

2. Setup

Introduction to React

Index



Dependencies



Bundling



Code style



Dev Tools

Index



Dependencies



Bundling



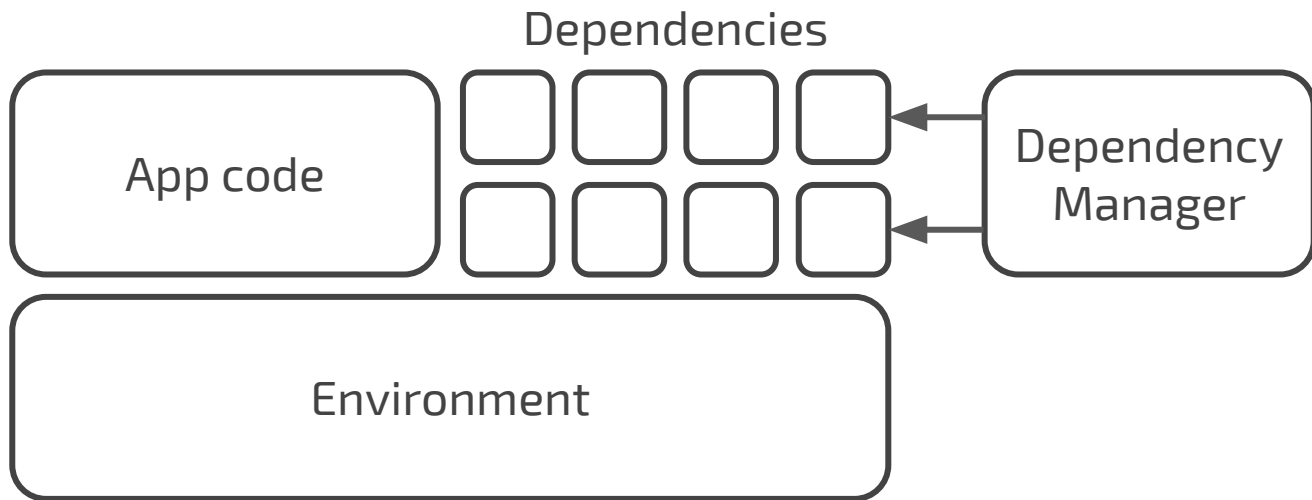
Code style



Dev Tools

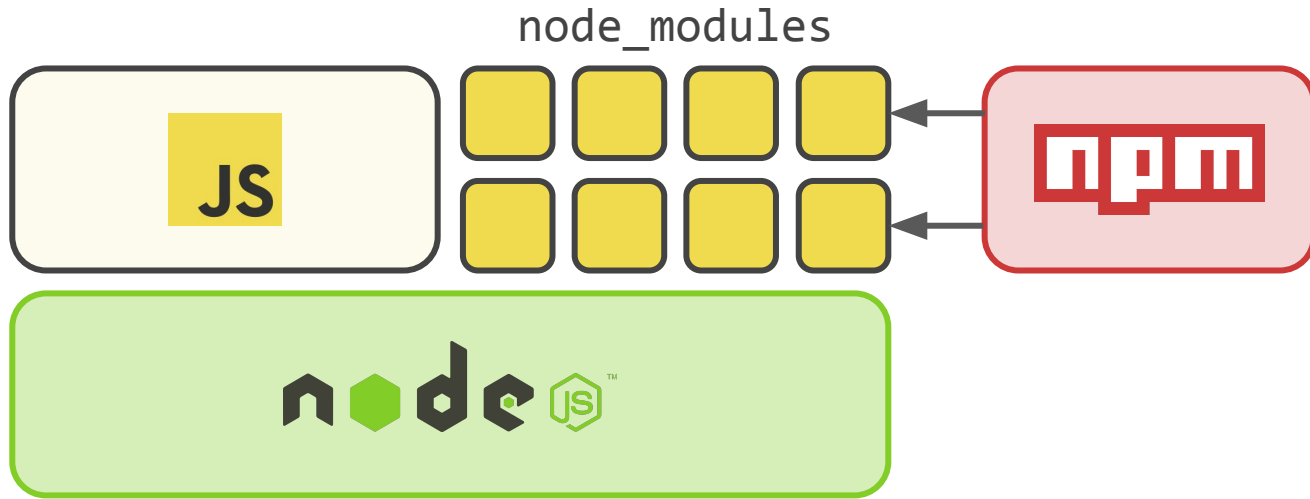


Dependencies



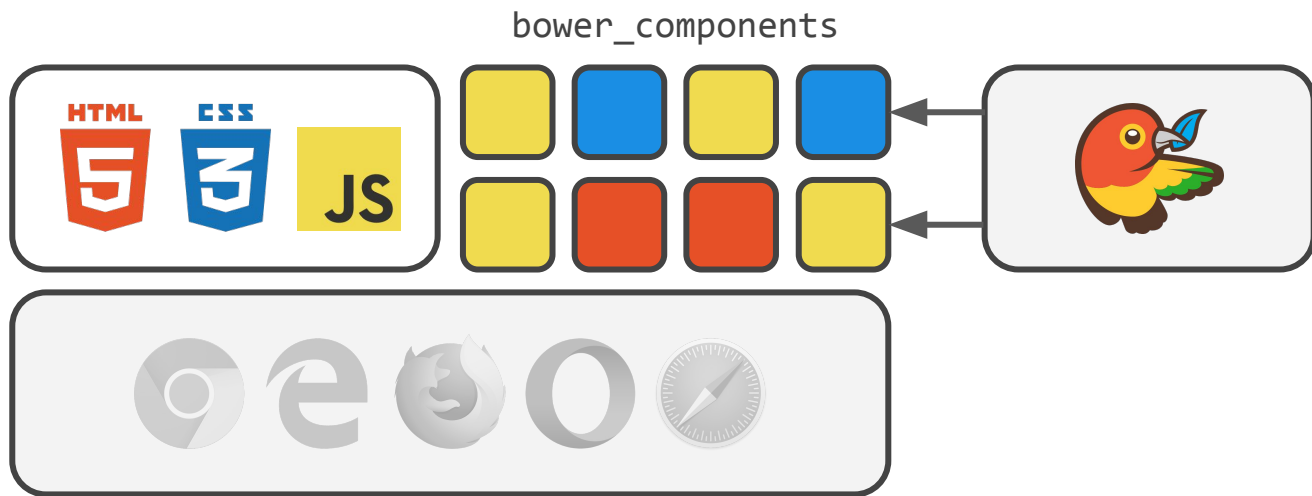


Dependencies - NodeJS



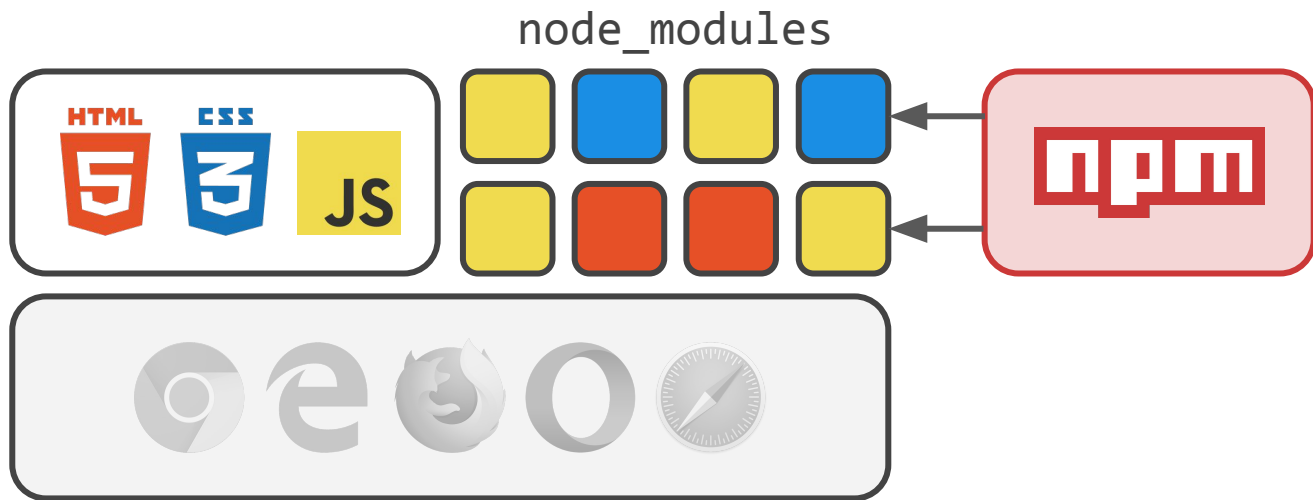


Dependencies - Browser classic approach





Dependencies - Browser unified approach





Dependencies – Create project

1. Install NodeJS

```
https://nodejs.org
