

Собираем Javascript

- Пакетный менеджер - npm
- Сборщик - webpack
- Транспайлер - babel
- Unit-тесты - jest
- Статический анализ кода - eslint
- Скриптинг, утилитинг, серверинг - node.js
- Backend - express

Собираем Javascript

Пакетные менеджеры

- bower
- CPM
- yarn
- npm

Собираем Javascript

Пакетные менеджеры

- bower
- CPM
- yarn
- npm



Собираем Javascript

NPM

- `npm install <pkg> [--global]`
- `npm version <version>`
- `npm view <pkg> versions`
- `npm ci`
- `npm publish`
- `npm init`

Собираем Javascript

NPM

- `npm install <pkg> [--global]` ➡ установка пакета
- `npm version <version>` ➡ установка версии с коммитом и тегом
- `npm view <pkg> versions` ➡ проверка доступных версий
- `npm ci` ➡ установка по `package-lock.json`
- `npm publish` ➡ публикация пакета в registry
- `npm init` ➡ инициализация `package.json`

Собираем Javascript

package.json

```
{
  "name": "my-perfect-project",
  "version": "0.1.0",
  "scripts": {
    "start": "npx cross-env NODE_ENV=development node ./dev-server.js",
    "test": "npx jest -ci",
    "build": "node ./build.js",
    "eslint": "npx eslint ./src --quiet"
  },
  "dependencies": { ... },
  "devDependencies": { ... }
}
```

Собираем Javascript

package.json

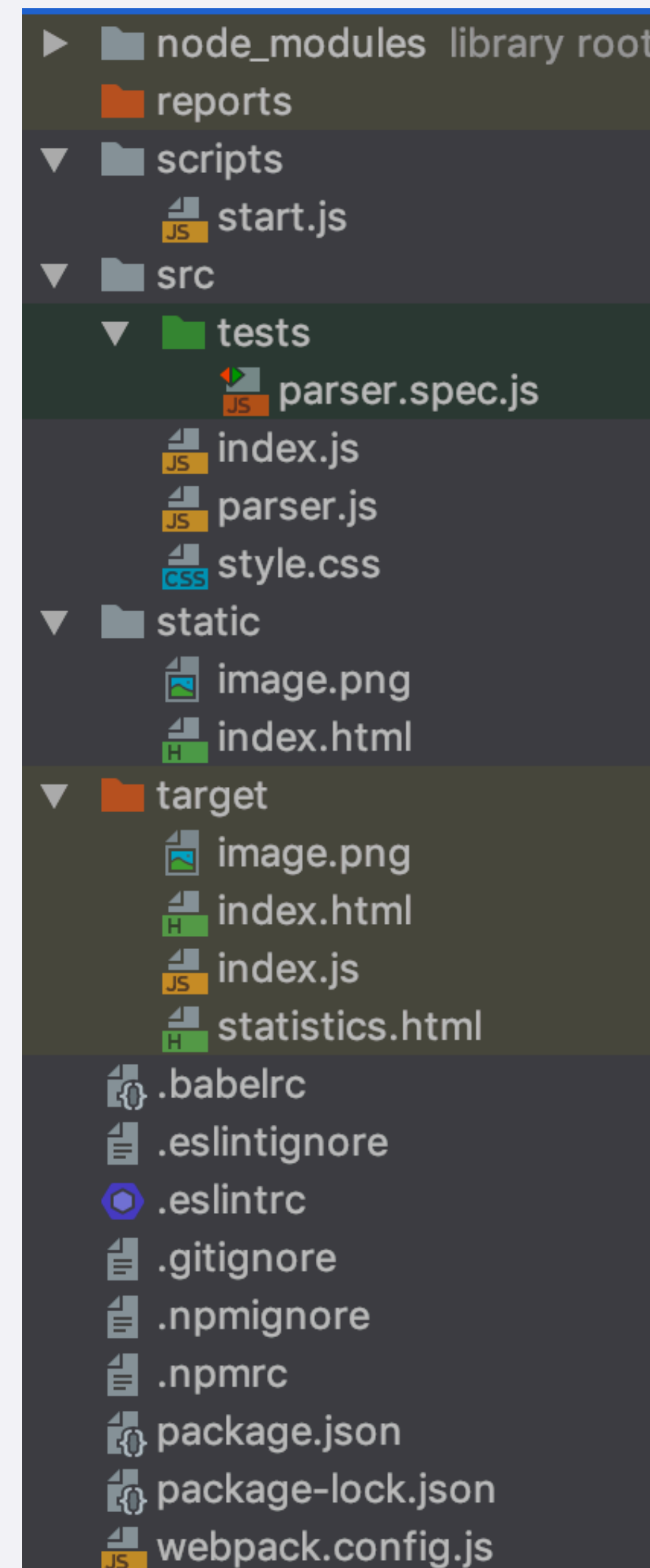
```
{  
  "name": "my-perfect-project",  
  "version": "0.1.0",  
  "scripts": {  
    "start": "npx cross-env NODE_ENV=development node ./dev-server.js",  
    "test": "npx jest -ci",  
    "build": "node ./build.js",  
    "eslint": "npx eslint ./src --quiet"  
  },  
  "dependencies": { ... },  
  "devDependencies": { ... }  
}
```

