

Monday.com SDK Guide

 Creator **Zachary Gilk**  Created **Dec 04, 2024, 10:14**  Last updated **Dec 04, 2024, 10:42**

The [Monday.com](#) SDK offers developers the ability to create, customize, and extend functionalities for a wide range of use cases within [Monday.com](#) work operating system. The SDK allows you to build board views, item views, widgets, automation, and integration features using a combination of JavaScript, REST APIs, and a robust client-side SDK. Below, I'll cover key aspects of working with the [Monday.com](#) SDK.

1. Setting Up the Environment

To get started, you'll need to set up your development environment. [Monday.com](#) supports an easy development process through its Apps Framework. You will need:

- **Node.js** (v12 or higher recommended) as it provides a runtime environment for JavaScript code.
- **ngrok** to expose your local server to the public internet for webhook testing.

Begin by creating a new [Monday.com](#) app using the [Monday Apps Framework CLI](<https://github.com/mondaycom/monday-cli>). This tool provides a convenient way to initialize, develop, and deploy your apps within the platform.

To initialize your app, use:

```
npx @mondaydotcomorg/monday-cli init
```

This command will guide you through the process of configuring your app, including adding capabilities such as **Board Views**, **Item Views**, **Widgets**, and **Integrations**.

2. The SDK Client Library

The [Monday.com](#) client SDK is a JavaScript library that allows you to interact directly with data in the [Monday.com](#) environment. To get started, install the SDK into your project:

```
npm install @mondaycom/sdk
```

The SDK provides utility functions that allow your application to fetch data, mutate board items, and interact with [Monday.com](#)'s interface. Here's an example of importing the SDK and connecting it to your app:

```
import mondaySdk from "@mondaycom/sdk";

const monday = mondaySdk();
monday.setToken("YOUR_USER_API_TOKEN");
```

The token can either be an OAuth token or an API token, and it should be used carefully to manage your app's access.

3. App Capabilities

A [Monday.com](#) app can offer a variety of capabilities, such as **views**, **widgets**, **integrations**, and **automations**.

- **Board View & Item View:** Board and Item views are custom UI elements that enhance the user experience. The `monday.execute` method can be used to communicate with [Monday.com](#) from these views. For example, if you want to run a query that retrieves board items:

```
monday.api(`{
  boards(ids: 123456789 {
    items {
      name
      column_values{
        id
        text
      }
    }
  }
}`).then((res) => {
  console.log(res.data);
});
```

- **Widgets:** Widgets can be used in dashboards and offer a rich way to display data from your boards. Widgets interact with data using the `monday.api()` method as well as the React framework, which helps in crafting a modern UI.

- **Integration Recipes:** Integration recipes define specific triggers and actions that your app can perform. These recipes are written in a user-friendly, no-code environment and typically involve three key parts: a **trigger**, a **condition**, and an **action**. For example, a trigger could be "When a new item is created" and the action could be "Send data to a third-party API".

- **Automations:** Automations within [Monday.com](#) involve creating seamless workflows that automatically execute when a trigger occurs. Using the SDK, you can register custom automations by configuring integration files with a trigger and action block.

4. Using GraphQL for Data Retrieval

The [Monday.com](#) API uses **GraphQL** to query data. Unlike traditional REST APIs, GraphQL allows you to define precisely what data you want returned, reducing payload sizes and increasing efficiency. You can use the `monday.api()` method to run GraphQL queries directly from within your application, such as to retrieve items, users, and board configurations.

```
const query = `{
  boards(limit: 1) {
    id
    name
    items {
      id
      name
    }
  }
}`;

monday.api(query).then((res) => {
  console.log(res.data.boards);
});
```

5. OAuth for Authentication

To allow your application to interact with users' [Monday.com](#) accounts, you'll need to use **OAuth 2.0**. [Monday.com](#) provides an OAuth flow for authenticating users securely. This involves:

1. **Redirecting users** to the authorization URL.
2. **Exchanging** the authorization code for an access token.
3. Using the **access token** to perform actions on behalf of the user.

Authentication ensures your app has permissions to interact with users' data, depending on the provided scopes.

6. Utilizing the Storage API

The **Storage API** provided by [Monday.com](#) allows you to store key-value data related to your app. It's useful for persisting app-specific settings, like user preferences or state information:

```
// Storing data
monday.storage.instance.setItem('key', 'value').then((res) => {
  console.log('Data saved successfully', res);
});

// Retrieving data
monday.storage.instance.getItem('key').then((res) => {
  console.log('Data retrieved', res.data.value);
});
```

7. Developing, Testing, and Deploying Your App

During development, **ngrok** is highly useful to make your local environment publicly accessible. You can run your local server and expose it using ngrok, enabling [Monday.com](#) to interact with your app's endpoints for webhooks or integration testing.

After development, apps can be submitted to the **Monday.com Apps Marketplace**. The submission involves providing detailed documentation, a video demo, and a verification process to ensure security and stability.

8. Handling Webhooks

Webhooks allow you to listen for changes within your [Monday.com](#) boards. You can create custom webhooks to react to board events, such as when an item is updated or deleted. The [Monday.com](#) SDK and API make it easy to set up webhooks and direct them to your local server for processing.

9. Best Practices

- **Error Handling:** Ensure you have robust error-handling mechanisms, as API queries and mutations can fail due to connectivity issues, incorrect board IDs, or permission problems.
- **Efficient API Usage:** Limit API usage to reduce costs and avoid hitting rate limits. Use GraphQL to query only necessary fields and avoid redundant calls.
- **Secure Token Management:** Always manage tokens securely. For OAuth tokens, use secure storage and restrict permissions as much as possible.
- **UI/UX Design:** If building board views or widgets, keep the UI intuitive, as the end users are often non-technical users managing their workflows.

Conclusion

The [Monday.com](https://monday.com) SDK provides a versatile set of tools to build powerful customizations within the Monday ecosystem. From querying data using GraphQL to developing sophisticated widgets and board views, the SDK empowers developers to extend [Monday.com](https://monday.com)'s functionalities significantly. Whether you're creating simple automation workflows or sophisticated apps, understanding the SDK capabilities will help leverage the platform to its fullest potential.