

readme.md for @api.global/sdk

an sdk package for api.global

Install

To install `@api.global/sdk`, you need to have Node.js installed on your machine. This package is available through the npm registry. Installation can be done using the npm or yarn command line tools.

```
npm install @api.global/sdk --save
```

or

```
yarn add @api.global/sdk
```

Usage

Using `@api.global/sdk` requires an understanding of TypeScript and asynchronous programming concepts in Node.js. Below, we will outline a series of use cases to demonstrate how to utilize the SDK effectively in your projects.

Setting Up Your Project

Before proceeding with the usage examples, ensure your TypeScript configuration (`tsconfig.json`) is correctly set up for an ES Module syntax project.

Importing the SDK

Once installed, you can import the entire SDK or specific parts of it into your project files:

```
import * as ApiGlobalSdk from '@api.global/sdk';
```

Or import specific classes if you're only interested in a subset of functionalities:

```
import { AAgHandler, AgEnvironment, AuthInfo } from '@api.global/sdk';
```

Creating an Environment Handler

`AgEnvironment` is an abstract class that manages the provision of environment variables. A concrete implementation could look like this:

```
import { AgEnvironment } from '@api.global/sdk';

class MyEnvironment extends AgEnvironment {
  private envVariables: Map<string, string> = new Map([
    ['API_KEY', 'your-api-key-here'],
    // Add other environment variables
  ]);

  public async getEnvVar(envVarName: string): Promise<string> {
    return this.envVariables.get(envVarName) || '';
  }
}

const myEnvironment = new MyEnvironment();
```

Implementing an Authorization Handler

The SDK requires implementing an authorization handler by extending the `AuthInfo` class. This class manages authentication and potential security flags for incoming requests.

```
import { AuthInfo } from '@api.global/sdk';

class MyAuthInfo extends AuthInfo<any> {
  public authenticated: boolean = false;
  public potentiallyMalicious: boolean = false;
  public claim: any;
```

```
constructor(claim: any) {
    super();
    this.claim = claim;
    // Implement logic to set 'authenticated' and 'potentiallyMalicious'
}
}
```

Creating an API Handler

Create an API handler by extending `AAgHandler`. This handler will manage API requests and connect them to your business logic.

```
import { AAgHandler, IRequirementResult } from '@api.global/sdk';
import { TypedRouter } from '@apiglobal/typedrequest';

class MyApiHandler extends AAgHandler<any> {
    public slug: string = 'myApi';
    public typedrouter: TypedRouter = new TypedRouter();

    public async checkRequirements(): Promise<IRequirementResult> {
        // Implement checks for your requirements here
        return { allOk: true, reason: '' };
    }

    constructor(agEnvironment: AgEnvironment) {
        super(agEnvironment);
        // Additional constructor logic
    }

    public async start(): Promise<void> {
        // Implement starting logic here
    }

    public async stop(): Promise<void> {
        // Implement stopping logic here
    }
}

// Usage
```

```
const myApiHandler = new MyApiHandler(myEnvironment);
```

Starting Your API Handler

Finally, start your API handler as part of your application's boot-up sequence.

```
async function startApplication() {
  try {
    await myApiHandler.start();
    console.log('API handler started successfully.');
```

```
  } catch (error) {
    console.error('Failed to start API handler:', error);
  }
}
```

```
startApplication();
```

Advanced Use Cases

- **Custom AuthInfo Implementations:** Depending on your application's authentication mechanism (e.g., OAuth2, JWT), you may need to create more sophisticated `AuthInfo` implementations.
- **Feature Flags and Environment Variables:** Use `AgEnvironment` to manage feature flags or other configuration that might change between different environments.
- **TypedRequest Integrations:** Leverage the `TypedRouter` from `@apiglobal/typedrequest` to create strongly typed endpoints and request handlers within your `AAgHandler` implementation.

Conclusion

The `@api.global/sdk` offers powerful abstractions to create well-structured, environment-aware, and secure API handlers for Node.js applications. By extending its base classes and implementing the required methods, you can quickly build scalable and maintainable backend services.

Remember, the key to effectively using `@api.global/sdk` is understanding its core concepts and how they fit into your application's architecture. With this knowledge, you can extend the SDK to meet your specific needs, ensuring a robust and flexible API layer.

License and Legal Information

This repository contains open-source code that is licensed under the MIT License. A copy of the MIT License can be found in the [license](#) file within this repository.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH and are not included within the scope of the MIT license granted herein. Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines, and any usage must be approved in writing by Task Venture Capital GmbH.

Company Information

Task Venture Capital GmbH
Registered at District court Bremen HRB 35230 HB, Germany

For any legal inquiries or if you require further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

Revision #7

Created 2026-03-28 10:48:08 UTC by foss.global Team

Updated 2026-03-28 12:13:16 UTC by foss.global Team