

# Enhanced Signup API Documentation

## Endpoint

POST /api/auth/signup

## Description

Creates a new user account with comprehensive profile information, role assignment, and authentication token. Combines basic signup functionality with detailed registration features including contact information, trading levels, and learning preferences.

## Request Headers

Content-Type: application/json

## Request Body Parameters

### Required Fields

Field	Type	Validation	Description
<input type="text" value="username"/>	string	3-50 chars, trimmed, escaped	Unique username for the account
<input type="text" value="password"/>	string	Min 6 chars	Account password (will be hashed)
<input type="text" value="email"/>	string	Valid email format	User's email address
<input type="text" value="firstName"/>	string	1-100 chars, trimmed, escaped	User's first name
<input type="text" value="lastName"/>	string	1-100 chars, trimmed, escaped	User's last name

### Optional Fields

Field	Type	Validation	Description
role	string	'student' or 'staff'	User role (defaults to 'student')
middleName	string	-	User's middle name
phoneNumber	string	Valid mobile phone	Contact phone number
dateOfBirth	string	ISO8601 date (YYYY-MM-DD)	Date of birth
gender	string	'Male', 'Female', or 'Other'	Gender identity
education	string	1-100 chars, trimmed	Educational background
address	string	1-500 chars, trimmed	Street address
city	string	1-100 chars, trimmed	City name
province	string	1-100 chars, trimmed	Province/State
tradingLevel	string	1-50 chars, trimmed	Trading experience level (student only)
device	string/array	-	Preferred device(s) for learning (student only)
learningStyle	string/array	-	Preferred learning style(s) (student only)

## Sample Request - Student Account

```
json
{
  "username": "johndoe123",
  "password": "securepass123",
  "email": "john.doe@example.com",
  "firstName": "John",
  "middleName": "Michael",
  "lastName": "Doe",
  "role": "student",
  "phoneNumber": "+1234567890",
  "dateOfBirth": "1990-05-15",
  "gender": "Male",
  "education": "Bachelor's Degree",
  "address": "123 Main Street",
  "city": "New York",
  "province": "NY",
  "tradingLevel": "Beginner",
  "device": ["Desktop", "Mobile"],
  "learningStyle": ["Visual", "Interactive"]
}
```

## Sample Request - Staff Account

json

```
{
  "username": "janestaff",
  "password": "staffpass456",
  "email": "jane.staff@company.com",
  "firstName": "Jane",
  "lastName": "Smith",
  "role": "staff",
  "phoneNumber": "+0987654321",
  "dateOfBirth": "1985-08-20",
  "gender": "Female",
  "education": "Master's Degree"
}
```

## Sample Response - Success (201 Created)

### Student Account Response

json

```
{
  "message": "Account created successfully",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "accountId": 12345,
    "username": "johndoe123",
    "role": "student",
    "firstName": "John",
    "middleName": "Michael",
    "lastName": "Doe",
    "email": "john.doe@example.com",
    "phoneNumber": "+1234567890",
    "dateOfBirth": "1990-05-15",
    "gender": "Male",
    "education": "Bachelor's Degree",
    "address": "123 Main Street, New York, NY",
    "permissions": "student_permissions_string",
    "profile": {
      "student_id": "S1703123456789_67890",
      "graduation_status": "enrolled",
      "academic_standing": "good",
      "trading_level": "Beginner",
      "learning_preferences": {
        "device": ["Desktop", "Mobile"],
        "learning_style": ["Visual", "Interactive"]
      }
    }
  }
}
```

## Staff Account Response

json

```
{
  "message": "Account created successfully",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "accountId": 12346,
    "username": "janestaff",
    "role": "staff",
    "firstName": "Jane",
    "middleName": null,
    "lastName": "Smith",
    "email": "jane.staff@company.com",
    "phoneNumber": "+0987654321",
    "dateOfBirth": "1985-08-20",
    "gender": "Female",
    "education": "Master's Degree",
    "address": null,
    "permissions": "staff_permissions_string",
    "profile": {
      "employee_id": "EMP1703123456789",
      "employment_status": "active"
    }
  }
}
```

## Error Responses

### Validation Error (400 Bad Request)

json

```
{
  "errors": [
    {
      "field": "email",
      "message": "Valid email is required"
    },
    {
      "field": "password",
      "message": "Password must be at least 6 characters"
    }
  ]
}
```