Создайте новый проект с командой `npm init`

Собираем Javascript

Структура проекта

- Зависимости
- Отчеты
- Скрипты и конфиги
- Тесты
- Статика
- 📁 Исходники 📁



Собираем Javascript

Сборщики

- gulp
- grunt
- webpack
- neutrino

Собираем Javascript

Сборщики

- gulp
- grunt
- webpack
- neutrino



Собираем Javascript

Сборщики

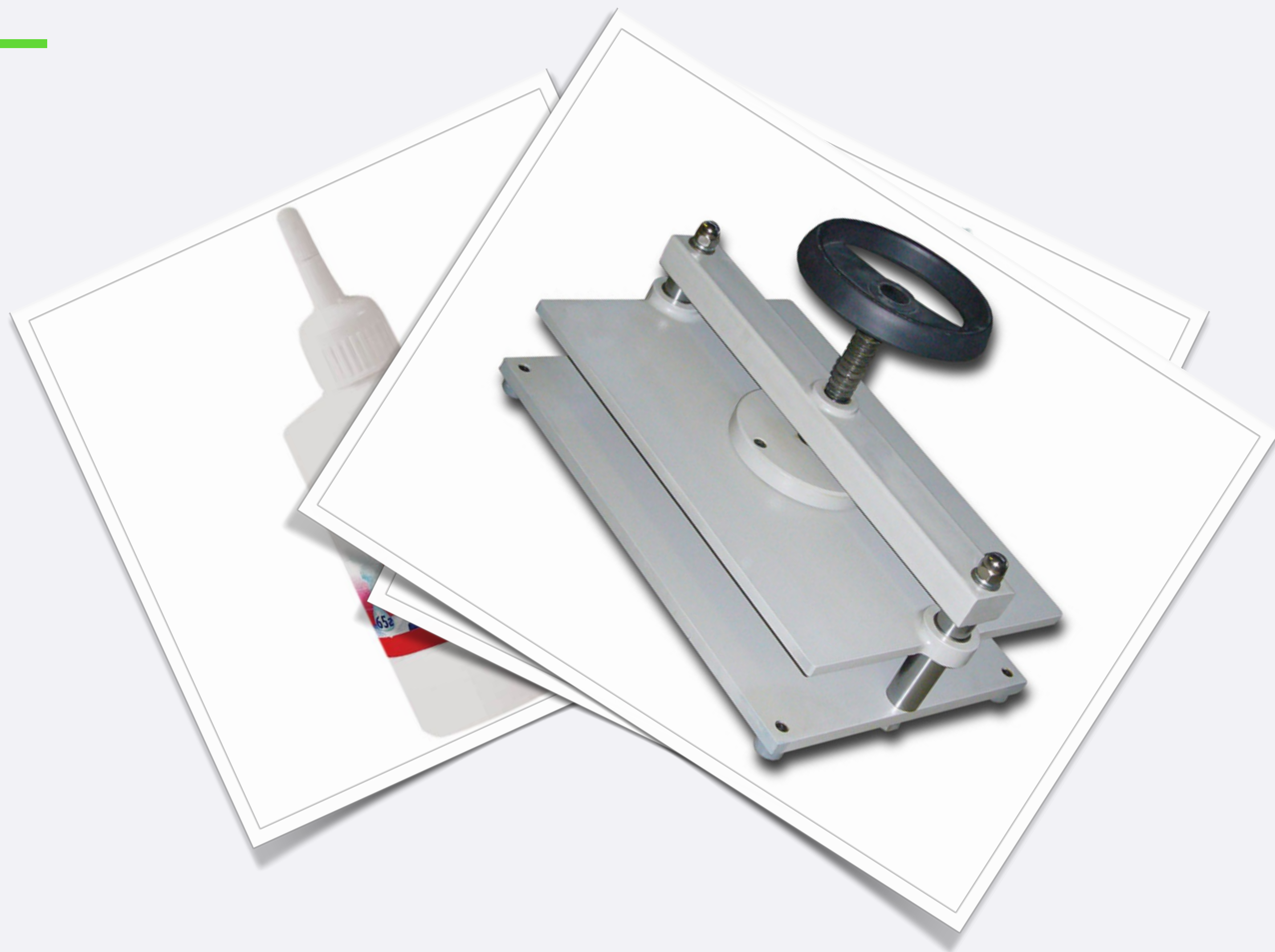
- gulp
- grunt
- webpack
- neutrino



Собираем Javascript

Сборщики

- gulp
- grunt
- webpack
- neutrino



Собираем Javascript

Webpack

```
{  
  "entry": {},  
  "output": {},  
  "module": {  
    "rules": [loaders]  
  },  
  "plugins": [plugins]  
}
```

Собираем Javascript

Webpack - сборка js

- Собрать вместе js
- Реализовать lazy-loading
- Минимизировать
- Использовать только лучший и новейший синтаксис
- Провести любые другие дополнительные издевательства над js

Собираем Javascript

Webpack - сборка css

- Собрать вместе css
- Реализовать модульность
- Минимизировать
- Использовать только лучший и новейший синтаксис
- Провести любые другие дополнительные издевательства над css

Собираем Javascript

Webpack - loaders

- Получаем файл как строку и возвращаем строку
- Декоратор для строк
- Может превратить код `window.navigator` в `window['nav' + 'i' + 'gator']`, если захочет
- Работает здесь и сейчас

Собираем Javascript

Webpack - плагины

- Совокупность обработчиков событий над моделью событий сборки
