**TradeBOT365 API Documentation Template**

**Overview**

This document outlines the API endpoints for the user-facing features of TradeBOT365. All endpoints require user authentication and apply strict data filtering to ensure users can only access their own data.

**Base URL**

https://api.tradebot365.com

**Authentication**

All endpoints require authentication via an access token.

**Request Headers**:

Authorization: Bearer {access\_token}

**User API Endpoints**

**Authentication & User Management**

**1. User Login**

**Endpoint:** POST /auth/login

**Body Parameters:**

{

"email": "string",

"password": "string"

}

**Response:**

{

"success": true,

"data": {

"user": {

"id": "string",

"email": "string",

"name": "string",

"role": "user",

"status": "active",

"plan": "basic"

// Other user properties

},

"accessToken": "string",

"refreshToken": "string"

},

"message": "Login successful"

}

**2. User Registration**

**Endpoint:** POST /auth/register

**Body Parameters:**

{

"email": "string",

"password": "string",

"name": "string"

}

**Response:**

{

"success": true,

"data": {

"user": {

"id": "string",

"email": "string",

"name": "string"

// Other user properties

},

"accessToken": "string"

},

"message": "Registration successful"

}

**3. Get User Profile**

**Endpoint:** GET /auth/profile

**Response:**

{

"success": true,

"data": {

"id": "string",

"email": "string",

"name": "string",

"firstName": "string",

"lastName": "string",

"username": "string",

"phone": "string",

"country": "string",

"city": "string",

"address": "string",

"postalCode": "string",

"avatar": "string",

"role": "user",

"status": "active",

"plan": "basic",

"createdAt": "string",

"updatedAt": "string",

"lastLogin": "string",

"emailVerified": true,

"twoFactorEnabled": false,

"bots": 5,

"joinDate": "string",

"botTypes": ["string"],

"activity": "string",

"preferences": {

"theme": "dark",

"language": "vi",

"notificationsEnabled": true,

"emailNotificationsEnabled": true,

"smsNotificationsEnabled": false

}

}

}

**4. Update User Profile**

**Endpoint:** PUT /users/profile

**Body Parameters:**

{

"name": "string",

"firstName": "string",

"lastName": "string",

"phone": "string",

"country": "string",

"city": "string",

"address": "string",

"postalCode": "string",

"avatar": "string"

}

**Response:**

{

"success": true,

"data": {

// Updated user data

},

"message": "Profile updated successfully",

"updated": true

}

**5. Change Password**

**Endpoint:** POST /users/change-password

**Body Parameters:**

{

"currentPassword": "string",

"newPassword": "string",

"confirmPassword": "string"

}

**Response:**

{

"success": true,

"message": "Password changed successfully"

}

**6. Request Password Reset**

**Endpoint:** POST /auth/reset-password

**Body Parameters:**

{

"email": "string"

}

**Response:**

{

"success": true,

"message": "Password reset instructions sent to your email"

}

**7. Verify Email**

**Endpoint:** POST /auth/verify-email

**Body Parameters:**

{

"token": "string"

}

**Response:**

{

"success": true,

"message": "Email verified successfully"

}

**Dashboard**

**8. Get Dashboard Overview**

**Endpoint:** GET /dashboard

**Response:**

{

"success": true,

"data": {

"totalBots": 5,

"activeBots": 3,

"todayActiveBots": 2,

"totalAccounts": 12,

"connectedAccounts": 8,

"todayConnectedAccounts": 5,

"todaySignals": 8,

"pendingSignals": 3,

"processedSignals": 5,

"failedSignals": 0,

"monthlyOrders": 398,

"systemLoad": 42,

"uptime": "99.8%",

"lastUpdated": "2023-06-15T10:30:00Z"

}

}

**User Bots Management**

**9. List User Bots**

**Endpoint:** GET /bots/user

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

status: "Active" | "Inactive" | "All"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "user\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"owner": "string",

