliche Anforderungen bewertet.

Nr.	Beschreibung
AO1	Die Mitarbeiter können zu einem Auftrag einen Freitext bzw. Kommentar hinterlegen
AO2	Ein Auftrag kann mit sämtlichen Datenfeldern geändert werden
AO3	Das Login des Mitarbeiters wird nach drei nachfolgenden Falschanmeldungen automatisch gesperrt.
AO4	Personalisierte Auftragsliste des eingeloggten Mitarbeiters. Der Mitarbeiter kann sich zusätzlich zur gesamten Auftragsliste nur die von ihm übernommenen Aufträge ansehen.
AO6	Eingeloggte Mitarbeiter können ein gesperrtes Login zurücksetzen.
AO7	Gelöscht Aufträge werden nicht aus der Datenbank entfernt, sondern nur als gelöscht markiert.

✓

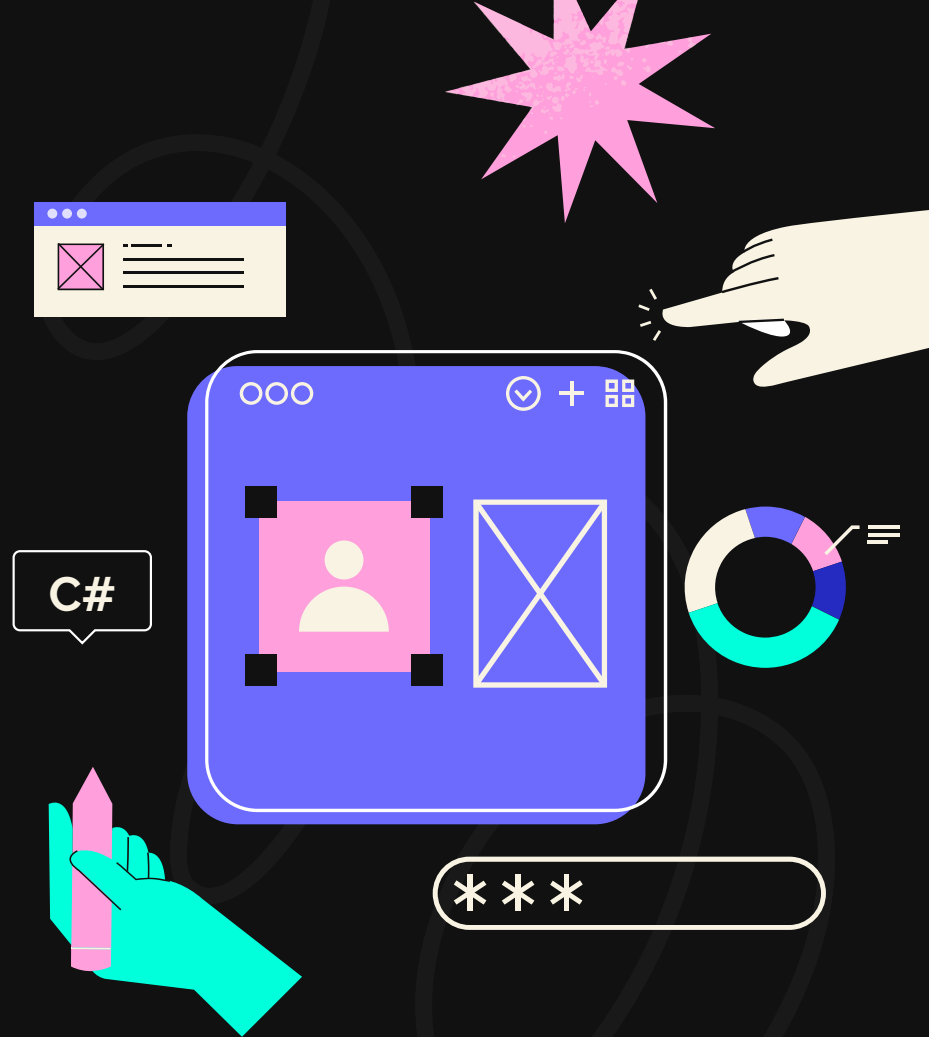
✓

○



03

Techno- logiestack



Technologie Stack

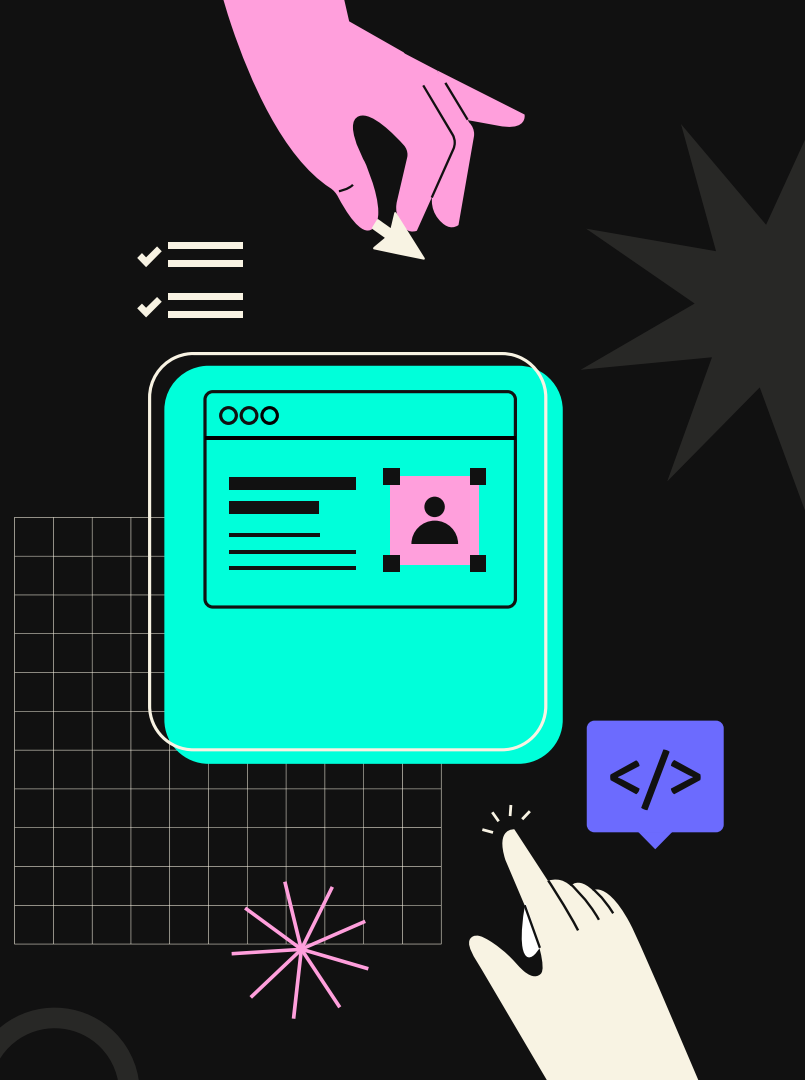
Backend-Technologien:

- Visual Studio
- .NET 7.0
- ASP.NET Core Authentication (JwtBearer)
- Microsoft.AspNetCore.OpenApi
- Entity Framework Core 7.0
- Entity Framework Core SQL Server
- Entity Framework Core Tools
- Microsoft Identity Model Tokens
- Serilog
- Serilog.AspNetCore
- Serilog.Sinks.File
- Swashbuckle.AspNetCore
- System.IdentityModel.Tokens.Jwt



Authentifizierung und Sicherheit:

- Rollenbasierte Zugriffskontrolle
- Implementierung von JWT für sichere Benutzerauthentifizierung