```

2. Check NPM version

```
npm -v
```

3. Create and go to a new folder

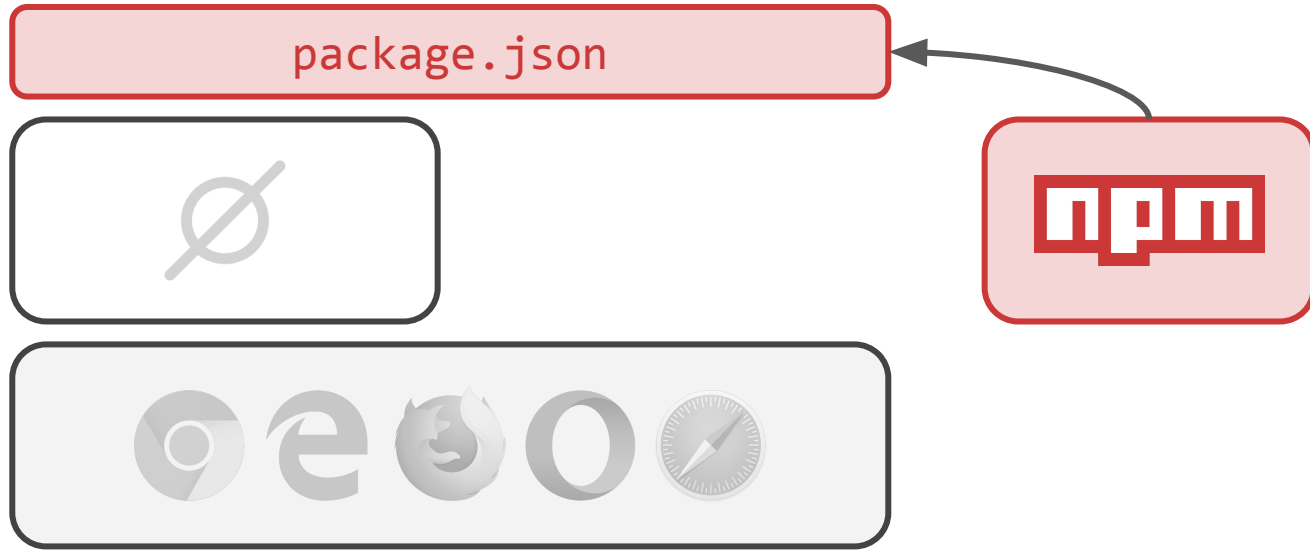
```
mkdir wishlist && cd wishlist
```

4. Start a project in the folder

```
npm init
```



Dependencies





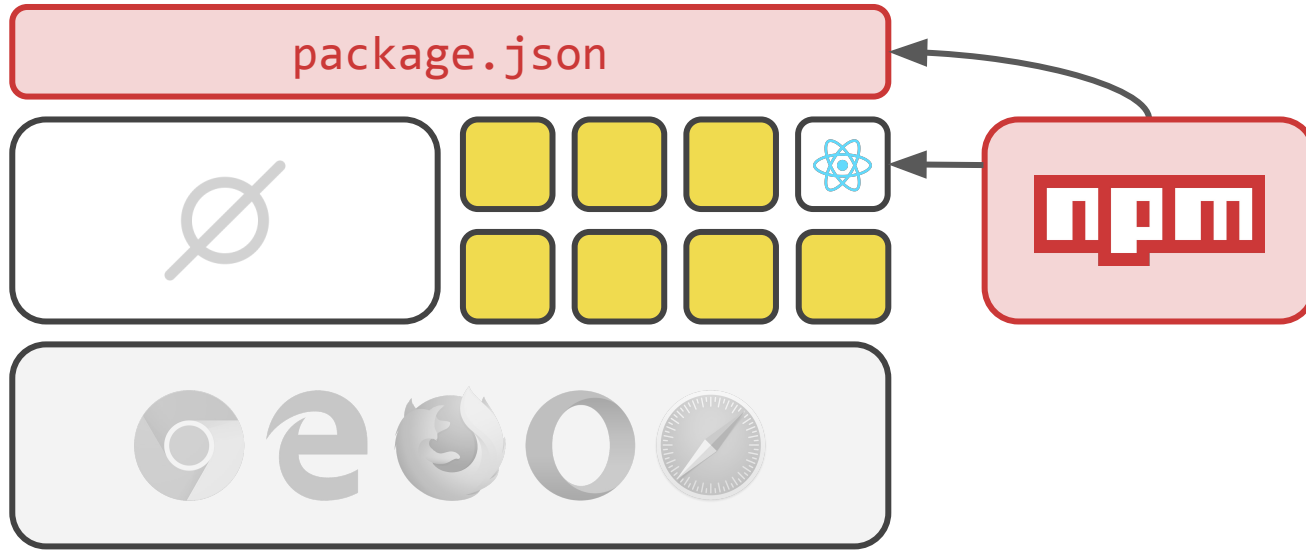
Dependencies - Create project

Install React in the project

```
npm install --save react
```

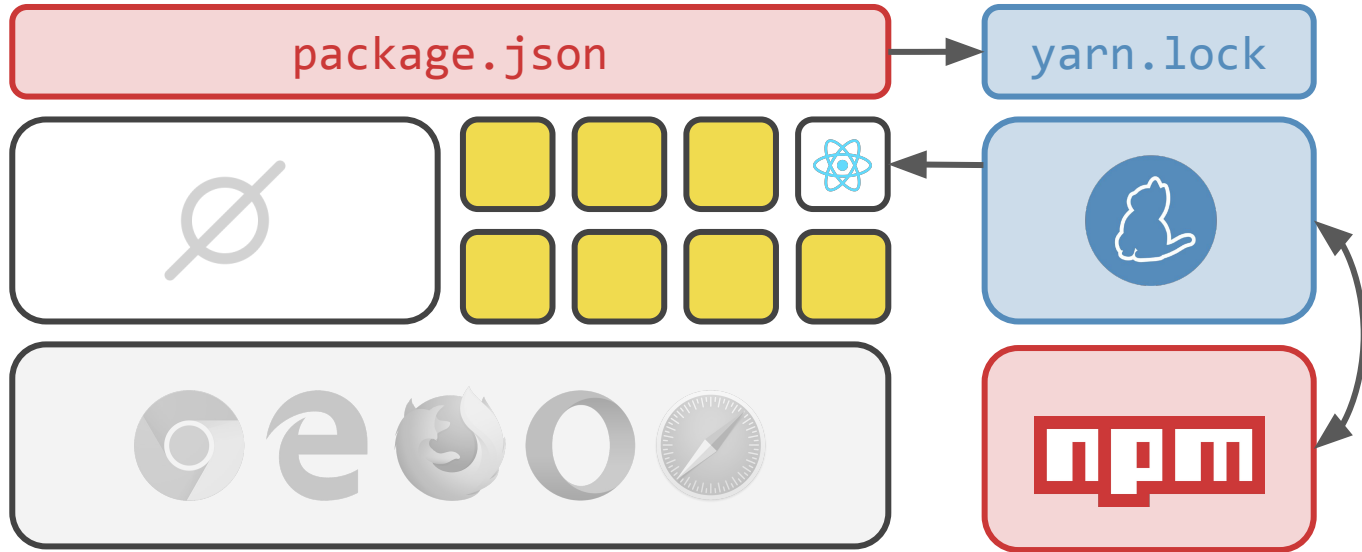


Dependencies





Dependencies - Yarn



Index



Dependencies



Bundling



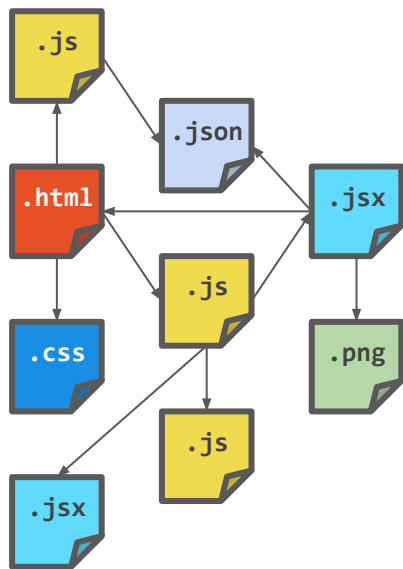
