Budget Tracker API Documentation

Base URL

```
http://localhost:3001/api
```

Authentication

Currently, the API does not require authentication. In production, add JWT-based authentication.

Response Format

All API responses follow this format:

json

```json
{
  "success": true,
  "data": {}, // Response data
  "message": "Optional message",
  "meta": {
    "timestamp": "2023-01-01T00:00:00.000Z"
  }
}
```

Error responses:

json

```json
{
  "success": false,
  "error": "Error code",
  "message": "Human readable error message",
  "errors": [] // Optional validation errors
}
```

Endpoints

Categories

GET /api/categories

Get all categories with optional filtering

Query Parameters:

- `type` (string): Filter by type ('income' or 'expense')
- `is_active` (boolean): Filter by active status
- `parent_id` (string): Filter by parent category
- `include_children` (boolean): Include child categories

GET /api/categories/:id

Get category by ID

POST /api/categories

Create new category

Request Body:

```json
{
  "name": "string",
  "type": "income | expense",
  "color": "#FF0000",
  "icon": "string",
  "description": "string (optional)",
  "parent_id": "uuid (optional)"
}
```

PUT /api/categories/:id

Update category

DELETE /api/categories/:id

Delete category

POST /api/categories/bulk

Bulk operations on categories

Request Body:

```json
{
  "action": "create | update | delete",
  "categories": []
}
```

Transactions

GET /api/transactions

Get transactions with pagination and filtering

Query Parameters:

- `page` (number): Page number (default: 1)
- `limit` (number): Items per page (default: 20, max: 100)
- `type` (string): Filter by type ('income' or 'expense')
- `category_id` (string): Filter by category
- `start_date` (string): Start date (YYYY-MM-DD)
- `end_date` (string): End date (YYYY-MM-DD)
- `min_amount` (number): Minimum amount
- `max_amount` (number): Maximum amount
- `search` (string): Search in description
- `sort` (string): Sort field (date, amount, description, created_at)
- `order` (string): Sort order (asc, desc)
- `include_category` (boolean): Include category details

GET /api/transactions/:id

Get transaction by ID

POST /api/transactions

Create new transaction

Request Body:

```json
{
  "type": "income | expense",
  "amount": 100.00,
  "description": "string",
  "category_id": "uuid",
  "date": "YYYY-MM-DD"
}
```

PUT /api/transactions/:id

Update transaction

DELETE /api/transactions/:id

Delete transaction

POST /api/transactions/bulk

Bulk operations on transactions (max 50)

GET /api/transactions/summary

Get transaction summary with date range

Query Parameters:

- `start_date` (string): Start date
- `end_date` (string): End date

GET /api/transactions/search

Search transactions

Query Parameters:

- `query` (string): Search query (required)
- `limit` (number): Max results (default: 20)

Budgets

GET /api/budgets

Get budgets with filtering

Query Parameters:

- `page` (number): Page number
- `limit` (number): Items per page
- `category_id` (string): Filter by category
- `period` (string): Filter by period (weekly, monthly, yearly)
- `is_active` (boolean): Filter by active status
- `include_category` (boolean): Include category details
- `include_progress` (boolean): Include spending progress
- `overspent_only` (boolean): Only overspent budgets
- `sort` (string): Sort field
- `order` (string): Sort order

GET /api/budgets/:id

Get budget by ID

Query Parameters:

- `include_progress` (boolean): Include spending progress

POST /api/budgets

Create new budget

Request Body:

```json
{
  "category_id": "uuid",
  "budget_amount": 1000.00,
  "period": "weekly | monthly | yearly",
  "start_date": "YYYY-MM-DD",
  "end_date": "YYYY-MM-DD (optional)"
}
```

PUT /api/budgets/:id

Update budget

DELETE /api/budgets/:id

Delete budget

GET /api/budgets/alerts

Get budget alerts

Query Parameters:

- `approaching_threshold` (number): Threshold percentage (default: 80)
- `high_threshold` (number): High alert threshold (default: 95)

GET /api/budgets/summary

Get budget summary

POST /api/budgets/bulk

Bulk operations on budgets

Analytics

GET /api/analytics/dashboard

Get comprehensive dashboard summary

GET /api/analytics/spending-insights

Get spending insights

Query Parameters:

- `period` (string): Time period (week, month, quarter, year, custom)
- `start_date` (string): Start date (for custom period)
- `end_date` (string): End date (for custom period)
- `category_ids` (array): Filter by categories

GET /api/analytics/budget-performance

Get budget performance analysis

GET /api/analytics/trends

Get trends analysis

Query Parameters:

- `period` (string): Time period
- `granularity` (string): Data granularity (daily, weekly, monthly)
- `include_predictions` (boolean): Include predictions
- `category_ids` (array): Categories to analyze

GET /api/analytics/health-score

Get financial health score

GET /api/analytics/comparison

Compare two periods

Query Parameters:

- `period` (string): Period to analyze
- `compare_previous_period` (boolean): Compare with previous period

GET /api/analytics/overview

Get comprehensive analytics overview

Import/Export

GET /api/import-export/config

Get import/export configuration and limits

POST /api/import-export/import

Import data from file

Form Data:

- `file`: CSV or Excel file
- `type`: Data type (transactions, categories, budgets, full)
- `validateData`: Validate before import
- `skipDuplicates`: Skip duplicate entries
- `updateExisting`: Update existing entries

GET /api/import-export/export

Export data to file

Query Parameters:

- `format` (string): Export format (csv, xlsx)
- `type` (string): Data type to export
- `start_date` (string): Filter start date
- `end_date` (string): Filter end date
- `includeHeaders` (boolean): Include headers
- `includeMetadata` (boolean): Include metadata

GET /api/import-export/template/:type

Download import template

Parameters:

- `type`: Template type (transactions, categories, budgets)

Query Parameters:

- `format` (string): File format (csv, xlsx)
- `includeExamples` (boolean): Include example data

POST /api/import-export/validate

Validate uploaded file structure

Cache Management (Development Only)

GET /api/cache/stats

Get cache statistics

DELETE /api/cache/:type?

Clear cache (optionally by type)

Rate Limiting

- General endpoints: 100 requests per minute
- Import endpoints: 5 requests per minute
- Export endpoints: 10 requests per minute

Error Codes

- `400` - Bad Request: Invalid parameters
- `401` - Unauthorized: Authentication required
- `403` - Forbidden: Insufficient permissions
- `404` - Not Found: Resource not found
- `422` - Unprocessable Entity: Validation errors
- `429` - Too Many Requests: Rate limit exceeded
- `500` - Internal Server Error: Server error

Pagination

Paginated responses include:

json

```
{
  "data": {
    "items": [],
    "pagination": {
      "page": 1,
      "limit": 20,
      "total": 100,
      "pages": 5,
      "has_next": true,
      "has_prev": false
    }
  }
}
```

Data Validation

All endpoints validate input data and return detailed validation errors:

```json
{
  "success": false,
  "error": "VALIDATION_ERROR",
  "message": "Validation failed",
  "errors": [
    {
      "field": "amount",
      "message": "Amount must be greater than 0"
    }
  ]
}
```

CORS

CORS is enabled for all origins in development. Configure specific origins in production.

Compression

Responses are compressed using gzip for better performance.

Caching

The API implements response caching for GET requests:

- Dashboard: 1 minute
- Analytics: 5 minutes

- Budgets: 3 minutes
- Transactions: 2 minutes
- Categories: 10 minutes

Cache is automatically invalidated on data modifications.