04

Implementierung



Schwerpunkt

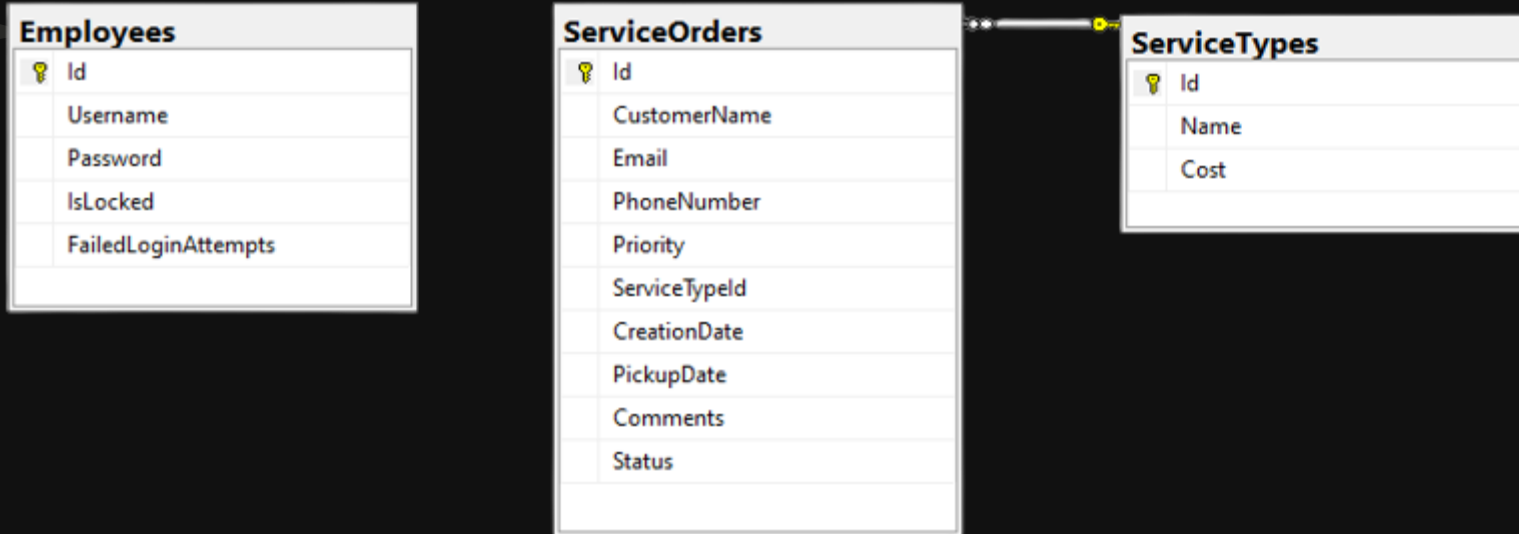
Backend:

- Es wurde nicht viel Wert auf Frontend gesetzt da es für dieses Modul irrelevant ist
- Effiziente Umsetzung

Entity Framework



Datenbankstruktur 3NF



SWAGGER UI

Employees

POST /api/Employees/create

POST /api/Employees/login

POST /api/Employees/unlock/{username}

GET /api/Employees/validate

Registrations

POST /api/Registrations

GET /api/Registrations

DELETE /api/Registrations

GET /api/Registrations/search


PATCH /api/Registrations/{id}

DELETE /api/Registrations/{id}

GET /api/Registrations/ByPriority

PATCH /api/Registrations/{id}/status

LOGIN Interface

 **Jetstream - Service** [Über uns](#) [Online-Anmeldung](#) [Kontakt](#) [Credits](#) [Swagger-UI \(Backend\)](#) [Login](#)

LOGIN

Please enter your credentials to login.

LOGIN

[Not registered? Create an account](#)

Dashboard



Jetstream - Service

[Über uns](#) [Online-Anmeldung](#) [Kontakt](#) [Credits](#) [Swagger-UI \(Backend\)](#)