Code style



Dev Tools



Bundling



Developer friendly



Builder



Browser friendly



Bundling

1. Install Parcel in the project

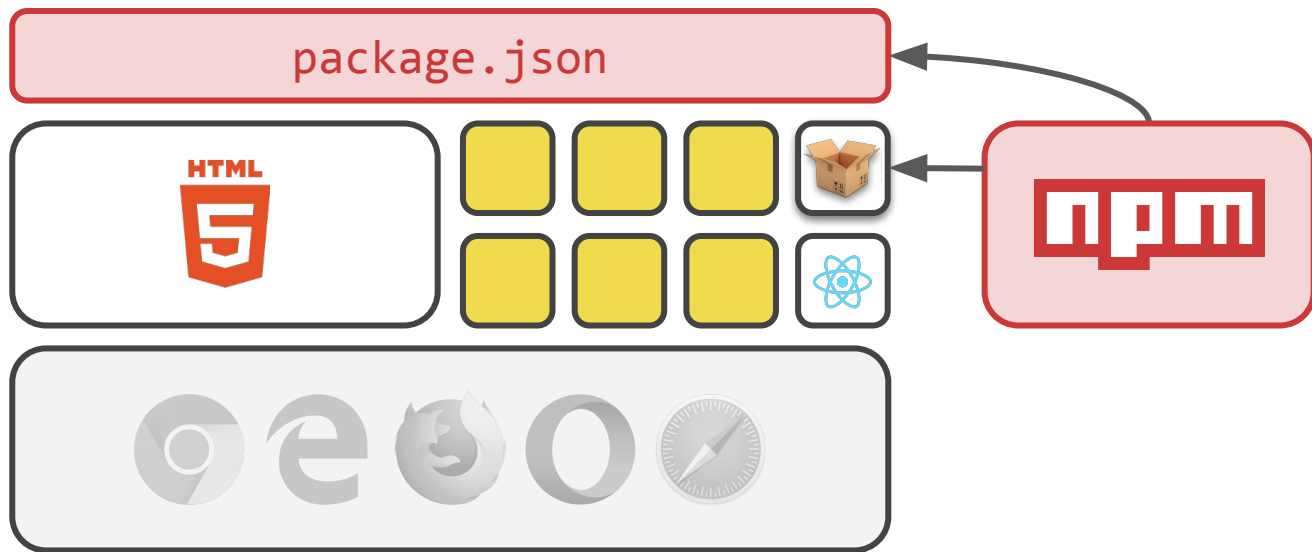
```
npm install --save-dev parcel-bundler
```

2. Create a simple index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Wishlist</title>
  </head>
  <body>My wishlist</body>
</html>
```




Bundling





Bundling

1. Define a start script in `package.json`

```
{ ...  
  "scripts": {  
    "start": "parcel index.html"  
  },  
},
```

