API Migration Spezifikation

SPEZIFIKATION · SPEC-2025-001

Projekt:	API Migration Projekt
Auftraggeber:	DataFlow Analytics AG
Version:	1.2
Status:	**REVIEW**
Betreff:	REST API v2.0 - User Management
Erstellt:	2025-01-05
Aktualisiert:	2025-01-18
Autoren:	Max Mustermann, Anna Fischer

API REST MIGRATION BACKEND

Inhaltsverzeichnis

1 Einleitung

1.1 Zweck des Dokuments

Dieses Dokument spezifiziert die Anforderungen und das Design der neuen REST API v2.0 für das User Management System von DataFlow Analytics AG.

1.2 Zielgruppe

- Backend-Entwickler
- Frontend-Entwickler (API-Konsumenten)
- QA-Team
- DevOps-Team

1.3 Änderungshistorie

Version	Datum	Autor	Änderung
1.0	05.01.2025	Max Mustermann	Initiale Version
1.1	12.01.2025	Anna Fischer	Authentication Endpoints hinzugefügt
1.2	18.01.2025	Max Mustermann	Rate Limiting spezifiziert

2 Systemübersicht

2.1 Architektur

Die neue API folgt einer modernen REST-Architektur mit folgenden Komponenten:

- **API Gateway:** NGINX als Reverse Proxy
- **Application Server:** Node.js mit Express.js Framework
- **Database:** PostgreSQL 14
- **Cache:** Redis 7
- **Authentication:** JWT (JSON Web Tokens)

2.2 Technologie-Stack

```
// Package Dependencies
{
  "express": "^4.18.0",
  "jsonwebtoken": "^9.0.0",
  "bcrypt": "^5.1.0",
  "pg": "^8.11.0",
  "redis": "^4.6.0",
  "helmet": "^7.1.0",
```

```
  "express-rate-limit": "^7.1.0"
}
```

3 Funktionale Anforderungen

3.1 User Management

3.1.1 REQ-001: Benutzer erstellen [HIGH]

Beschreibung: Das System muss es Administratoren ermöglichen, neue Benutzer anzulegen.

Endpoint: `POST /api/v2/users`

Request Body:

```
{
  "email": "user@example.com",
  "firstName": "John",
  "lastName": "Doe",
  "role": "user",
  "permissions": ["read", "write"]
}
```

Response (201):

```
{
  "id": "uuid-v4",
  "email": "user@example.com",
  "firstName": "John",
  "lastName": "Doe",
  "role": "user",
  "createdAt": "2025-01-18T10:30:00Z"
}
```

Validierung:
- Email muss gültig und eindeutig sein
- firstName und lastName sind Pflichtfelder (min. 2 Zeichen)
- role muss einem der definierten Rollen entsprechen: `admin`, `user`, `readonly`

3.1.2 REQ-002: Benutzer abrufen [HIGH]

Beschreibung: Abruf eines einzelnen Benutzers anhand der ID

Endpoint: `GET /api/v2/users/{userId}`

Response (200):

```
{
  "id": "uuid-v4",
  "email": "user@example.com",
  "firstName": "John",
  "lastName": "Doe",
  "role": "user",
  "permissions": ["read", "write"],
```

```
  "createdAt": "2025-01-10T10:30:00Z",
  "updatedAt": "2025-01-18T14:20:00Z",
  "lastLogin": "2025-01-18T09:15:00Z"
}
```

3.1.3 REQ-003: Benutzer aktualisieren [MEDIUM]

Endpoint: `PATCH /api/v2/users/{userId}`

Request Body (partial update):

```
{
  "firstName": "Jane",
  "permissions": ["read", "write", "admin"]
}
```

3.1.4 REQ-004: Benutzer löschen [MEDIUM]

Endpoint: `DELETE /api/v2/users/{userId}`

Response (204): No Content

Hinweis: Soft-Delete - Benutzer wird als „deactivated“ markiert, nicht physisch gelöscht

