StockGro API Documentation

Overview

This document provides comprehensive documentation for the StockGro public API endpoints, including authentication, request/response formats, and detailed parameter descriptions.

Base URL: `https://prod.stockgro.com/public/api/v1`

Content-Type: `application/json`

Authentication

All public API endpoints use **HMAC-SHA256 signature-based authentication** with the following headers:

Required Headers

Header	Description	Example
X-Client-Id	Client identifier registered in the system	33XuyZJfIhhtXKkaVvTx7T5xb7y
X-Signature	HMAC-SHA256 signature of the request	a1b2c3d4e5f6...
X-Nonce	Unique nonce value for each request	1234567890

Step-by-Step Authentication Guide

1. Obtain Client Credentials

Your client ID and secret must be registered in the `clientSecrets`

2. Generate Authentication Headers

For each request, you need to:

Step 2.1: Generate a unique nonce

```
const nonce = Date.now().toString() + Math.random().toString(36).substr(2, 9);
```

Step 2.2: Create the message to sign

```
const clientId = "33XuyZJfIhhtXKkaVvTx7T5xb7y";
const message = `${clientId}:${nonce}`;
```

Step 2.3: Generate HMAC-SHA256 signature

```
const crypto = require('crypto');
const clientSecret = "33XuyVulURPuh8vivAHVdATzboa";
const signature = crypto
    .createHmac('sha256', clientSecret)
    .update(message)
    .digest('hex');
```

Step 2.4: Include headers in your request

```
const headers = {
    'X-Client-Id': clientId,
    'X-Signature': signature,
    'X-Nonce': nonce,
    'Content-Type': 'application/json'
};
```

3. Complete Request Example

```bash
# Generate signature (using bash)
CLIENT_ID="33XuyZJfIhhtXKkaVvTx7T5xb7y"
CLIENT_SECRET="33XuyVulURPuh8vivAHVdATzboa"
NONCE="1234567890"

MESSAGE="${CLIENT_ID}:${NONCE}"
SIGNATURE=$(echo -n "$MESSAGE" | openssl dgst -sha256 -hmac "$CLIENT_SECRET" -hex |
cut -d' ' -f2)

curl -X GET "https://api.example.com/public/api/v1/stocks" \
  -H "X-Client-Id: $CLIENT_ID" \
  -H "X-Signature: $SIGNATURE" \
  -H "X-Nonce: $NONCE"
```

Authentication Errors

401 Unauthorized - Unknown Client:

```
{
    "error": "unknown client_id"
}
```

401 Unauthorized - Invalid Signature:

```
{
    "error": "invalid signature"
}
```

API Endpoints

1. GET `/stocks/search`

Search for stocks by symbol or company name with optional filtering and sorting.

Query Parameters

Parameter	Type	Required	Description	Example
tenant_id	integer	Yes	Tenant identifier	1
search_param	string	No	Search term (min 3 chars)	DU

page	integer	No	Page number	1
limit	integer	No	Items per page (max 3000)	50
country_code	string	No	Country code filter	IN
sort	string	No	Sort field	percentage
order	string	No	Sort order (asc or desc)	desc
symbols	string	No	Comma-separated symbols	DU,TALABAT
sectors	string	No	Comma-separated sectors	IT,Oil
industry	string	No	Comma-separated industries	Software,Refinery
market_cap	string	No	Market cap filter	large,mid
stock_score	string	No	Stock score filter	80-100,60-79
analyst_recommendation	string	No	Analyst rating filter	buy,hold
screener	string	No	Screener name	momentum
pattern	string	No	Pattern within screener	bullish

Sort Options

Sort Key	Description	Default Order
ltp	Last Traded Price	-
score	Stock Score	-
symbol	Stock Symbol	-
price_change	Price Change	-
percentage	Percentage Change	desc
gainers	Top gainers	desc
losers	Top losers	asc
ft_week_high	52-week high breakers	desc
ft_week_low	52-week low breakers	asc
volume_shockers	Volume shockers	desc

Request Examples

Basic Search:

```
curl -X GET "https://api.example.com/public/api/v1/stocks/search?
tenant_id=3&search_param=DU" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Advanced Search with Filters:

```
curl -X GET "https://api.example.com/public/api/v1/stocks/search?
tenant_id=3&search_param=bank&page=1&limit=20&sectors=Banking&sort=score&order=desc" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Search with Screener:

```
curl -X GET "https://api.example.com/public/api/v1/stocks/search?
tenant_id=3&screener=momentum&pattern=bullish&page=1&limit=50" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Response Format

Success Response (200):

```
{
    "success": true,
    "data": {
        "stocks": [
            {
                "symbol": "DU",
                "name": "DU",
                "sector": "Oil & Gas",
                "industry": "Refinery",
                "ltp": 2450.50,
                "previous_close": 2420.00,
                "percentage_change": 2.50,
                "volume": 1500000,
                "market_cap": 1650000000000,
                "score": 85,
                "exchange": "NSE",
                "currency": "INR"
            }
        ],
        "count": 150,
        "updated_at": "2024-01-15T10:30:00Z"
    }
}
```

Empty Results (200):