2. Run the start script

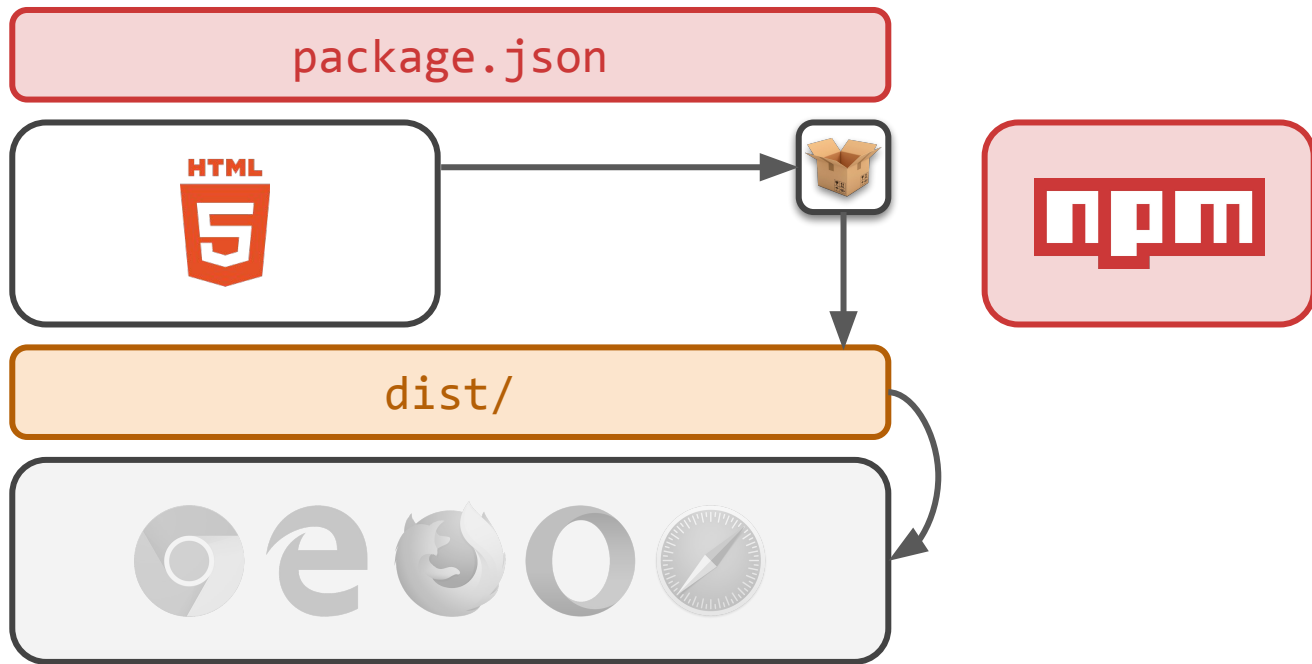
```
npm start
```

3. Open browser

```
http://localhost:1234
```



Bundling





Bundling

Install React DOM and start trying React

```
npm install --save react-dom
```

index.html

```
<html lang="en">
  ...
  <body>
    <div id="root"></div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.jsx

```
import React from "react";
import ReactDOM from "react-dom";

ReactDOM.render(
  <div> Mi wishlist </div>,
  document.getElementById("root")
);
```

