





Task 3 phase 3

 Status	Done
 People	 Abdul Rehman
 Dates	

Add Linters and SAST tools to your project

Add and configure Linter to your project

Linters are software tools that analyze source code to identify issues, errors, and potential problems that might otherwise go unnoticed. Linters typically check code for compliance with coding standards, style conventions, and best practices, as well as for syntax errors, unused variables, and other issues that can cause bugs or performance problems. Linters can be used for many programming languages and can be integrated with development environments and build systems to provide immediate feedback to developers as they write code. Linters are an essential part of the modern software development process, helping to improve code quality, maintainability, and reliability.

To add linters to CodeceptJS project, you can follow these steps:

1. Install the linters you want to use, such as ESLint or Prettier, using npm or yarn. For example, to install ESLint, you can run:

```
npm install eslint --save-dev
```

2. Next, install `eslint-plugin-codeceptjs/`

```
npm install eslint-plugin-codeceptjs --save-dev
```

3. Initialize the linter configuration file. For example, to initialize ESLint, you can run:

```
npx eslint --init
```

This will prompt you to answer some questions to set up your configuration file. You can choose the options that best suit your project's needs.

4. Update the Codecept.config.js file

```
module.exports = {  
  .....  
  lint: {  
    enabled: true,  
    files: ["./*test.js"],  
    options: {  
      eslint: {  
        configFile: "./.eslintrc.json",  
      },  
    },  
  },  
  .....  
};
```

5. Usage

The simplest way to use this plugin is to add the `/recommended` config to the extends section of your `.eslintrc` configuration file:

```
{  "extends": ["plugin:codeceptjs/recommended"] }
```

Alternatively, add `codeceptjs` to the plugins section of your `.eslintrc` configuration file. You can omit the `eslint-plugin-` prefix:

```
{  "plugins": [  "codeceptjs"  ],  
  "env": {  
    "codeceptjs/codeceptjs": true  
  }  
}
```

Then configure the rules you want to use under the rules section.

```
{
  "rules": {
    "codeceptjs/no-actor-in-scenario": 2
  }
}
```

3. Run

```
npx eslint test-file-name
```

Link of video :

<https://drive.google.com/file/d/1h-qwLUOpnnd4BafZdfbSihBPfN18P2Cm/view?usp=sharing>

Add some SAST tools according to framework to application.

I also used ESLint as SAST tool in my project because it can be used according to my research.