- Может встроиться и получить доступ к `compiler`, `compile` и любому этапу жизненного цикла сборки
- Может превратить код `window.navigator` в `window['nav' + 'i' + 'gator']`, если захочет, и положить его к соседу
- Работает всегда и везде

Собираем Javascript

Webpack - остальные свойства

- `externals` - вырезает кусок кода, ожидая его получить как-либо из другого места
- `output.publicPath` - относительный путь, по которому сборка будет пытаться себя искать
- `output.libraryTarget` - условный тип файла
- `resolve.modules` - позволяет прописать `import 'src/foo.js'`
- `resolve.alias` - позволяет прописать вместо `import 'src/foo.js'` даже `import 'JAVASCRIPTISGREAT'`
- `devtool` - внедрение в сборку утилит для дебага
- `optimization.minimizer` - минимизатор и прочие оптимизации над сборкой

Собираем Javascript

Babel

- Позволяет поддерживать старые браузеры, меняя синтаксис
- Позволяет пользоваться новыми фичами, которые даже еще не заехали в браузеры
- Позволяет добавить полифиллы
- Позволяет творить дичь с кодом

Собираем Javascript

Babel

- presets - набор плагинов, собираемый на основе дополнительных параметров
- plugins - функции, преобразующие код согласно атомарному сценарию

Собираем Javascript

Unit-тестирование

- Простые и атомарные
- Хорошо сочетаются с TDD
- Быстрые в выполнении и отладке
- Работает в любых окружениях