3.1.5 REQ-005: Benutzerliste abrufen [HIGH]

Endpoint: `GET /api/v2/users`

Query Parameters:
- `page` : Seitennummer (default: 1)
- `limit` : Einträge pro Seite (default: 20, max: 100)
- `role` : Filter nach Rolle
- `search` : Volltextsuche in Name/Email
- `sort` : Sortierung (z.B. `createdAt:desc`)

Response (200):

```
{
  "data": [...],
  "pagination": {
    "page": 1,
    "limit": 20,
    "total": 157,
    "pages": 8
  }
}
```

3.2 Authentication & Authorization

3.2.1 REQ-010: Login [HIGH]

Endpoint: `POST /api/v2/auth/login`

Request:

```
{
  "email": "user@example.com",
  "password": "SecurePass123!"
}
```

Response (200):

```
{
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "expiresIn": 3600,
  "user": {
    "id": "uuid",
    "email": "user@example.com",
    "role": "user"
  }
}
```

Sicherheit:
- Passwörter werden mit bcrypt (cost factor 12) gehasht
- Rate Limiting: 5 Versuche pro 15 Minuten pro IP
- Account-Sperre nach 10 fehlgeschlagenen Versuchen

3.2.2 REQ-011: Token Refresh [HIGH]

Endpoint: `POST /api/v2/auth/refresh`

Request:

```
{
  "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

3.2.3 REQ-012: Logout [MEDIUM]

Endpoint: `POST /api/v2/auth/logout`

Funktion: Invalidiert das aktuelle Access Token durch Blacklisting in Redis

4 Nicht-funktionale Anforderungen

4.1 Performance

- **Response Time:** < 200ms für 95% aller Requests (p95)
- **Throughput:** Min. 1000 Requests pro Sekunde
- **Concurrent Users:** Bis zu 10.000 gleichzeitige Nutzer

4.2 Sicherheit

- HTTPS-Only (TLS 1.3)
- CORS-Policy konfiguriert
- Helmet.js für Security Headers

- Input Validation & Sanitization
- SQL Injection Prevention (Prepared Statements)
- XSS Protection

4.3 Rate Limiting

```
// Rate Limiting Konfiguration
const limiter = rateLimit({
  windowMs: 15 * 60 * 1000, // 15 Minuten
  max: 100, // Max 100 Requests pro Window
  message: 'Too many requests from this IP'
});
```

Spezielle Limits:
- Login: 5 Requests / 15 Minuten
- User Creation: 10 Requests / Stunde
- Standard: 100 Requests / 15 Minuten

4.4 Monitoring & Logging

- Strukturiertes Logging (JSON-Format)
- Request/Response Logging
- Error Tracking mit Sentry
- Metrics: Prometheus
- Visualization: Grafana Dashboard

5 API-Dokumentation

5.1 OpenAPI Specification

Die vollständige API wird mit OpenAPI 3.0 dokumentiert und unter `/api/v2/docs` verfügbar gemacht (Swagger UI).

5.2 Versionierung

- URL-basierte Versionierung: `/api/v2/...`
- Alte Version v1 läuft parallel für 6 Monate
- Deprecation Warnings in Response Headers

6 Testing

6.1 Unit Tests

- Code Coverage: Min. 80%
- Framework: Jest

- Alle Business Logic Funktionen

6.2 Integration Tests

- Framework: Supertest
- Alle API Endpoints
- Positive & Negative Test Cases

6.3 Load Tests

- Tool: k6
- Szenarien: 100, 500, 1000 concurrent users
- Ziel: < 200ms Response Time bei 1000 Users

7 Deployment

7.1 Staging

- Automatisches Deployment bei Merge in `develop` Branch
- URL: `https://api-staging.dataflow.com`

7.2 Production

- Blue-Green Deployment Strategie
- URL: `https://api.dataflow.com`
- Rollback-Mechanismus innerhalb 5 Minuten

8 Offene Punkte

- [ ] Entscheidung über OAuth2 Integration für externe Clients
- [ ] Finalisierung der GDPR-konformen User-Daten-Export Funktion
- [ ] Definition der Backup & Recovery Strategie