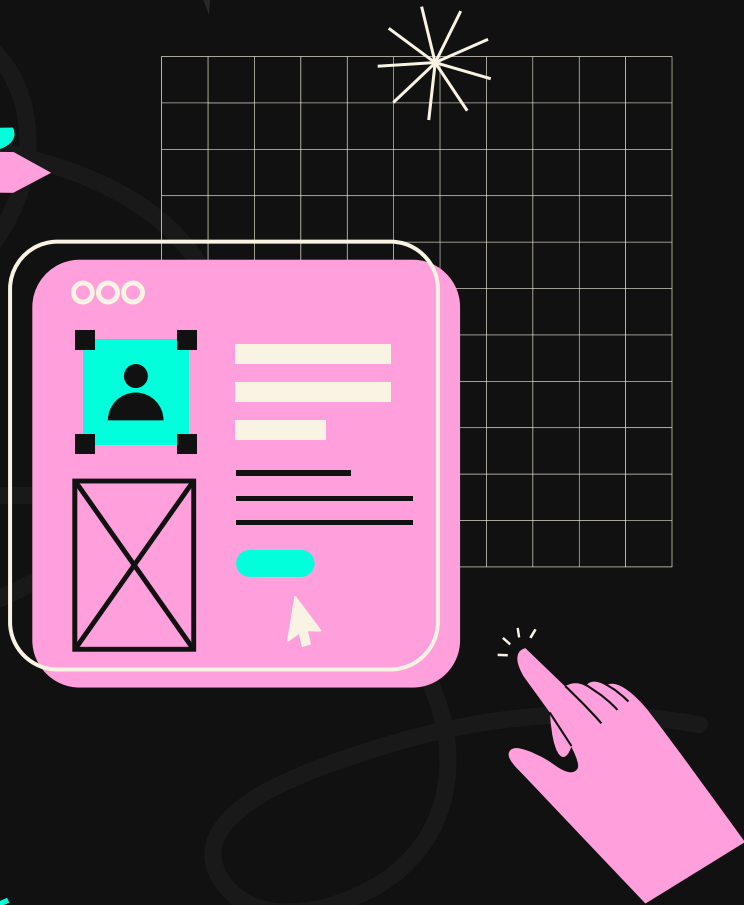
Logout

Service Orders Dashboard

Priorität auswählen:

Alle

#	Customer Name	Email	Phone	Priority	Service Type	Creation Date	Pickup Date	Status	Comments	Actions
11	Müller	lukas@test.com	+41 54864	low	Grosser-Service	16.11.2023	28.11.2023	In Bearbeitung		Edit Löschen
12	Luigi	Mario@test.com	+41 234	express	Rennski-Service	16.11.2023	21.11.2023	Offen	Kunde hat sich beschv	Löschen Save Cancel
13	Donald	Trump@test.com	+41 234	low	Rennski-Service	16.11.2023	28.11.2023	Offen		Edit Löschen



