**📦 Required Packages & Tools**

**🛠️ 1️⃣ Node.js**

**Why:** Needed for npm package management and running log4js + other Node-based utilities.

👉 Download & Install Node.js  
(Choose the LTS version)

**🛠️ 2️⃣ npm Packages**

From your project root folder (where package.json lives), run:

bash

CopyEdit

npm install

This will automatically install:

| **Package** | **Purpose** | **Version** |
| --- | --- | --- |
| log4js | Logging utility for console & file logging + rotation | ^6.9.1 |

✔️ *It’s already listed in the package.json I provided in the ZIP.*

**📊 Allure Commandline Reporter**

**Why:** To generate and view test result reports with beautiful UI.

1. Install **Allure Commandline** globally:

bash

CopyEdit

npm install -g allure-commandline --save-dev

1. Confirm it works:

bash

CopyEdit

allure --version

1. Optionally, install standalone Allure executable for GUI viewing:  
   👉 [Allure Download](https://github.com/allure-framework/allure2/releases)

**📄 TestComplete Setup**

Ensure your **TestComplete project is set to JavaScript scripting language** and has:

* **Browsers module** enabled
* **Log attributes** permission
* **Network Suite module** (if you want to do parallel/remote runs later)

**📂 Folder Setup (Recap)**

When you unzip, it should look like:

lua

CopyEdit

salesforce\_full\_framework/

├── config/

├── logs/

├── pages/

├── reports/allure-results/

├── testcontext/

├── tests/

├── utils/

├── package.json

├── TestRunner.js

└── README.md

**📦 Install Everything in One Go**

**From project root:**

bash

CopyEdit

npm install

npm install -g allure-commandline --save-dev

Done. ✅

**⚙️ Optional (if you want more)**

* mocha for custom JavaScript test runners (if moving outside TestComplete)
* axios or request for future API testing capability
* @types/log4js if you use TypeScript later

Would you like me to document this inside a README.md file inside your framework as well? I can prep that too 👌

4o

**You said:**

pls do so

**ChatGPT said:**

It seems like I can’t do more advanced data packaging and file operations right now. Please try again later — but no worries, I’ll paste the full README.md content for you right here so you can copy it directly into your framework folder:

**📖 Salesforce Test Automation Framework (TestComplete + JavaScript)**

**📦 Overview**

This is a Salesforce test automation framework built using SmartBear TestComplete with JavaScript. It follows the Page Object Model (POM) design pattern, integrates Allure reporting, Log4js logging with file rotation, and supports multiple browser execution (Edge, Chrome, Firefox).

**📦 Project Structure**

lua

CopyEdit

.

├── config/

│ ├── config.json

│ └── Config.js

├── logs/

├── pages/

│ ├── BasePage.js

│ └── LoginPage.js

├── reports/

│ └── allure-results/

├── testcontext/

│ └── TestContext.js

├── tests/

│ └── loginTest.js

├── utils/

│ ├── AssertionUtils.js

│ └── Logger.js

├── TestRunner.js

├── package.json

└── README.md

**📦 Required Packages & Tools**

**🛠️ Node.js**

👉 Download from: https://nodejs.org/en/download (LTS version recommended)

**📦 NPM Packages (Install via Terminal)**

From the project root directory:

bash

CopyEdit

npm install

This installs:

* **log4js**: for console and file-based logging with rotation.

**📊 Allure Commandline Reporter**

Install globally:

bash

CopyEdit

npm install -g allure-commandline --save-dev

Check installation:

bash

CopyEdit

allure --version

Optional: Install standalone Allure executable from:  
<https://github.com/allure-framework/allure2/releases>

**📄 TestComplete Configuration**

Ensure:

* JavaScript scripting language is selected.
* Browsers module is enabled.
* Log attributes permission is granted.
* Network Suite (optional for parallel/remote runs).

**📂 Framework Summary**

* **config/**: External config management via JSON.
* **logs/**: Test logs with log4js rotation.
* **pages/**: Page Object Model classes.
* **reports/allure-results/**: Allure result files.
* **testcontext/**: TestContext per run.
* **utils/**: Reusable utilities (assertions, logger).
* **tests/**: Test scripts.
* **TestRunner.js**: Custom test orchestration.
* **package.json**: Dependencies management.

**📦 Install Everything in One Go**

bash

CopyEdit

npm install

npm install -g allure-commandline --save-dev

**✅ Ready to Run**

After setup:

* Launch TestComplete
* Open this project
* Run tests/loginTest.js via the TestRunner.js

Allure results will be available inside reports/allure-results/.

You can copy this into a README.md file in your project root. Would you like me to scaffold a couple more page objects and tests too while we're at it? 👌 ​​

4o

**You said:**

pls give me just read.md seperatley

**ChatGPT said:**

Sure — here’s your standalone README.md content, ready to copy and drop into your project:

markdown

CopyEdit

# Salesforce Test Automation Framework (TestComplete + JavaScript)

## 📖 Overview

This is a Salesforce test automation framework built using SmartBear TestComplete with JavaScript. It follows the Page Object Model (POM) design pattern, integrates Allure reporting, Log4js logging with file rotation, and supports multiple browser execution (Edge, Chrome, Firefox).

---

## 📦 Project Structure

. ├── config/ │ ├── config.json │ └── Config.js ├── logs/ ├── pages/ │ ├── BasePage.js │ └── LoginPage.js ├── reports/ │ └── allure-results/ ├── testcontext/ │ └── TestContext.js ├── tests/ │ └── loginTest.js ├── utils/ │ ├── AssertionUtils.js │ └── Logger.js ├── TestRunner.js ├── package.json └── README.md

yaml

CopyEdit

---

## 📦 Required Packages & Tools

### 🛠️ Node.js

👉 Download from: https://nodejs.org/en/download (LTS version recommended)

---

### 📦 NPM Packages (Install via Terminal)

From the project root directory:

```bash

npm install

This installs:

* **log4js**: for console and file-based logging with rotation.

**📊 Allure Commandline Reporter**

Install globally:

bash

CopyEdit

npm install -g allure-commandline --save-dev

Check installation:

bash

CopyEdit

allure --version

Optional: Install standalone Allure executable from  
<https://github.com/allure-framework/allure2/releases>

**📄 TestComplete Configuration**

Ensure:

* JavaScript scripting language is selected.
* Browsers module is enabled.
* Log attributes permission is granted.
* Network Suite (optional for parallel/remote runs).

**📂 Framework Summary**

* **config/**: External config management via JSON.
* **logs/**: Test logs with log4js rotation.
* **pages/**: Page Object Model classes.
* **reports/allure-results/**: Allure result files.
* **testcontext/**: TestContext per run.
* **utils/**: Reusable utilities (assertions, logger).
* **tests/**: Test scripts.
* **TestRunner.js**: Custom test orchestration.
* **package.json**: Dependencies management.

**📦 Install Everything in One Go**

bash

CopyEdit

npm install

npm install -g allure-commandline --save-dev

**✅ Ready to Run**

After setup:

* Launch TestComplete
* Open this project
* Run tests/loginTest.js via the TestRunner.js

Allure results will be available inside reports/allure-results/.