

SupportDesk API Documentation

Complete API reference for the SupportDesk customer support ticket system.

Base URL

```
http://localhost:3000/api
```

Authentication

All API endpoints (except `/api/signup` and `/api/auth/*`) require authentication via NextAuth session. Include credentials in your requests.

Authentication Endpoints

POST /api/signup

Create a new user account.

Request Body:

```
{
  "email": "user@example.com",
  "password": "securepassword",
  "name": "John Doe",
  "role": "customer" // optional, defaults to 'customer'
}
```

Response (201):

```
{
  "user": {
    "id": "clxxxx...",
    "email": "user@example.com",
    "name": "John Doe",
    "role": "customer",
    "createdAt": "2024-01-01T00:00:00.000Z"
  },
  "message": "User created successfully"
}
```

Errors:

- 400 : Missing required fields or user already exists
 - 500 : Internal server error
-

POST /api/auth/signin

Handled by NextAuth. Use NextAuth client methods.

Ticket Endpoints

GET /api/tickets

Retrieve tickets based on user role and filters.

Query Parameters:

- `status` (optional): Filter by status (open, in_progress, resolved, closed)
- `priority` (optional): Filter by priority (low, medium, high)
- `category` (optional): Filter by category (technical_issue, billing, feature_request, general_inquiry)
- `assignedTo` (optional): Filter by assigned agent ID
- `search` (optional): Search in title, description, or ID

Response (200):

```
{
  "tickets": [
    {
      "id": "clxxxx...",
      "title": "Login issue",
      "description": "Cannot login to my account",
      "status": "open",
      "priority": "high",
      "category": "technical_issue",
      "createdById": "clxxxx...",
      "assignedToId": "clxxxx...",
      "createdAt": "2024-01-01T00:00:00.000Z",
      "updatedAt": "2024-01-01T00:00:00.000Z",
      "createdBy": {
        "id": "clxxxx...",
        "name": "John Doe",
        "email": "john@example.com",
        "role": "customer"
      },
      "assignedTo": {
        "id": "clxxxx...",
        "name": "Agent Smith",
        "email": "agent@supportdesk.com",
        "role": "agent"
      },
      "comments": []
    }
  ]
}
```

Access Control:

- **Customer:** Can only see their own tickets
 - **Agent:** Can see tickets assigned to them or created by them
 - **Admin:** Can see all tickets
-

POST /api/tickets

Create a new support ticket.

Request Body:

```
{
  "title": "Need help with billing",
  "description": "I was charged twice for my subscription",
  "priority": "medium", // optional, defaults to 'medium'
  "category": "billing"
}
```

Response (201):

```
{
  "ticket": {
    "id": "clxxxx...",
    "title": "Need help with billing",
    "description": "I was charged twice for my subscription",
    "status": "open",
    "priority": "medium",
    "category": "billing",
    "createdById": "clxxxx...",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "createdBy": {
      "id": "clxxxx...",
      "name": "John Doe",
      "email": "john@example.com",
      "role": "customer"
    }
  }
}
```

GET /api/tickets/[id]

Get detailed information about a specific ticket.

Response (200):

```
{
  "ticket": {
    "id": "clxxxx...",
    "title": "Login issue",
    "description": "Cannot login to my account",
    "status": "in_progress",
    "priority": "high",
    "category": "technical_issue",
    "createdById": "clxxxx...",
    "assignedToId": "clxxxx...",
    "createdAt": "2024-01-01T00:00:00.000Z",
    "updatedAt": "2024-01-01T12:00:00.000Z",
    "createdBy": { /* user object */ },
    "assignedTo": { /* user object */ },
    "comments": [
      {
        "id": "clxxxx...",
        "content": "I'm looking into this",
        "userId": "clxxxx...",
        "createdAt": "2024-01-01T10:00:00.000Z",
        "user": { /* user object */ }
      }
    ],
    "activities": [
      {
        "id": "clxxxx...",
        "action": "updated",
        "description": "Status changed from open to in_progress",
        "userId": "clxxxx...",
        "createdAt": "2024-01-01T09:00:00.000Z",
        "user": { /* user object */ }
      }
    ]
  }
}
```

PATCH /api/tickets/[id]

Update ticket properties (agents and admins only).