"ownerId": "string",

"accounts": 3,

"accountCount": "3/10",

"strategy": "string",

"isActive": true,

"botId": "BOT7459",

"settings": {

"maxLoss": "string",

"maxRisk": "string",

"maxPositions": 5,

"activeTradingHours": ["string"],

"excludedSymbols": ["string"],

"includedSymbols": ["string"],

"stopLoss": "string",

"takeProfit": "string",

"lotSize": "string",

"leverage": "string"

}

}

],

"total": 5,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**10. Get Bot Detail**

**Endpoint:** GET /bots/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "user\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"owner": "string",

"ownerId": "string",

"accounts": 3,

"accountsList": [

{

"id": "string",

"name": "string",

"status": "Connected",

"apiName": "string",

"tradingAccountBalance": "string"

// Other account properties

}

],

"strategy": "string",

"isActive": true,

"botId": "BOT7459",

"settings": {

"maxLoss": "string",

"maxRisk": "string",

"maxPositions": 5,

"activeTradingHours": ["string"],

"excludedSymbols": ["string"],

"includedSymbols": ["string"],

"stopLoss": "string",

"takeProfit": "string",

"lotSize": "string",

"leverage": "string"

},

"webhookUrl": "string",

"signalToken": "string"

}

}

**11. Create User Bot**

**Endpoint:** POST /bots

**Body Parameters:**

{

"name": "string",

"description": "string",

"exchange": "string",

"botForm": "trading\_view",

"colorScheme": "red",

"risk": "medium",

"settings": {

"maxLoss": "string",

"maxRisk": "string",

"maxPositions": 5,

"activeTradingHours": ["string"],

"excludedSymbols": ["string"],

"includedSymbols": ["string"],

"stopLoss": "string",

"takeProfit": "string",

"lotSize": "string",

"leverage": "string"

}

}

**Response:**

{

"success": true,

"data": {

"id": "string",

"botId": "BOT1234",

"name": "string",

"webhookUrl": "string",

"signalToken": "string"

// Other bot properties

},

"message": "Bot created successfully",

"id": "BOT1234"

}

**12. Update User Bot**

**Endpoint:** PUT /bots/{botId}

**Path Parameters:**

botId: string

**Body Parameters:**

{

"name": "string",

"description": "string",

"exchange": "string",

"colorScheme": "blue",

"risk": "low",

"settings": {

"maxLoss": "string",

"maxRisk": "string",

"maxPositions": 3,

"activeTradingHours": ["string"],

"excludedSymbols": ["string"],

"includedSymbols": ["string"],

"stopLoss": "string",

"takeProfit": "string",

"lotSize": "string",

"leverage": "string"

}

}

**Response:**

{

"success": true,

"data": {

// Updated bot data

},

"message": "Bot updated successfully",

"updated": true

}

**13. Delete User Bot**

**Endpoint:** DELETE /bots/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"message": "Bot deleted successfully",

"deleted": true

}

**Premium Bots**

**14. List Premium Bots**

**Endpoint:** GET /bots/premium

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "premium\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"minCapital": "string",

"subscribers": 120,

"colorScheme": "blue",

"isIntegrated": false,

"features": ["string"],

"tradingStyle": "string",

"timeframe": "string",

"markets": ["string"],

"botId": "PB001"

}

],

"total": 15,

"page": 1,

"pageSize": 10,

"totalPages": 2,

"hasMore": true

}

}

**15. Get Premium Bot Detail**

**Endpoint:** GET /bots/premium/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "premium\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"minCapital": "string",

"subscribers": 120,

"colorScheme": "blue",

"isIntegrated": false,

"features": ["string"],

"tradingStyle": "string",

"timeframe": "string",

"markets": ["string"],

"botId": "PB001",

"performance": {

"totalPnL": "string",

"totalTrades": 150,

"winRate": "68%",

"avgProfit": "string",

"avgLoss": "string",

"maxDrawdown": "string",

"profitFactor": "string",

"lastMonthPerformance": "string",

"allTimePerformance": "string",

"monthlyPerformance": [

{

"month": "string",

"performance": "string",

"trades": 28,

"winRate": "string"

}

]

