

Create a simple application that can interact with an existing Ethereum smart contract via web3.js.

Requirements:

1. Setting up the environment for Node.js and Web3.js:
2. Connecting to the Ethereum network:
  - Create an index.js file and initialize web3 connecting to the Ethereum network via Infura (or whatever RPC URL you provide).
3. Interaction with smart contract:
  - Choose an existing smart contract or create your own simple Solidity contract.
  - Get the ABI and address of this contract and use them to initialize the contract via web3 in your application.
  - Write a function that calls one of the smart contract functions and prints the result to the console.
  - Write a function that listens for a specific contract event and prints information about each event to the console.
4. Testing:
  - Write a set of basic unit tests to verify that features in your application work correctly.

Additional requirements:

1. Use of TypeScript is recommended.

As a result, we expect to receive the code of your application, including:

- index.js file with the application code.
- Files with tests.
- README file with instructions for installing, configuring and running the application.

Evaluation criteria:

- Code operability.
- Adhering to clean code principles and JavaScript and Node.js best practices.
- Completeness and correctness of the tests.
- Correctness and completeness of documentation.

Notes:

- Make sure you handle possible errors and exceptions.
- Feel free to use any additional libraries or tools you deem necessary.
- Be prepared to explain your decisions and approaches during the code review stage.