Index



Dependencies



Bundling



Code style



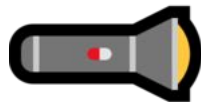
Dev Tools



Code style



Folders Structure



Linting



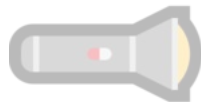
Format



Code style



Folders Structure



Linting



Format



Code style – Folders structure

project-folder/

src/

dist/

package.json

.all-config-files



Code style – Folders structure

project-folder/

src/

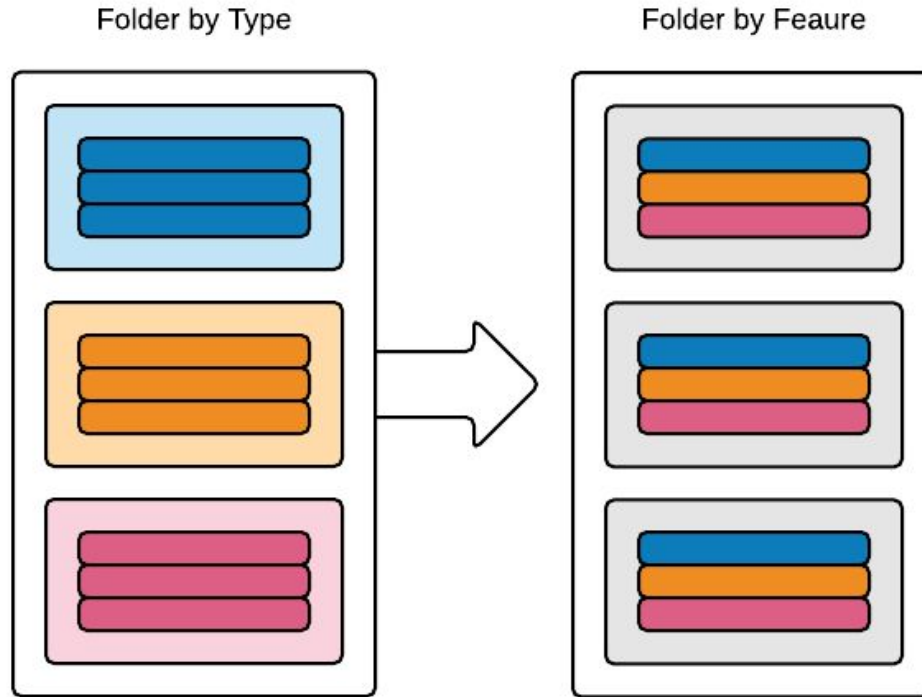
dist/

package.json

.all-config-files



Code style - Folders structure





Code style – Folders structure

Folder-by-Type

```
src/  
├── index.jsx  
├── App.jsx  
├── components/  
│   ├── TaskItem.jsx  
│   ├── TaskForm.jsx  
│   ├── DeleteTask.jsx  
│   ├── ProfileForm.jsx  
│   ├── DeleteProfile.jsx  
│   ├── FriendItem.jsx  
│   └── Avatar.jsx  
├── pages/  
│   ├── TasksPage.jsx  
│   └── ProfilePage.jsx  
├── containers/  
│   ├── TaskListContainer.jsx  
│   ├── NewTaskContainer.jsx  
│   ├── FriendsContainer.jsx  
│   └── ProfileContainer.jsx  
└── services/  
    ├── users.js  
    └── tasks.js
```



Code style - Folders structure

Folder-by-Type

```
src/  
├── index.jsx  
├── App.jsx  
├── components/  
│   ├── TaskItem.jsx  
│   ├── TaskForm.jsx  
│   ├── DeleteTask.jsx  
│   ├── ProfileForm.jsx  
│   ├── DeleteProfile.jsx  
│   ├── FriendItem.jsx  
│   └── Avatar.jsx  
├── pages/  
│   ├── TasksPage.jsx  
│   └── ProfilePage.jsx  
├── containers/  
│   ├── TaskListContainer.jsx  
│   ├── NewTaskContainer.jsx  
│   ├── FriendsContainer.jsx  
│   └── ProfileContainer.jsx  
└── services/  
    ├── users.js  
    └── tasks.js
```

Folder-by-Feature

```
src/  
├── index.jsx  
├── App/  
│   ├── index.jsx  
│   ├── ProfilePage/  
│   │   ├── index.jsx  
│   │   ├── FriendsContainer.jsx  
│   │   ├── ProfileContainer.jsx  
│   │   ├── ProfileForm.jsx  
│   │   ├── DeleteProfile.jsx  
│   │   └── FriendItem.jsx  
│   └── TasksPage/  
│       ├── index.jsx  
│       └── TaskListContainer.jsx  
│           ├── NewTaskContainer.jsx  
│           ├── TaskItem.jsx  
│           ├── TaskForm.jsx  
│           └── DeleteTask.jsx  
├── components/  
│   └── Avatar.jsx  
└── services/  
    ├── users.js  
    └── tasks.js
```



Code style - Folders structure

Folder-by-Type

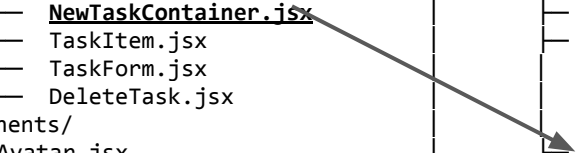
```
src/
├── index.jsx
├── App.jsx
├── components/
│   ├── TaskItem.jsx
│   ├── TaskForm.jsx
│   ├── DeleteTask.jsx
│   ├── ProfileForm.jsx
│   ├── DeleteProfile.jsx
│   ├── FriendItem.jsx
│   └── Avatar.jsx
├── pages/
│   ├── TasksPage.jsx
│   └── ProfilePage.jsx
├── containers/
│   ├── TaskListContainer.jsx
│   ├── NewTaskContainer.jsx
│   ├── FriendsContainer.jsx
│   └── ProfileContainer.jsx
└── services/
    ├── users.js
    └── tasks.js
```

Folder-by-Feature

```
src/
├── index.jsx
├── App/
│   ├── index.jsx
│   ├── ProfilePage/
│   │   ├── index.jsx
│   │   ├── FriendsContainer.jsx
│   │   ├── ProfileContainer.jsx
│   │   ├── ProfileForm.jsx
│   │   ├── DeleteProfile.jsx
│   │   └── FriendItem.jsx
│   └── TasksPage/
│       ├── index.jsx
│       └── TaskListContainer.jsx
│           ├── NewTaskContainer.jsx
│           ├── TaskItem.jsx
│           ├── TaskForm.jsx
│           └── DeleteTask.jsx
├── components/
│   └── Avatar.jsx
├── services/
│   ├── users.js
│   └── tasks.js
```

Extreme Fold.-by-Feat.