}

}

}

**16. Integrate Premium Bot**

**Endpoint:** POST /bots/premium/{botId}/integrate

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"originalId": "string",

"integratedDate": "string",

// Other integrated bot properties

},

"message": "Bot integrated successfully"

}

**Integrated Bots**

**17. List Integrated Premium Bots**

**Endpoint:** GET /bots/integrated/premium

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "premium\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"originalId": "string",

"integratedDate": "string",

"accounts": [

// Account list

]

}

],

"total": 3,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**18. Get Integrated Premium Bot Detail**

**Endpoint:** GET /bots/integrated/premium/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "premium\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"originalId": "string",

"integratedDate": "string",

"accounts": [

// Account list

],

"performance": {

// Performance data

}

}

}

**Prop Trading Bots**

**19. List Prop Trading Bots**

**Endpoint:** GET /bots/prop

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "prop\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"minCapital": "string",

"users": 85,

"profit": "string",

"maxDrawdown": "string",

"propFirm": "string",

"challengeDuration": "string",

"accountSizes": ["string"]

}

],

"total": 8,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**20. Get Prop Trading Bot Detail**

**Endpoint:** GET /bots/prop/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "prop\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"minCapital": "string",

"users": 85,

"profit": "string",

"maxDrawdown": "string",

"propFirm": "string",

"challengeDuration": "string",

"accountSizes": ["string"],

"challengeRules": ["string"],

"performance": {

// Performance data

},

"challengeData": {

"phase": "string",

"progress": 68,

"accountBalance": "string",

"profitTarget": "string",

"maxDrawdown": "string",

"daysRemaining": "string",

"description": "string"

}

}

}

**21. Integrate Prop Trading Bot**

**Endpoint:** POST /bots/prop/{botId}/integrate

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"originalId": "string",

"integratedDate": "string",

// Other integrated bot properties

},

"message": "Prop trading bot integrated successfully"

}

**Integrated Prop Trading Bots**

**22. List Integrated Prop Trading Bots**

**Endpoint:** GET /bots/integrated/prop

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "prop\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"originalId": "string",

"integratedDate": "string",

"accounts": [

// Account list

]

}

],

"total": 2,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**23. Get Integrated Prop Trading Bot Detail**

**Endpoint:** GET /bots/integrated/prop/{botId}

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"description": "string",

"status": "Active",

"type": "prop\_bot",

"createdDate": "string",

"lastUpdated": "string",

"risk": "medium",

"imageUrl": "string",

"exchange": "string",

"performanceLastMonth": "string",

"performanceAllTime": "string",

"originalId": "string",

"integratedDate": "string",

"accounts": [

// Account list

],

"performance": {

// Performance data

},

"challengeData": {

// Challenge data

}

}

}

**Account Management**

**24. List Accounts**

**Endpoint:** GET /accounts

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

status: "Connected" | "Disconnected" | "Pending" | "All"

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"status": "Connected",

"createdDate": "string",

"lastUpdated": "string",

"userId": "string",

"apiName": "string",

"apiId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"volumeMultiplier": "string",

"clientId": "string",

"secretId": "string",

"accessToken": "string",

"ctidTraderAccountId": "string",

"expireDate": "string",

"userAccount": "string",

"userEmail": "string"

}

],

"total": 5,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**25. Get Account Detail**

**Endpoint:** GET /accounts/{accountId}

**Path Parameters:**

accountId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"status": "Connected",

"createdDate": "string",

"lastUpdated": "string",

"userId": "string",

"apiName": "string",

"apiId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"volumeMultiplier": "string",

"clientId": "string",

"secretId": "string",

"accessToken": "string",

"ctidTraderAccountId": "string",

"expireDate": "string",

"userAccount": "string",

"userEmail": "string",

"connectedBots": [

{

"id": "string",

"name": "string",

"type": "user\_bot",

"status": "Active"

}

]

}

}

**26. Create Account**

**Endpoint:** POST /accounts

**Body Parameters:**

{

"name": "string",

"apiName": "string",

"apiId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"volumeMultiplier": "string",

