Trello REST API Documentation

Overview

The Trello REST API provides programmatic access to Trello's features and data. It allows you to create applications that can:

- Read and write data to boards, lists, and cards
- Create automations and integrations
- Manage members and organizations
- Handle attachments and custom fields
- Set up webhooks for real-time updates

Authentication

Trello uses API keys and tokens for authentication. There are two main authentication methods:

1. API Key + Token Authentication

```
Authorization: OAuth oauth_consumer_key="YOUR_API_KEY", oauth_token="USER_TOKEN"
```

2. Query Parameters

```
?key=YOUR_API_KEY&token=USER_TOKEN
```

To obtain authentication credentials:

1. Get your API key from your [Trello account settings](https://trello.com/app-key)

2. Generate a token by authorizing your application

Common Objects

Board Object

```json
{
  "id": "string",
  "name": "string",
  "desc": "string",
  "closed": boolean,
  "idOrganization": "string",
  "url": "string",
  "shortUrl": "string"
}
```

List Object

```json
{
  "id": "string",
  "name": "string",
  "closed": boolean,
  "idBoard": "string",
```

```
  "pos": number
}
```

Card Object

```json
{
  "id": "string",
  "name": "string",
  "desc": "string",
  "closed": boolean,
  "idList": "string",
  "idBoard": "string",
  "due": "datetime",
  "dueComplete": boolean
}
```

API Endpoints

Boards

Get a Board

```http
GET /1/boards/{id}
```

Parameters:

- `id` (required): The ID of the board to retrieve

Create a Board

```http
POST /1/boards
```

Parameters:

- `name` (required): The name of the board
- `defaultLists` (optional): Boolean to determine if default lists should be added
- `idOrganization` (optional): The ID of the organization to add board to

Lists

Get Lists on a Board

```http
GET /1/boards/{id}/lists
```

Parameters:

- `id` (required): The ID of the board
- `filter` (optional): Filter by `open`, `closed`, or `all`

Create a List

```http
POST /1/lists
```

```

Parameters:

- `name` (required): The name of the list
- `idBoard` (required): The ID of the board to add list to
- `pos` (optional): Position of list (top, bottom, or positive number)

### Cards

#### Get a Card

```http
GET /1/cards/{id}
```

Parameters:

- `id` (required): The ID of the card to retrieve

Create a Card

```http
POST /1/cards
```

Parameters:

- `name` (required): The name of the card
- `idList` (required): The ID of the list to add card to
- `desc` (optional): The description of the card
- `due` (optional): Due date for the card
```

Organizations

Get an Organization

```http
GET /1/organizations/{id}
```

Parameters:

- `id` (required): The ID or name of the organization
- `fields` (optional): Fields to return about the organization
- `members` (optional): Include member details
- `memberships` (optional): Include membership details
- `boards` (optional): Include board details

Create an Organization

```http
POST /1/organizations
```

Parameters:

- `displayName` (required): The display name for the organization
- `name` (optional): Short name used in the organization's URL
- `desc` (optional): Description of the organization
- `website` (optional): The organization's website
- `prefs/permissionLevel` (optional): 'private' or 'public'

Update an Organization

```http
PUT /1/organizations/{id}
```

Parameters:

- `id` (required): The ID of the organization
- `displayName` (optional): New display name
- `name` (optional): New short name
- `desc` (optional): New description
- `website` (optional): New website URL

Search

Search Trello

```http
GET /1/search
```

Parameters:

- `query` (required): Search query (2 character minimum)
- `idBoards` (optional): Limit search to specific boards
- `idOrganizations` (optional): Limit search to specific organizations
- `idCards` (optional): Limit search to specific cards
- `modelTypes` (optional): Array of types to search: "actions", "boards", "cards", "members", "organizations"