Собираем Javascript

Unit-тестирование

- chai + mocha
- karma + jasmine
- jest

Собираем Javascript

Unit-тестирование

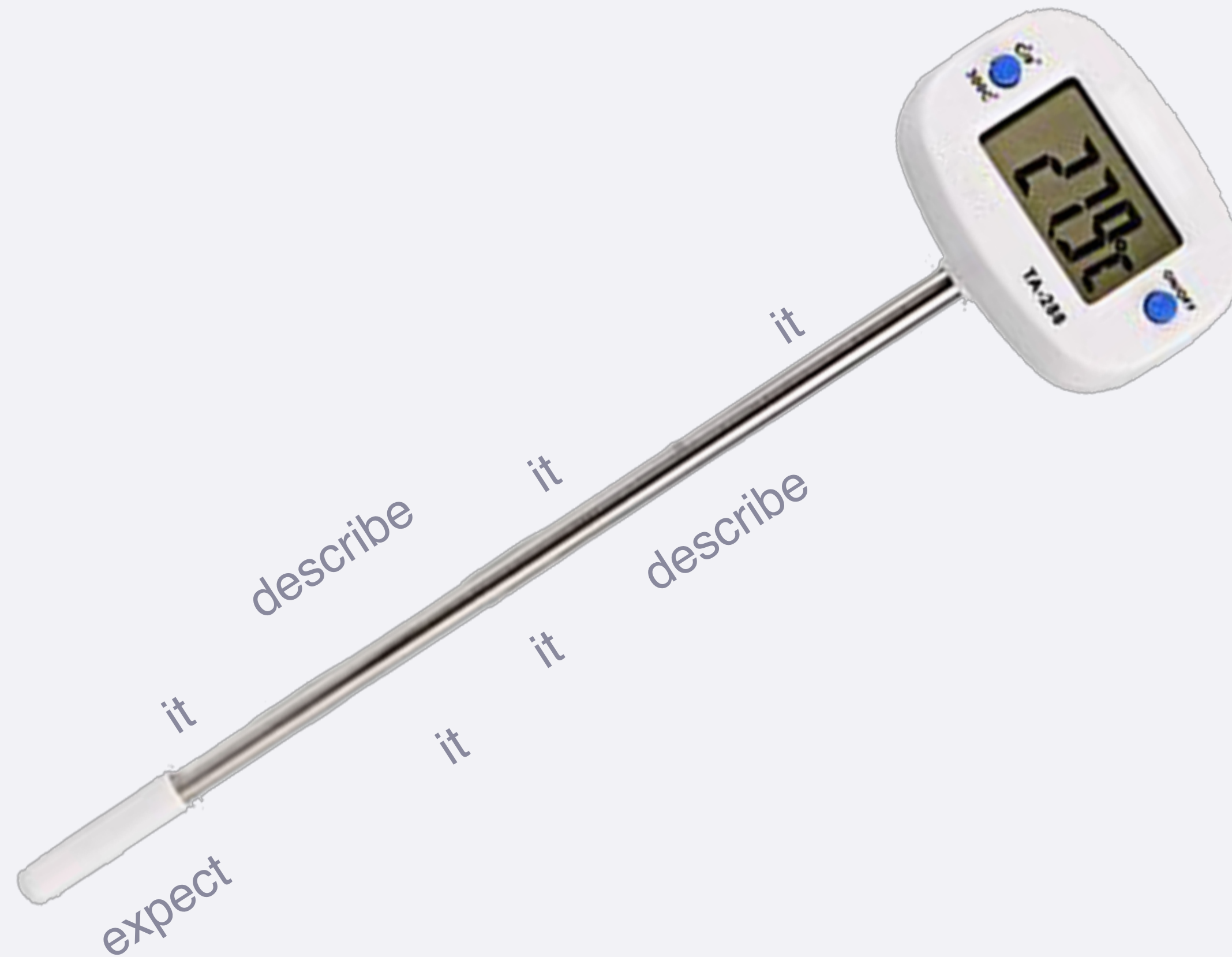
- chai + mocha
- karma + jasmine
- jest

```
describe('Feature', () => {  
  it('Case', () => {  
    expect(true).toBe(true)  
  })  
})
```

Собираем Javascript

Unit-тестирование

- chai + mocha
- karma + jasmine
- jest



Собираем Javascript

Jest

- Taskrunner
- Matching utilities
- Configuration
- Reporters
- Работает со сборкой
- Работает в виртуальном окружении
- Умеет делать snapshot
- Умеет делать изоляцию и заглушки файлов