"clientId": "string",

"secretId": "string",

"accessToken": "string",

"ctidTraderAccountId": "string",

"userEmail": "string"

}

**Response:**

{

"success": true,

"data": {

"id": "string",

"name": "string",

"status": "Pending",

// Other account properties

},

"message": "Account created successfully",

"id": "string"

}

**27. Update Account**

**Endpoint:** PUT /accounts/{accountId}

**Path Parameters:**

accountId: string

**Body Parameters:**

{

"name": "string",

"apiName": "string",

"apiId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"volumeMultiplier": "string",

"clientId": "string",

"secretId": "string",

"accessToken": "string",

"ctidTraderAccountId": "string",

"userEmail": "string"

}

**Response:**

{

"success": true,

"data": {

// Updated account data

},

"message": "Account updated successfully",

"updated": true

}

**28. Delete Account**

**Endpoint:** DELETE /accounts/{accountId}

**Path Parameters:**

accountId: string

**Response:**

{

"success": true,

"message": "Account deleted successfully",

"deleted": true

}

**29. Connect Account**

**Endpoint:** POST /accounts/{accountId}/connect

**Path Parameters:**

accountId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"status": "Connected",

"lastUpdated": "string"

// Other account properties

},

"message": "Account connected successfully"

}

**30. Disconnect Account**

**Endpoint:** POST /accounts/{accountId}/disconnect

**Path Parameters:**

accountId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"status": "Disconnected",

"lastUpdated": "string"

// Other account properties

},

"message": "Account disconnected successfully"

}

**Bot Accounts**

**31. Get Bot Accounts**

**Endpoint:** GET /bots/{botId}/accounts

**Path Parameters:**

botId: string

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"name": "string",

"status": "Connected",

// Other account properties

}

],

"total": 3,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**32. Add Account to Bot**

**Endpoint:** POST /bots/{botId}/accounts

**Path Parameters:**

botId: string

**Body Parameters:**

{

"accountId": "string"

}

**Response:**

{

"success": true,

"message": "Account added to bot successfully"

}

**33. Remove Account from Bot**

**Endpoint:** DELETE /bots/{botId}/accounts/{accountId}

**Path Parameters:**

botId: string

accountId: string

**Response:**

{

"success": true,

"message": "Account removed from bot successfully",

"deleted": true

}

**Signals**

**34. List Signals**

**Endpoint:** GET /signals

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

status: "Pending" | "Processed" | "Failed" | "All"

botId: string

dateFrom: string

dateTo: string

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"action": "ENTER\_LONG",

"instrument": "string",

"timestamp": "string",

"signalToken": "string",

"maxLag": "string",

"investmentType": "string",

"amount": "string",

"status": "Processed",

"errorMessage": "string",

"userId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"botId": "string",

"botType": "string",

"botName": "string",

"exchange": "string"

}

],

"total": 120,

"page": 1,

"pageSize": 10,

"totalPages": 12,

"hasMore": true

}

}

**35. Get Signal Detail**

**Endpoint:** GET /signals/{signalId}

**Path Parameters:**

signalId: string

**Response:**

{

"success": true,

"data": {

"id": "string",

"action": "ENTER\_LONG",

"instrument": "string",

"timestamp": "string",

"signalToken": "string",

"maxLag": "string",

"investmentType": "string",

"amount": "string",

"status": "Processed",

"errorMessage": "string",

"userId": "string",

"tradingAccount": "string",

"tradingAccountType": "string",

"tradingAccountBalance": "string",

"botId": "string",

"botType": "string",

"botName": "string",

"exchange": "string",

"processedAccounts": [

{

"accountId": "string",

"userId": "string",

"name": "string",

"timestamp": "string",

"status": "success"

}

],

"failedAccounts": [

{

"accountId": "string",

"userId": "string",

"name": "string",

"timestamp": "string",

"reason": "string",

"errorCode": "string",

"status": "failed"

}

]