```json
{
    "success": true,
    "data": {
        "stocks": [],
        "count": 0
    }
}
```

---

2. GET `/stocks`

Get a paginated list of stocks with comprehensive filtering options.

Query Parameters

Parameter	Type	Required	Description	Example
tenant_id	integer	Yes	Tenant identifier	1
page	integer	No	Page number	1
limit	integer	No	Items per page (max 3000)	50
country_code	string	No	Country code filter	IN
sort	string	No	Sort field	score
order	string	No	Sort order (asc or desc)	desc
symbols	string	No	Comma-separated symbols	DI,TALABAT
sectors	string	No	Comma-separated sectors	IT,BFSI
industry	string	No	Comma-separated industries	Software,Banking
market_cap	string	No	Market cap filter	large,mid,small
stock_score	string	No	Stock score filter	90-100,80-89
analyst_recommendation	string	No	Analyst rating filter	strong_buy,buy
screener	string	No	Screener name	value
pattern	string	No	Pattern within screener	undervalued

Request Examples

Basic Stock List:

```
curl -X GET "https://api.example.com/public/api/v1/stocks?tenant_id=3&page=1&limit=50"
\
  -H "X-Client-Id: your-client-id" \
```

```
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Filtered Stock List:

```
curl -X GET "https://api.example.com/public/api/v1/stocks?
tenant_id=3&sectors=IT,Pharma&market_cap=large&stock_score=80-
100&sort=score&order=desc&page=1&limit=100" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Stock List with Screener:

```
curl -X GET "https://api.example.com/public/api/v1/stocks?
tenant_id=3&screener=growth&pattern=high_growth&limit=200" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce"
```

Response Format

Success Response (200):

```
{
    "success": true,
    "data": {
        "stocks": [
            {
                "symbol": "TALABAT",
                "name": "TALABAT",
                "sector": "Information Technology",
                "industry": "IT Services & Consulting",
                "ltp": 3200.75,
                "previous_close": 3180.50,
                "percentage_change": 0.64,
                "volume": 850000,
                "market_cap": 12000000000000,
                "score": 92,
                "analyst_rating": "BUY",
                "exchange": "NSE",
                "currency": "INR",
                "fifty_two_week_high": 3300.00,
                "fifty_two_week_low": 2800.00
            }
        ],
        "count": 1250,
        "updated_at": "2024-01-15T10:30:00Z"
    }
}
```

3. POST `/stock/details/{stock_id}`

Get detailed information about a specific stock with configurable data sections.

Path Parameters

Parameter	Type	Required	Description	Example
stock_id	string	Yes	Stock symbol or identifier	DU

Request Body

```json
{
    "tenant_id": 1,
    "sections": [
        {
            "type": "stock_info",
            "params": null
        },
        {
            "type": "candle_chart",
            "params": {
                "resolution": "1D",
                "from": "2024-01-01",
                "to": "2024-01-31",
                "count_back": "30"
            }
        },
        {
            "type": "news_details",
            "params": {
                "limit": 10,
                "page": 1
            }
        }
    ]
}
```

Supported Section Types

Section Type	Description	Required Parameters
stock_info	Basic stock information	None
candle_chart	OHLC price data	resolution, from, to, count_back
period_line_chart	Line chart for time period	period
line_chart	Custom line chart	resolution, from, to, count_back
stats	Stock statistics	None
returns_comparison	Returns comparison	None
stock_score_report	Detailed score report	None
forecasts	Analyst forecasts	None

analyst_rankings	Analyst rankings	None
peer_price_comparison	Peer price comparison	limit, page
peer_ratio_comparison	Peer ratio comparison	limit, page
mutual_fund_ownership	MF holdings	limit, page
news_details	Stock-specific news	limit, page
upcoming_announcements	Corporate actions	limit, page
shareholding_details	Shareholding pattern	None
similar_stocks	Similar stocks	limit, page
about	Company information	None

Section Parameters

Candle Chart Parameters

- resolution : `"1m"`, `"5m"`, `"15m"`, `"30m"`, `"1H"`, `"4H"`, `"1D"`, `"1W"`, `"1M"`
- from : Start date in `YYYY-MM-DD` format
- to : End date in `YYYY-MM-DD` format
- count_back : Number of periods to go back from current date

Pagination Parameters

- limit : Number of items per page (default: 50, max: varies by section)
- page : Page number (default: 1)
- ids : Array of specific IDs to filter (for certain sections)

Request Examples

Basic Stock Info:

```
curl -X POST "https://api.example.com/public/api/v1/stock/details/DU" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce" \
  -H "Content-Type: application/json" \
  -d '{
    "tenant_id": 3,
    "sections": [
      {
        "type": "stock_info"
      }
    ]
  }'
```

Stock with Candle Data:

```
curl -X POST "https://api.example.com/public/api/v1/stock/details/TCS" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce" \
  -H "Content-Type: application/json" \
```