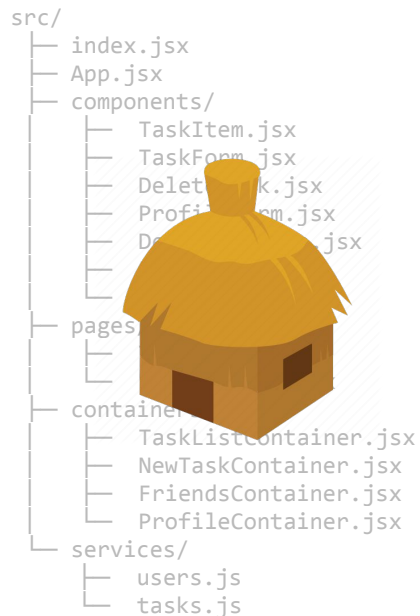
```
src/
├── index.jsx
├── App/
│   ├── index.jsx
│   ├── ProfilePage/
│   │   ├── index.jsx
│   │   ├── FriendsContainer/
│   │   │   ├── index.jsx
│   │   │   └── FriendItem.jsx
│   │   └── ProfileContainer/
│   │       ├── index.jsx
│   │       └── ProfileForm.jsx
│   └── DeleteProfile.jsx
│       └── TasksPage/
│           ├── index.jsx
│           ├── TaskListContainer/
│           │   ├── index.jsx
│           │   ├── DeleteTask.jsx
│           │   ├── TaskItem.jsx
│           │   └── NewTaskContainer/
│           │       ├── index.jsx
│           │       └── TaskForm.jsx
├── components/...
└── services/...
```



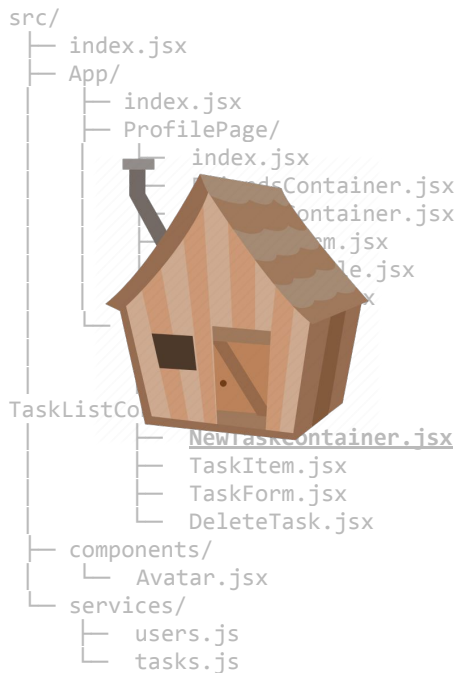


Code style – Folders structure

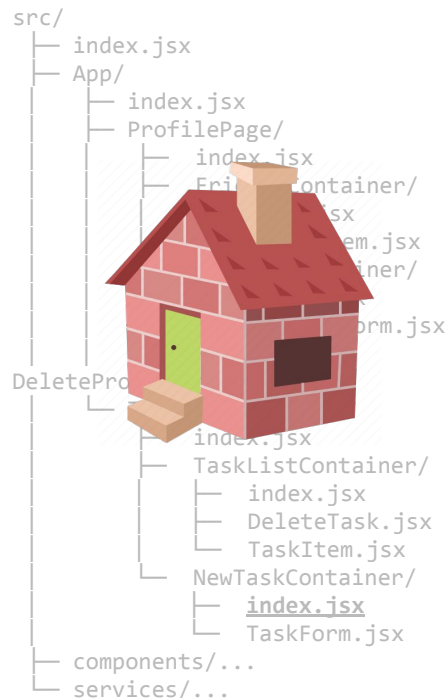
Folder-by-Type



Folder-by-Feature



Extreme Fold.-by-Feat.