}

}

**36. List Error Signals**

**Endpoint:** GET /signals/errors

**Query Parameters:**

page: number

pageSize: number

search: string

sortBy: string

sortDirection: "asc" | "desc"

botId: string

dateFrom: string

dateTo: string

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"action": "ENTER\_LONG",

"instrument": "string",

"timestamp": "string",

"status": "Failed",

"errorMessage": "string",

"errorCode": "string",

"errorSeverity": "medium",

"botId": "string",

"botName": "string",

"botType": "string",

"accountId": "string",

"accountName": "string"

}

],

"total": 8,

"page": 1,

"pageSize": 10,

"totalPages": 1,

"hasMore": false

}

}

**Bot Logs**

**37. Get TradingView Logs for Bot**

**Endpoint:** GET /bots/{botId}/logs/tradingview

**Path Parameters:**

botId: string

**Query Parameters:**

page: number

pageSize: number

dateFrom: string

dateTo: string

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"action": "ENTER\_LONG",

"instrument": "string",

"timestamp": "string",

"signalToken": "string",

"source": "string",

"processingTime": "string",

"accountId": "string",

"accountName": "string",

"status": "Processed"

}

],

"total": 45,

"page": 1,

"pageSize": 10,

"totalPages": 5,

"hasMore": true

}

}

**38. Get Coinstrat Logs for Bot**

**Endpoint:** GET /bots/{botId}/logs/coinstrat

**Path Parameters:**

botId: string

**Query Parameters:**

page: number

pageSize: number

dateFrom: string

dateTo: string

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"action": "ENTER\_LONG",

"instrument": "string",

"timestamp": "string",

"signalToken": "string",

"originalSignalId": "string",

"processedAccounts": [

{

"accountId": "string",

"userId": "string",

"name": "string",

"timestamp": "string",

"status": "success"

}

],

"failedAccounts": [

{

"accountId": "string",

"userId": "string",

"name": "string",

"timestamp": "string",

"reason": "string",

"errorCode": "string",

"status": "failed"

}

],

"botId": "string",

"botName": "string",

"status": "Processed"

}

],

"total": 35,

"page": 1,

"pageSize": 10,

"totalPages": 4,

"hasMore": true

}

}

**User Preferences and Settings**

**39. Get User Preferences**

**Endpoint:** GET /users/preferences

**Response:**

{

"success": true,

"data": {

"theme": "dark",

"language": "vi",

"notificationsEnabled": true,

"emailNotificationsEnabled": true,

"smsNotificationsEnabled": false

}

}

**40. Update User Preferences**

**Endpoint:** PUT /users/preferences

**Body Parameters:**

{

"theme": "light",

"language": "en",

"notificationsEnabled": true,

"emailNotificationsEnabled": true,

"smsNotificationsEnabled": true

}

**Response:**

{

"success": true,

"data": {

"theme": "light",

"language": "en",

"notificationsEnabled": true,

"emailNotificationsEnabled": true,

"smsNotificationsEnabled": true

},

"message": "Preferences updated successfully",

"updated": true

}

**41. Get User Notifications**

**Endpoint:** GET /users/notifications

**Query Parameters:**

page: number

pageSize: number

read: boolean

**Response:**

{

"success": true,

"data": {

"items": [

{

"id": "string",

"userId": "string",

"title": "string",

"message": "string",

"timestamp": "string",

"read": false,

"type": "info",

"link": "string",

"linkText": "string",

"source": "system"

}

],

"total": 15,

"page": 1,

"pageSize": 10,

"totalPages": 2,

"hasMore": true

}

}

**42. Mark Notification as Read**

**Endpoint:** PUT /users/notifications/{notificationId}

**Path Parameters:**

notificationId: string

**Body Parameters:**

{

"read": true

}

**Response:**

{

"success": true,

"message": "Notification marked as read",

"updated": true

}

**System Stats and Information**

**43. Get System Status**

**Endpoint:** GET /system/status

**Response:**

{

"success": true,

"data": {

"version": "string",

"buildNumber": "string",

"releaseDate": "string",

"environment": "production",

"features": ["string"],

"changelog": ["string"],

"systemLoad": 35,

"uptime": "string",

"lastUpdated": "string"

}

}