Собираем Javascript

Статический анализ кода

- JSLint
- ESLint
- Stylelint
- TSLint
- SVGLint
- FSoLint

Собираем Javascript

Статический анализ кода

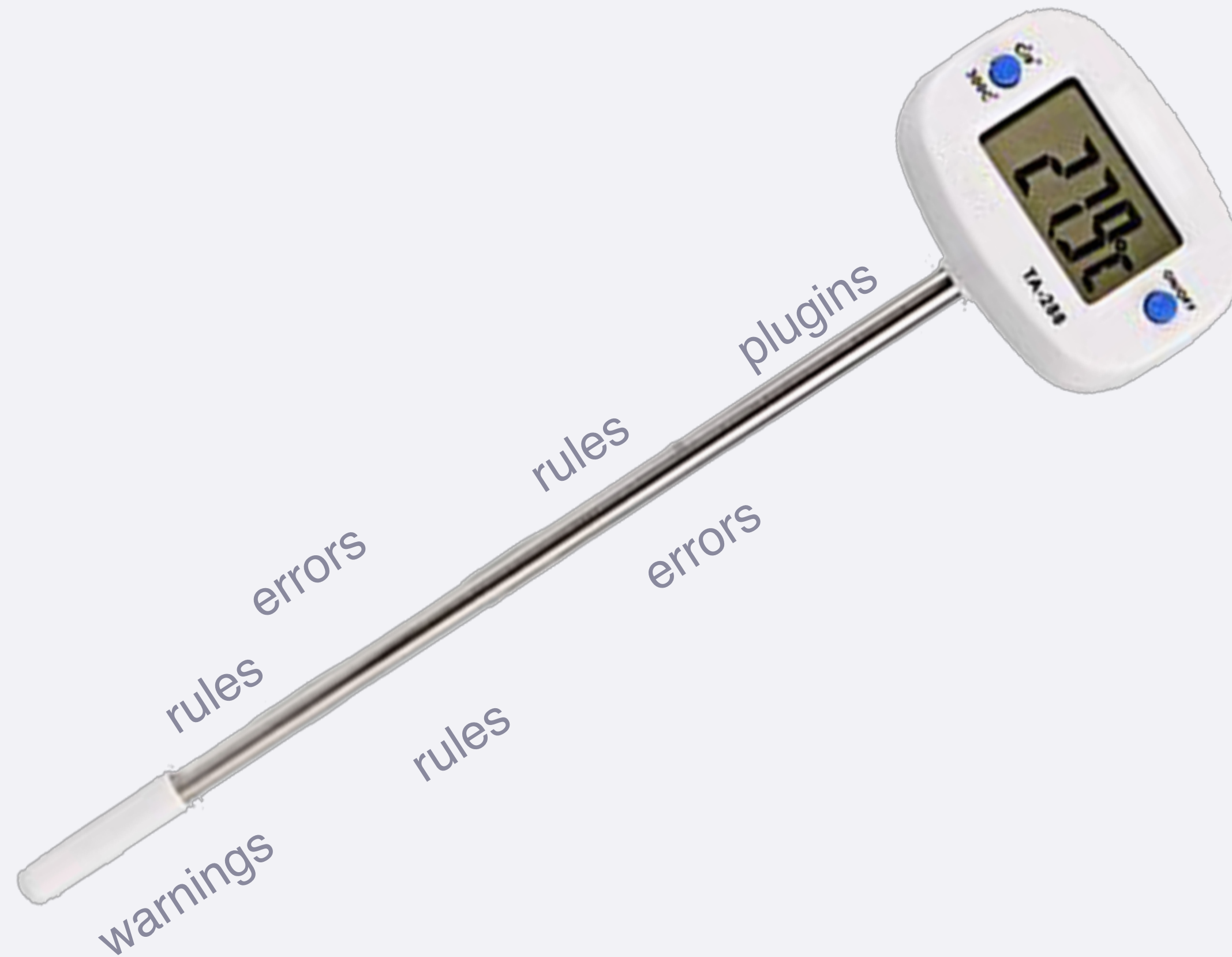
- JSLint
- ESLint
- Stylelint
- TSLint
- SVGLint
- FSoLint



Собираем Javascript

Статический анализ кода

- JSLint
- ESLint
- Stylelint
- TSLint
- SVGLint
- FSoLint



Собираем Javascript

Eslint

- CLI
- Engine
- configs
- plugins
- rules

Собираем Javascript

Eslint

- CLI ➡ командная строка для запуска
- Engine ➡ ядро линтера, можно использовать при программном взаимодействии
- configs ➡ конфигурации, что и как линтим
- plugins ➡ дополнительные правила, расширяем стандартный линтинг
- rules ➡ основная концептуальная единица линтера

Собираем Javascript

node.js

Все перечисленное ранее в презентации выполняется исключительно на node.js. Можно сказать, что node.js - это браузер для сервера. Он выполняется на средствах машины, имеет версионирование и дает свой набор методов и утилит.

У него есть множество пакетов:

- `require('fs')`
- `require('path')`

Собираем Javascript

express.js - сущности экспресса

- App
- Router
- Route
- Middlewares
- static

Собираем Javascript

express.js - подключение роутов

- use
- get
- post

Собираем Javascript

express.js - аргументы роутов

- req
- res
- next

Собираем Javascript

Домашнее задание

- Завести проект и реализовать в нем взаимодействие с сервером из предыдущего урока, парсинг, геттеры и сборку. Протестировать некоторые из них и собрать bundle, reports
- Подробное ТЗ ждите к выходным
- Срок реализации - неделя с момента получения ТЗ
- Необязательно, но если интересно: подключить stylelint, git hooks, расширить перечень скриптов под разные сценарии, реализовать заглушку с возвратом html в качестве сервиса (без перекладки в target), реализовать заглушку с разными тестовыми сценариями в зависимости от process.env.SCENARIO

Собираем Javascript

Полезные ссылки

- <https://www.npmjs.com/>
- <https://webpack.js.org/>
- <https://jestjs.io/>
- <https://babeljs.io/>
- <https://eslint.org/>
- <https://post-css.ru/>
- <https://expressjs.com/ru/>