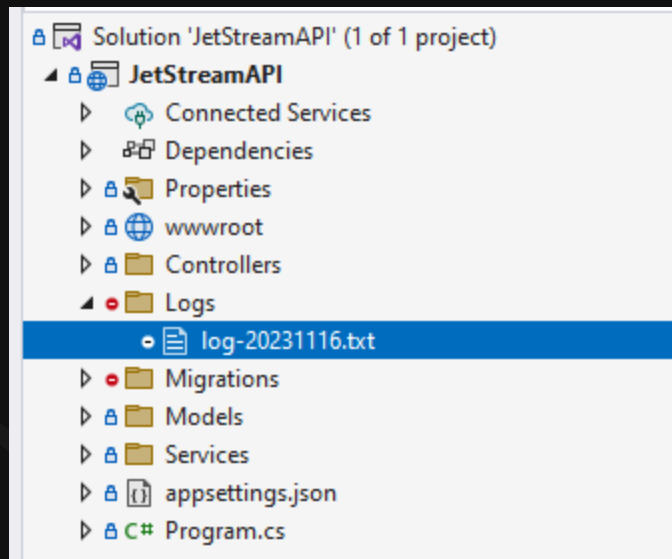
05

Testing

Postman

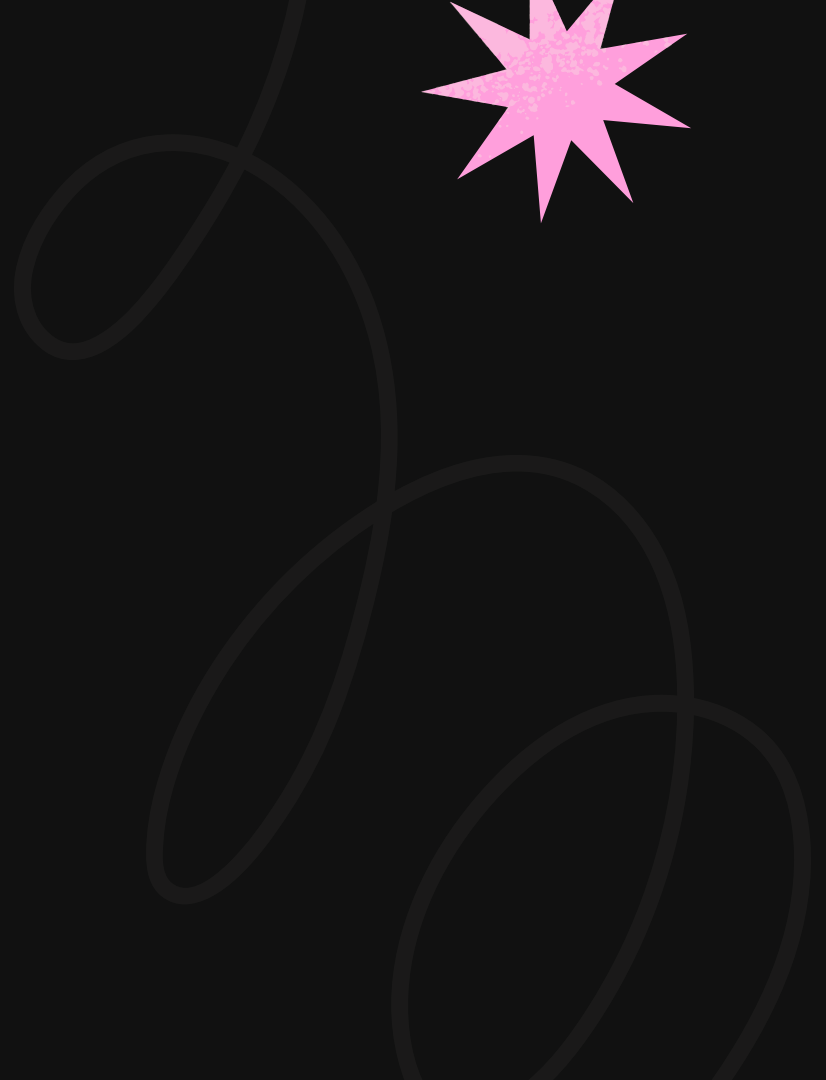
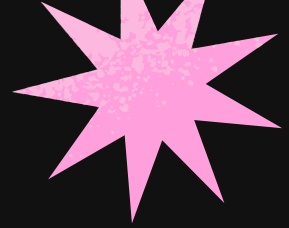
The screenshot displays the Postman application interface. At the top, the navigation bar includes 'Home', 'Workspaces', 'API Network', and 'Explore'. A search bar is present with the text 'Search Postman'. The main workspace is titled 'My Workspace' and shows a list of collections on the left sidebar, including 'JSONPlaceholder', 'Modul 294', 'Modul 295 - Arda', 'PA', 'Orders', and 'Employees'. The 'Orders' collection is expanded, showing a 'DELETE All' endpoint. The main panel shows a 'DELETE' request to 'https://localhost:7092/api/Registrations'. The 'Authorization' tab is active, showing a 'Bearer Token' type. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables'. The token value is 'eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.e...'. The 'Body' tab is selected, showing a status of '204 No Content', 'Time: 170 ms', and 'Size: 81 B'. The bottom panel shows 'No logs yet' and a message to 'Send a request to view its details in the console.' The bottom status bar includes icons for 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Trash', and other utility icons.

Logfiles



06

Live Demo



07 Fazit

- **Herausforderungen und Lessons Learned**
 - Aufgetretene Probleme
 - Optimismus
 - Umgang mit DTO und JWT
 - Immer noch Optimierbar