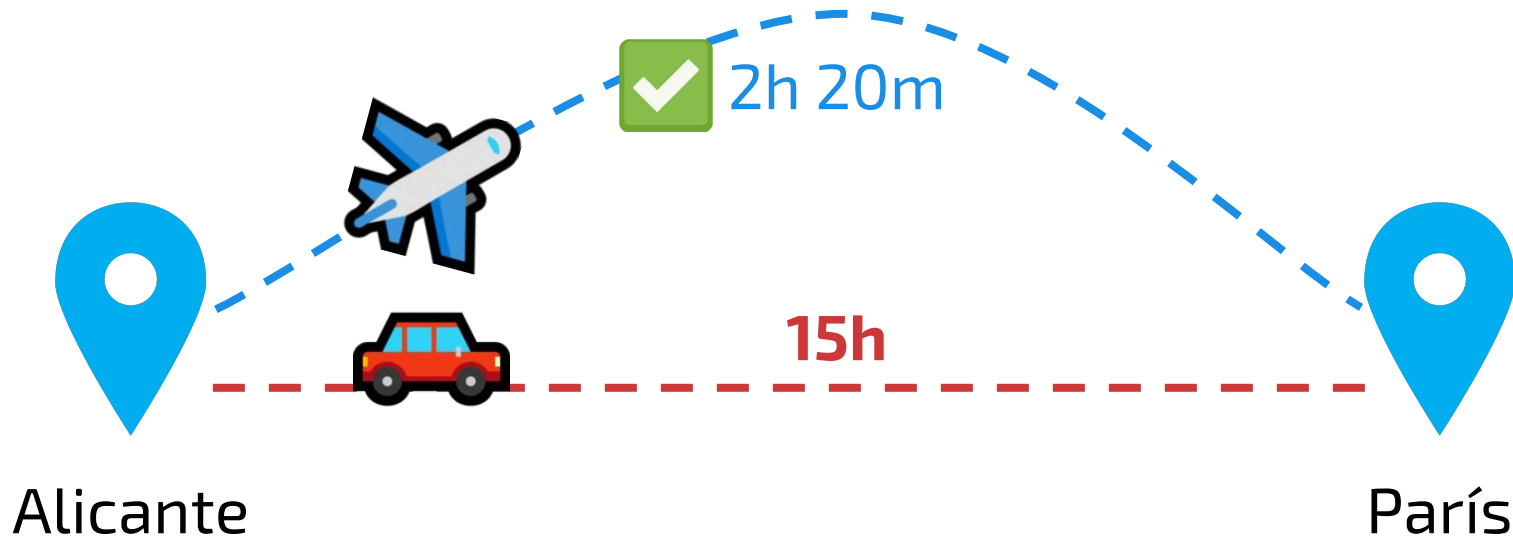


Code style – Folders structure



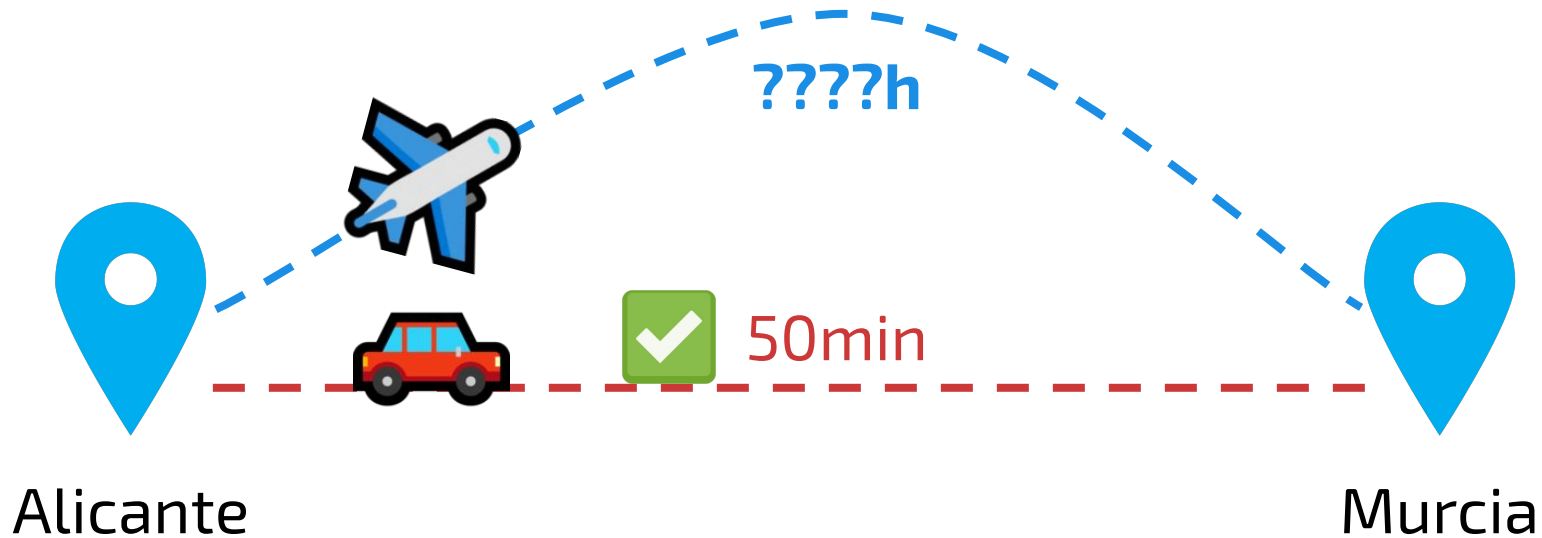


Code style - Folders structure





Code style - Folders structure

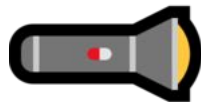




Code style



Folders Structure



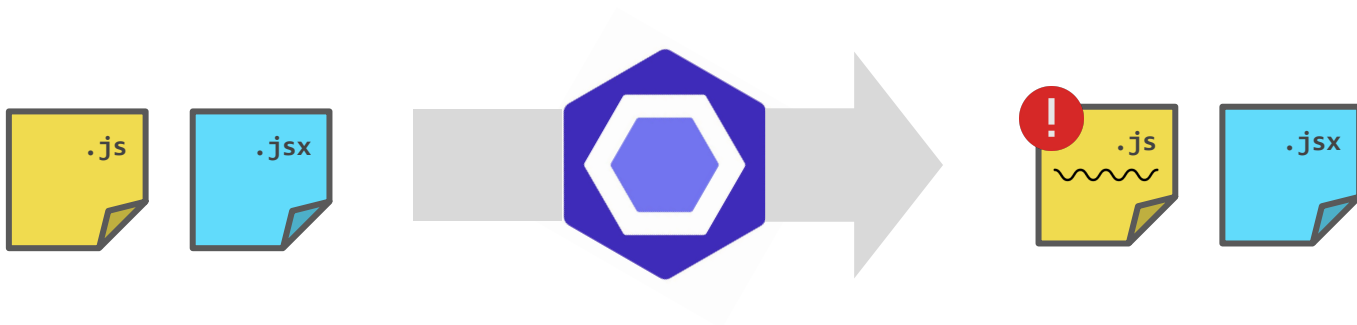
Linting



Format



Code style - Linting





Code style – Linting

1. Install ESLint globally

```
npm install -g eslint
```

2. Init ESLint in the project

```
eslint --init
```

3. Add a lint script in package.json

```
{  
  "scripts": {  
    ...  
    "lint": "eslint --fix src/**/*.{js,jsx}"  
  },  
}
```



Code style

4. Adjust `.eslint.json` with this optional recommendation

Can be placed as a `eslintConfig` field in the `package.json`

```
{  
  "extends": "airbnb",  
  "env": {  
    "browser": true  
  },  
  "rules": {  
    "linebreak-style": "off"  
  }  
}
```

