

readme.md for @push.rocks/smartinteract

smart cli interaction

Install

To install `@push.rocks/smartinteract`, use npm or yarn as follows:

```
npm install @push.rocks/smartinteract --save
# or using yarn
yarn add @push.rocks/smartinteract
```

This will add it to your project's `node_modules` folder and list it as a dependency in your project's `package.json` file.

Usage

The `@push.rocks/smartinteract` module is designed to facilitate smart command-line interactions by providing a flexible way to query input from users in a structured manner. It leverages TypeScript for enhanced development experience, offering strong types and IntelliSense in supported editors.

Getting Started

To begin, import `SmartInteract` from the module:

```
import { SmartInteract } from '@push.rocks/smartinteract';
```

Creating and Running Interactions

You can specify questions for user interaction as shown below:

```
const smartInteract = new SmartInteract([
  {
    name: 'userName',
    type: 'input',
    message: 'What is your name?',
    default: 'John Doe',
  },
  {
    name: 'userConfirmation',
    type: 'confirm',
    message: 'Do you agree with the terms?',
    default: false,
  }
]);
```

`SmartInteract` supports various question types such as 'input', 'confirm', 'list', 'rawlist', 'checkbox', 'password', and 'editor'. This allows for a diverse range of user inputs from simple text inputs to selections from a list.

Adding Additional Questions Dynamically

You can dynamically add more questions to the interaction:

```
smartInteract.addQuestions([
  {
    name: 'userHobby',
    type: 'input',
    message: 'What is your favorite hobby?',
    default: 'Programming',
  }
]);
```

Running the Interaction Queue

Execute the queued interaction and process the responses:

```
smartInteract.runQueue()
  .then(answerBucket => {
```

```
const userResponse = {
  name: answerBucket.getAnswerFor('userName'),
  agreed: answerBucket.getAnswerFor('userConfirmation'),
  hobby: answerBucket.getAnswerFor('userHobby')
};

console.log(userResponse);
// Output: { name: 'John Doe', agreed: true/false, hobby: 'Programming'}
});
```

Direct Questioning

For immediate prompting without queuing, you can utilize the `askQuestion` method:

```
smartInteract.askQuestion({
  name: 'urgentQuery',
  type: 'confirm',
  message: 'Do you need to save changes?',
  default: true,
})
.then(answerObject => {
  console.log(answerObject);
  // Output: { name: 'urgentQuery', value: true/false }
});
```

Advanced Usage

You can customize validations, provide choice selections for 'list', 'rawlist', 'expand', and 'checkbox' types, and much more according to the [Inquirer.js](#) module which `@push.rocks/smartinteract` leverages internally.

Conclusion

`@push.rocks/smartinteract` offers an elegant and robust solution for CLI interactions in Node.js projects. Its integration of TypeScript enhances code quality and developer workflow. Whether for simple prompts or complex inquisitive flows, it provides the tools necessary for obtaining and managing user inputs efficiently.

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 #3

Created 2026-03-28 11:10:43 UTC by foss.global Team

Updated 2026-03-28 12:17:32 UTC by foss.global Team