### **Username Already Exists (409 Conflict)**

json

```
{
  "error": "Username already exists"
}
```

### **Email Already Exists (409 Conflict)**

json

```
{
  "error": "Email already exists"
}
```

### **Invalid Role (400 Bad Request)**

json

```
{
  "error": "Invalid role specified"
}
```

### **Server Error (500 Internal Server Error)**

```
json
```

```
{  
  "error": "Account creation failed"  
}
```

## Features and Functionality

### Authentication

- Passwords are hashed using bcrypt with 12 salt rounds
- JWT tokens are generated with 24-hour expiration
- Tokens include accountId, username, and role claims

### Student-Specific Features

- Auto-generated unique student ID: `S{timestamp}_{person_id}`
- Contact information storage (phone, address, email)
- Trading level assignment with validation against `trading_levels` table
- Learning preferences storage (device types and learning styles)
- Default graduation status: 'enrolled'
- Default academic standing: 'good'

### Staff-Specific Features

- Auto-generated employee ID: `EMP{timestamp}`
- Employment status set to 'active'
- Hire date set to current date
- Contact information storage

### Database Operations

- Uses database transactions for data consistency
- Checks for duplicate usernames and emails before insertion
- Creates records across multiple tables:
  - `accounts` - Authentication data
  - `persons` - Personal information
  - `account_roles` - Role assignments

- `students` or `staff` - Role-specific data
- `contact_info` - Contact details
- `student_trading_levels` - Trading experience (students)
- `learning_preferences` - Learning preferences (students)

## cURL Examples

### Create Student Account

bash

```
curl -X POST http://localhost:3000/api/auth/signup \
-H "Content-Type: application/json" \
-d '{
  "username": "johndoe123",
  "password": "securepass123",
  "email": "john.doe@example.com",
  "firstName": "John",
  "lastName": "Doe",
  "role": "student",
  "phoneNumber": "+1234567890",
  "dateOfBirth": "1990-05-15",
  "gender": "Male",
  "education": "Bachelor's Degree",
  "address": "123 Main Street",
  "city": "New York",
  "province": "NY",
  "tradingLevel": "Beginner",
  "device": ["Desktop", "Mobile"],
  "learningStyle": ["Visual", "Interactive"]
}'
```

### Create Staff Account



bash

```
curl -X POST http://localhost:3000/api/auth/signup \  
-H "Content-Type: application/json" \  
-d '{  
  "username": "janestaff",  
  "password": "staffpass456",  
  "email": "jane.staff@company.com",  
  "firstName": "Jane",  
  "lastName": "Smith",  
  "role": "staff",  
  "phoneNumber": "+0987654321",  
  "dateOfBirth": "1985-08-20",  
  "gender": "Female",  
  "education": "Master\'\'s Degree"  
}'
```

## JavaScript/Fetch Example

javascript

```
const signupUser = async (userData) => {
  try {
    const response = await fetch('/api/auth/signup', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify(userData)
    });

    const result = await response.json();

    if (response.ok) {
      // Store token for authenticated requests
      localStorage.setItem('authToken', result.token);
      console.log('Signup successful:', result);
      return result;
    } else {
      console.error('Signup failed:', result);
      throw new Error(result.error || 'Signup failed');
    }
  } catch (error) {
    console.error('Network error:', error);
    throw error;
  }
};
```

*// Usage - Student*

```
const newStudent = {
  username: "studentuser",
  password: "password123",
  email: "student@example.com",
  firstName: "Alice",
  lastName: "Johnson",
  role: "student",
  phoneNumber: "+1555123456",
  dateOfBirth: "1995-03-10",
  gender: "Female",
  education: "High School",
  address: "789 Student St",
  city: "Boston",
  province: "MA",
  tradingLevel: "Beginner",
```

```
    device: "Mobile",
    learningStyle: "Visual"
  };

// Usage - Staff
const newStaff = {
  username: "staffuser",
  password: "staff123",
  email: "staff@company.com",
  firstName: "Bob",
  lastName: "Wilson",
  role: "staff",
  phoneNumber: "+1555987654",
  dateOfBirth: "1980-12-05",
  gender: "Male",
  education: "Master's Degree"
};

signupUser(newStudent);
```

## Important Notes

- All string fields are automatically trimmed and escaped for security
- Email addresses are normalized to lowercase
- Role defaults to 'student' if not specified
- Student IDs and Employee IDs are auto-generated
- Trading levels must exist in the `trading_levels` table
- Device and learning style fields can accept single values or arrays
- Phone numbers must be in valid mobile phone format
- Dates must be in ISO8601 format (YYYY-MM-DD)
- All database operations use transactions for data integrity
- JWT tokens expire after 24 hours
- Comprehensive contact information is stored for better user management