- `board_fields` (optional): Fields to return for board results
- `boards_limit` (optional): Limit on number of boards returned
- `card_fields` (optional): Fields to return for card results
- `cards_limit` (optional): Limit on number of cards returned
- `cards_page` (optional): Page of cards to return
- `card_board` (optional): Include card board information
- `card_list` (optional): Include card list information
- `card_members` (optional): Include card member information
- `card_stickers` (optional): Include card sticker information
- `card_attachments` (optional): Include card attachment information
- `organization_fields` (optional): Fields to return for organization results
- `organizations_limit` (optional): Limit on number of organizations returned
- `member_fields` (optional): Fields to return for member results
- `members_limit` (optional): Limit on number of members returned
- `partial` (optional): Include partial search results

Search Members

```http
GET /1/search/members
```

Parameters:

- `query` (required): Search query for members
- `limit` (optional): Number of members to return (default: 8, maximum: 20)
- `idBoard` (optional): Restrict results to members of a specific board
- `idOrganization` (optional): Restrict results to members of a specific organization

Search Cards

```http
GET /1/search/cards
```

Parameters:

- `query` (required): Search query for cards
- `idBoards` (optional): Array of board IDs to search within
- `idCards` (optional): Array of card IDs to search within
- `idOrganizations` (optional): Array of organization IDs to search within
- `limit` (optional): Number of cards to return (default: 10, maximum: 1000)

Rate Limiting

Trello implements rate limiting to ensure fair usage of the API:

- 100 requests per 10-second interval for each API key
- 100,000 requests per day for each token
- Rate limit headers are included in API responses:
  - `X-Rate-Limit-Api-Key-Remaining`
  - `X-Rate-Limit-Api-Token-Remaining`

Best Practices

1. **Batch Operations**
   - Use batch endpoints when possible
   - Combine multiple operations into single requests

2. **Caching**
   - Cache responses when appropriate
   - Use ETags for conditional requests

3. **Error Handling**
   - Always check HTTP status codes
   - Implement exponential backoff for rate limits
   - Handle common error scenarios gracefully

4. **Security**
   - Never expose API keys or tokens in client-side code
   - Use minimal scope tokens when possible
   - Regularly rotate tokens for security

Error Handling

Common HTTP Status Codes

- `200 OK`: Request successful
- `400 Bad Request`: Invalid parameters
- `401 Unauthorized`: Invalid authentication
- `403 Forbidden`: Valid authentication but insufficient permissions
- `404 Not Found`: Resource not found
- `429 Too Many Requests`: Rate limit exceeded

Error Response Format

```json
{
  "error": {
    "message": "string",
    "code": "string"
  }
}
```

Example Usage

Python
```python
import requests

API_KEY = "your_api_key"
TOKEN = "your_token"
BASE_URL = "https://api.trello.com/1"

# Get board details
board_id = "board_id"
response = requests.get(
    f"{BASE_URL}/boards/{board_id}",
    params={
        "key": API_KEY,
        "token": TOKEN
    }
```

)

board = response.json()

```

### JavaScript

```javascript
const API_KEY = 'your_api_key';
const TOKEN = 'your_token';
const BASE_URL = 'https://api.trello.com/1';

// Create a new card
async function createCard(listId, cardName) {
  const response = await fetch(`${BASE_URL}/cards`, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      name: cardName,
      idList: listId,
      key: API_KEY,
      token: TOKEN,
    }),
  });
  return response.json();
}
```
```

cURL

```bash
# Get all lists on a board
curl -X GET "https://api.trello.com/1/boards/{boardId}/lists?key={API_KEY}&token={TOKEN}"

# Create a new board
curl -X POST "https://api.trello.com/1/boards?name=New%20Board&key={API_KEY}&token={TOKEN}"
```

Additional Resources

- [Trello Developers Portal](https://developer.atlassian.com/cloud/trello/)
- [API Reference](https://developer.atlassian.com/cloud/trello/rest)
- [Developer Terms of Service](https://developer.atlassian.com/cloud/trello/guides/rest-api/api-introduction/)
- [API Changelog](https://developer.atlassian.com/cloud/trello/changelog/)