Request Body:

```
{
  "status": "in_progress",
  "priority": "high",
  "category": "technical_issue",
  "assignedToId": "clxxxx..." // admin only
}
```

Response (200):

```
{
  "ticket": { /* updated ticket object */ }
}
```

Access Control:

- **Agent:** Can update tickets assigned to them
 - **Admin:** Can update any ticket and assign agents
-

DELETE /api/tickets/[id]

Delete a ticket (admin only).

Response (200):

```
{
  "message": "Ticket deleted successfully"
}
```

Comment Endpoints**POST /api/comments**

Add a comment to a ticket.

Request Body:

```
{
  "ticketId": "clxxxx...",
  "content": "I've reset your password. Please try logging in again."
}
```

Response (201):

```
{
  "comment": {
    "id": "clxxxx...",
    "content": "I've reset your password...",
    "ticketId": "clxxxx...",
    "userId": "clxxxx...",
    "createdAt": "2024-01-01T12:00:00.000Z",
    "user": {
      "id": "clxxxx...",
      "name": "Agent Smith",
      "email": "agent@supportdesk.com",
      "role": "agent"
    }
  }
}
```

Analytics Endpoints

GET /api/analytics

Get comprehensive analytics data (admin only).

Response (200):

```

{
  "analytics": {
    "totalTickets": 150,
    "openTickets": 45,
    "closedTickets": 105,
    "averageResponseTime": 2, // hours
    "ticketsByCategory": [
      {
        "category": "technical_issue",
        "count": 65
      },
      {
        "category": "billing",
        "count": 40
      },
      {
        "category": "feature_request",
        "count": 30
      },
      {
        "category": "general_inquiry",
        "count": 15
      }
    ],
    "ticketsByPriority": [
      {
        "priority": "low",
        "count": 50
      },
      {
        "priority": "medium",
        "count": 70
      },
      {
        "priority": "high",
        "count": 30
      }
    ],
    "recentActivity": [
      {
        "id": "clxxxx...",
        "ticketId": "clxxxx...",
        "userId": "clxxxx...",
        "action": "updated",
        "description": "Status changed from open to in_progress",
        "createdAt": "2024-01-01T12:00:00.000Z",
        "user": { /* user object */ },
        "ticket": {
          "id": "clxxxx...",
          "title": "Login issue"
        }
      }
    ]
  }
}

```

User Management Endpoints

GET /api/users

Get list of users (agents and admins only).

Query Parameters:

- `role` (optional): Filter by role (admin, agent, customer)

Response (200):

```
{
  "users": [
    {
      "id": "clxxxx...",
      "name": "John Doe",
      "email": "john@example.com",
      "role": "customer",
      "image": null,
      "createdAt": "2024-01-01T00:00:00.000Z",
      "_count": {
        "ticketsCreated": 5,
        "ticketsAssigned": 0
      }
    }
  ]
}
```

PATCH /api/users

Update user role (admin only).

Request Body:

```
{
  "userId": "clxxxx...",
  "role": "agent"
}
```

Response (200):

```
{
  "user": {
    "id": "clxxxx...",
    "name": "John Doe",
    "email": "john@example.com",
    "role": "agent",
    "createdAt": "2024-01-01T00:00:00.000Z"
  }
}
```

Error Responses

All endpoints may return the following error responses:

401 Unauthorized

```
{  
  "error": "Unauthorized"  
}
```

403 Forbidden

```
{  
  "error": "Forbidden"  
}
```

404 Not Found

```
{  
  "error": "Ticket not found"  
}
```

500 Internal Server Error

```
{  
  "error": "Internal server error"  
}
```

Rate Limiting

Currently, there are no rate limits implemented. Consider implementing rate limiting in production.

Webhooks

Not currently supported. Future enhancement.

Pagination

Not currently implemented. All endpoints return full result sets. Consider implementing pagination for large datasets in production.