```
    -d '{
    "tenant_id": 1,
    "sections": [
      {
        "type": "stock_info"
      },
      {
        "type": "candle_chart",
        "params": {
          "resolution": "1D",
          "from": "2024-01-01",
          "to": "2024-01-31"
        }
      }
    ]
  }'
```

Comprehensive Stock Details:

```
curl -X POST "https://api.example.com/public/api/v1/stock/details/DU" \
  -H "X-Client-Id: your-client-id" \
  -H "X-Signature: your-signature" \
  -H "X-Nonce: your-nonce" \
  -H "Content-Type: application/json" \
  -d '{
    "tenant_id": 1,
    "sections": [
      {
        "type": "stock_info"
      },
      {
        "type": "candle_chart",
        "params": {
          "resolution": "1H",
          "count_back": "100"
        }
      },
      {
        "type": "news_details",
        "params": {
          "limit": 5,
          "page": 1
        }
      },
      {
        "type": "peer_price_comparison",
        "params": {
          "limit": 10,
          "page": 1
        }
      },
      {
```

```
          "type": "mutual_fund_ownership",
          "params": {
            "limit": 20,
            "page": 1
          }
        }
      ]
    }'
```

Response Format

Success Response (200):

```
{
    "success": true,
    "data": {
        "stock_id": "DU",
        "sections": [
            {
                "type": "stock_info",
                "data": {
                    "symbol": "DU",
                    "name": "DU",
                    "sector": "Oil, Gas & Consumable Fuels",
                    "industry": "Oil & Gas - Refining & Marketing",
                    "isin": "INE002A01018",
                    "exchange": "DFM",
                    "listing_date": "1996-01-01",
                    "market_cap": 1650000000000,
                    "enterprise_value": 1800000000000,
                    "outstanding_shares": 6760000000,
                    "face_value": 10,
                    "book_value": 450.75,
                    "dividend_yield": 0.35,
                    "eps": 85.50,
                    "pe_ratio": 28.65,
                    "pb_ratio": 2.15,
                    "roe": 7.85,
                    "debt_to_equity": 0.45,
                    "current_ratio": 1.25,
                    "quick_ratio": 0.85,
                    "promoter_holding": 50.25,
                    "fii_holding": 25.75,
                    "dii_holding": 15.50,
                    "public_holding": 8.50
                }
            },
            {
                "type": "candle_chart",
                "data": {
                    "candles": [
                        {
                            "timestamp": "2024-01-15T09:15:00Z",
```

```
                "open": 2420.00,
                "high": 2450.00,
                "low": 2415.00,
                "close": 2445.00,
                "volume": 125000
            }
        ],
        "meta": {
            "resolution": "1D",
            "from": "2024-01-01",
            "to": "2024-01-31",
            "total_candles": 22
        }
      }
    }
  ]
 }
}
```

Error Handling

Common HTTP Status Codes

Status Code	Description	Common Causes
200	Success	Request processed successfully
400	Bad Request	Invalid parameters, missing required fields
401	Unauthorized	Invalid client ID or signature
500	Internal Server Error	Server-side processing error

Error Response Format

400 Bad Request:

```
{
    "success": false,
    "error": {
        "code": "400",
        "message": "Invalid request parameters"
    }
}
```

400 Invalid Tenant:

```
{
    "success": false,
    "error": {
        "code": "400",
        "message": "Invalid tenant_id"
```

```
    }
}
```

400 Validation Error:

```
{
    "success": false,
    "error": {
        "code": "400",
        "message": "no params found for section candle_chart"
    }
}
```

500 Internal Server Error:

```
{
    "success": false,
    "error": {
        "code": "500",
        "message": "Something went wrong"
    }
}
```

Best Practices

Authentication

1. Generate a new nonce for each request
2. Use HTTPS for all requests
3. Store client secrets securely
4. Implement retry logic with new nonce on authentication failures

Rate Limiting

- Be mindful of pagination limits (max 3000 items per page)
- Implement proper pagination for large datasets
- Cache responses when appropriate

Error Handling

1. Check response status codes
2. Parse error messages for specific validation issues
3. Implement exponential backoff for retries
4. Log errors for debugging

Data Handling

1. Handle empty responses gracefully
2. Check for null/undefined values in responses
3. Be aware of timezone differences in timestamps
4. Validate data types before processing

Troubleshooting

Common Issues

"unknown client_id" Error:

- Verify your client ID is registered in the server configuration
- Ensure you're using the correct client ID in the header

"invalid signature" Error:

- Check that your client secret matches the server configuration
- Verify the message format: `{client_id}:{nonce}`
- Ensure you're using SHA256 HMAC, not other algorithms

Empty Results:

- Verify your search parameters
- Check if the stock symbol exists
- Ensure tenant_id is valid

500 Errors:

- Check server logs for detailed error information
- Verify all required parameters are provided
- Contact support if issue persists

Support

For technical support or questions about this API, please contact the development team or refer to the internal documentation.

Last Updated: January 2025 **Version:** 1.0.0