用户认证 API 文档

基础信息

- **Base URL**: `http://localhost:3000/api`
- **Content-Type**: `application/json`

认证接口

1. 用户注册

端点: `POST /auth/register`

请求体:

```json
{
  "email": "user@example.com",
  "password": "password123",
  "displayName": "用户名"
}
```

字段验证:

- `email`: 必须是有效的电子邮件地址
- `password`: 最少6个字符
- `displayName`: 最少2个字符

成功响应 (201):

```json
{
  "data": {
    "user": {
      "id": "uuid",
      "email": "user@example.com",
      "displayName": "用户名",
      "isActive": true,
      "lastLoginAt": null,
      "createdAt": "2025-10-17T00:00:00.000Z",
      "updatedAt": "2025-10-17T00:00:00.000Z",
      "roles": ["trader"]
    },
    "tokens": {
      "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
      "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
    }
  }
}
```

错误响应:

- **409 Conflict**: 邮箱已被注册
- **400 Bad Request**: 请求参数验证失败

---

2. 用户登录

端点: `POST /auth/login`

请求体:

```json
{
  "email": "user@example.com",
  "password": "password123"
}
```

成功响应 (201):

```json
{
  "data": {
    "user": {
      "id": "uuid",
      "email": "user@example.com",
      "displayName": "用户名",
      "isActive": true,
      "lastLoginAt": "2025-10-17T00:00:00.000Z",
      "createdAt": "2025-10-17T00:00:00.000Z",
      "updatedAt": "2025-10-17T00:00:00.000Z",
      "roles": ["trader"]
    },
    "tokens": {
      "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
      "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
    }
  }
}
```

错误响应:

- **401 Unauthorized**: 邮箱或密码错误

---

3. 刷新访问令牌

端点: `POST /auth/refresh`

请求体:

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

成功响应 (201):

```
{
  "data": {
    "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

错误响应:

- **401 Unauthorized**: Refresh token 无效或已过期

---

4. 获取当前用户信息

端点: `GET /auth/me`

请求头:

```
Authorization: Bearer {accessToken}
```

成功响应 (200):

```
{
  "data": {
    "id": "uuid",
    "email": "user@example.com",
    "displayName": "用户名",
    "isActive": true,
    "lastLoginAt": "2025-10-17T00:00:00.000Z",
    "createdAt": "2025-10-17T00:00:00.000Z",
    "updatedAt": "2025-10-17T00:00:00.000Z",
    "roles": ["trader"]
  }
}
```

错误响应:

- **401 Unauthorized**: Access token 无效或已过期

---

5. 登出

端点: `POST /auth/logout`

请求头:

```
Authorization: Bearer {accessToken}
```

成功响应 (201):

```
{
  "data": {
    "success": true
  }
}
```

说明: 登出后，当前的 refresh token 将被撤销，无法再用于刷新访问令牌。

错误响应:

- **401 Unauthorized**: Access token 无效或已过期

Token 说明

Access Token (访问令牌)

- **有效期**: 15分钟
- **用途**: 访问受保护的API端点
- **使用方式**: 在请求头中添加 `Authorization: Bearer {accessToken}`

Refresh Token (刷新令牌)

- **有效期**: 7天
- **用途**: 获取新的 access token
- **安全建议**:
  - 将 refresh token 安全存储（如 httpOnly cookie 或安全的本地存储）
  - 不要在 URL 或日志中暴露 refresh token

错误处理

所有错误响应都遵循以下格式:

```
{
  "statusCode": 400,
  "timestamp": "2025-10-17T00:00:00.000Z",
  "path": "/api/auth/register",
  "message": {
    "message": "Error description",
    "error": "Error Type",
    "statusCode": 400
  }
}
```

常见HTTP状态码:

- **200**: 成功
- **201**: 创建成功
- **400**: 请求参数错误
- **401**: 未授权（认证失败）
- **409**: 冲突（如邮箱已存在）
- **500**: 服务器内部错误

前端集成示例

JavaScript/TypeScript 示例

```
// 注册
async function register(email: string, password: string, displayName: string) {
  const response = await fetch('http://localhost:3000/api/auth/register', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ email, password, displayName }),
  });

  if (!response.ok) {
    throw new Error('Registration failed');
  }

  const data = await response.json();
  // 保存 tokens
  localStorage.setItem('accessToken', data.data.tokens.accessToken);
  localStorage.setItem('refreshToken', data.data.tokens.refreshToken);

  return data.data.user;
}

// 登录
```

```typescript
async function login(email: string, password: string) {
  const response = await fetch('http://localhost:3000/api/auth/login', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ email, password }),
  });

  if (!response.ok) {
    throw new Error('Login failed');
  }

  const data = await response.json();
  localStorage.setItem('accessToken', data.data.tokens.accessToken);
  localStorage.setItem('refreshToken', data.data.tokens.refreshToken);

  return data.data.user;
}

// 获取当前用户
async function getCurrentUser() {
  const accessToken = localStorage.getItem('accessToken');

  const response = await fetch('http://localhost:3000/api/auth/me', {
    headers: {
      'Authorization': `Bearer ${accessToken}`,
    },
  });

  if (!response.ok) {
    if (response.status === 401) {
      // Token 过期，尝试刷新
      await refreshAccessToken();
      // 重试请求
      return getCurrentUser();
    }
    throw new Error('Failed to get user');
  }

  const data = await response.json();
  return data.data;
}

// 刷新 Access Token
async function refreshAccessToken() {
  const refreshToken = localStorage.getItem('refreshToken');

  const response = await fetch('http://localhost:3000/api/auth/refresh', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ refreshToken }),
  });

  if (!response.ok) {
    // Refresh token 也过期了，需要重新登录
    localStorage.removeItem('accessToken');
    localStorage.removeItem('refreshToken');
    window.location.href = '/login';
    throw new Error('Session expired');
  }

  const data = await response.json();
  localStorage.setItem('accessToken', data.data.accessToken);
  localStorage.setItem('refreshToken', data.data.refreshToken);
}

// 登出
async function logout() {
  const accessToken = localStorage.getItem('accessToken');

  await fetch('http://localhost:3000/api/auth/logout', {
    method: 'POST',
    headers: {
      'Authorization': `Bearer ${accessToken}`,
    },
  });

  localStorage.removeItem('accessToken');
  localStorage.removeItem('refreshToken');
  window.location.href = '/login';
}
```

测试命令

```
# 注册
curl -X POST http://localhost:3000/api/auth/register \
  -H "Content-Type: application/json" \
  -d '{"email": "test@example.com", "password": "password123", "displayName": "测试用户"}'

# 登录
curl -X POST http://localhost:3000/api/auth/login \
  -H "Content-Type: application/json" \
  -d '{"email": "test@example.com", "password": "password123"}'

# 获取当前用户（需要替换 ACCESS_TOKEN）
curl -X GET http://localhost:3000/api/auth/me \
  -H "Authorization: Bearer ACCESS_TOKEN"

# 刷新令牌（需要替换 REFRESH_TOKEN）
```

```
curl -X POST http://localhost:3000/api/auth/refresh \
  -H "Content-Type: application/json" \
  -d '{"refreshToken": "REFRESH_TOKEN"}'

# 登出（需要替换 ACCESS_TOKEN）
curl -X POST http://localhost:3000/api/auth/logout \
  -H "Authorization: Bearer ACCESS_TOKEN"
```