5. Run linter

```
npm run lint
```



Code style - Linting stages



Editor
(e.g. VSCode)



Commit
(e.g. Husky)



Pipeline
(e.g. Jenkins)



Code style - Linting

1. Install lint-staged and husky

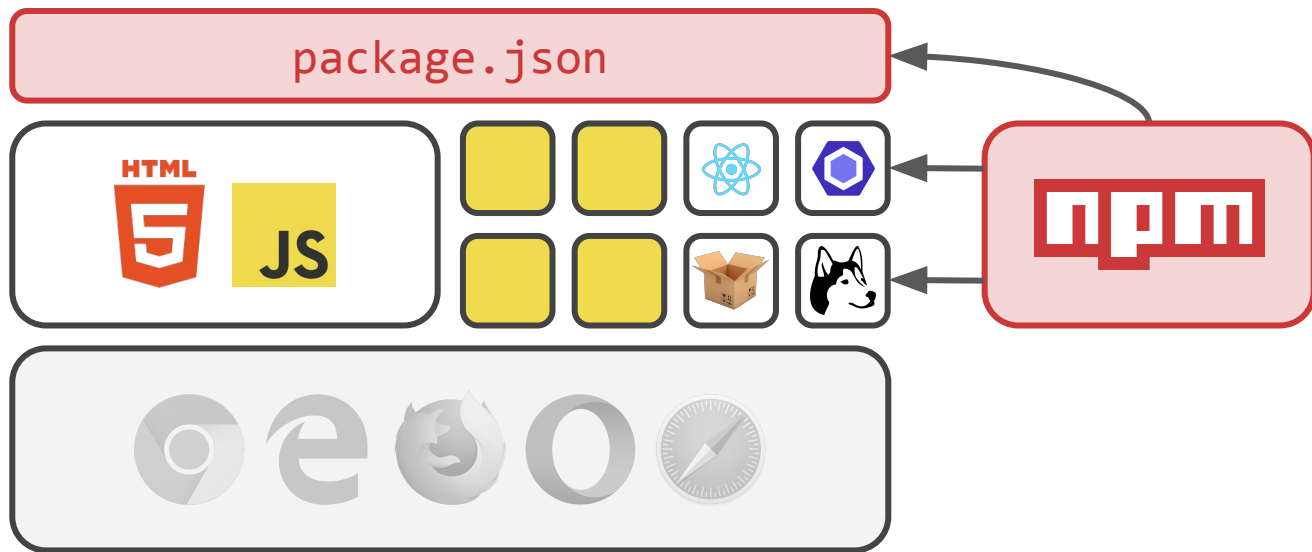
```
npm install --dev lint-staged husky
```

2. Configure the pre-commit hook in package.json

```
{
  ...
  "husky": {
    "hooks": {
      "pre-commit": "lint-staged"
    }
  },
  "lint-staged": {
    "src/**/*.js,jsx": [ "npm run lint" ]
  }
}
```



Bundling

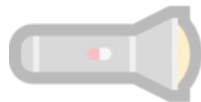




Code style



Folders Structure



Linting



Format



Code style - Linting





Code style – Format

1. Install Prettier

```
npm install --dev prettier
```

2. Add the script and the preferred rules in package.json

```
{
  ...
  "scripts": {
    ...
    "format": "prettier --write \"*.js,jsx,json,css\""
  },
  "prettier": {
    "trailingComma": "all",
    "singleQuote": true
  },
}
```



Code style – Format

3. Add the prettier script to lint staged fired by husky

```
{  
  ...  
  "lint-staged": {  
    "src/**/*.{js,jsx}": [  
      "npm run lint"  
    ],  
    "src/**/*.{js,jsx,json,css,scss}": [  
      "prettier --write",  
      "git add"  
    ]  
  }  
}
```

Index

 Dependencies

 Bundling

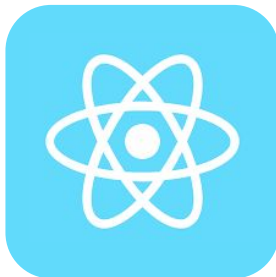
 Code style

 **Dev Tools**

Dev Tools



Extensions





Dev Tools



React Developer Tools

Offered by: Facebook

★★★★★ 1,095

[Developer Tools](#)

👤 1,489,422 users



Dev Tools



Index

 Dependencies

 Bundling

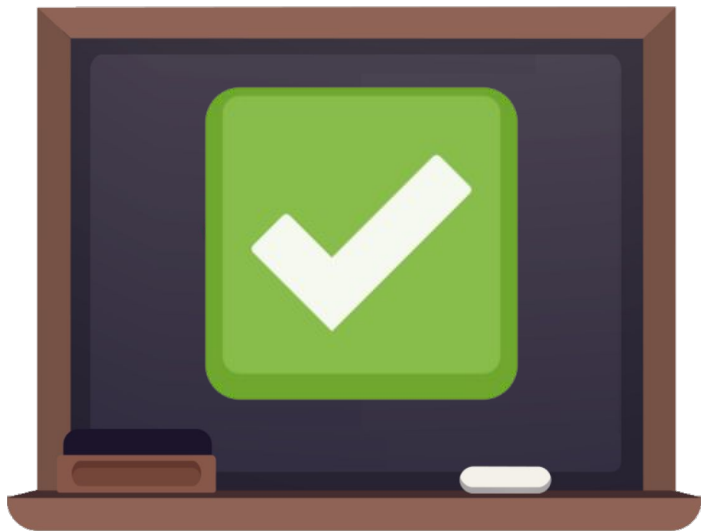
 Code style

 Dev Tools



Exercises!

1. Configure a project with all the necessary tools seen.
 - a. NPM package
 - b. Parcel bundling
 - c. ESLint
 - d. Prettier
 - e. Husky
2. Implement a “Hello World” React app from the project configured. Follow one of the folder structure proposed.



2. Setup

Introduction to React