# parcel-examples (0.0.1)

Maksim Kostromin

Version 0.0.1, 2020-03-17 01:08:07 UTC

## **Table of Contents**

1. Introduction	2
2. Implementation	3
2.1. basic example	3
2.2. babel stage-0	5
2.3. react	5
2.4. preact	6
2.5. vue	7
2.6. yaml	9
2.7. markdown	9
2.8. markdowns (2)	11
2.9. angularjs	12
2.10. react + bootstrap (4)	15
2.11. typescript	16
2.12. rx.js + typescript	17
2.13. bootstrap.scss	17
2.14. react-hooks	18
2.15. preact-hooks	18
3 Links	20

Travis CI status: [Build Status]

## Chapter 1. Introduction

- plain javascript
- typescript
- es next using babel
- rx
- vue
- react
- preact
- angularjs
- yaml
- markdown
- bootstrap
- react-hooks
- preact-hooks
- and many more!

Read reference documentation

other repos with old examples:

- GitHub: daggerok/parcel-vue-example
- GitHub: daggerok/parcel-react-example

generated by generator-jvm yeoman generator (java)

## Chapter 2. Implementation

### 2.1. basic example

prepare project

```
mkdir -p basic

echo "node_modules" >> basic/.gitignore
echo "dist" >> basic/.gitignore
echo ".cache" >> basic/.gitignore
echo '{"scripts":{"start":"parcel src/","build":"parcel build src/index.html"}}' >
basic/package.json

cd basic/
npm init -y
npm i -D parcel-bundler
```

#### prepare project files

```
mkdir -p src
touch src/index.html
touch src/styles.css
touch src/main.js
```

#### main.js

```
(function main() {
  'use strict';
  document.addEventListener('DOMContentLoaded', function onDOMContentLoaded() {
    document.querySelector('#app').innerHTML = '<h1>Hey! </h1>';
  }, false);
})();
```

```
html,
body {
  height: 100%;
}
body {
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Helvetica, Arial,
  sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol';
}
#app {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
}
h1 {
  font-weight: 300;
}
```

#### index.html

```
<!doctype html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport"
        content="width=device-width, initial-scale=1.0">
 <meta http-equiv="X-UA-Compatible" content="ie=edge">
 <title>Basic | Parcel</title>
 <link rel="shortcut icon" href="./favicon.ico" type="image/x-icon">
 <link rel="stylesheet" href="./styles.css">
</head>
<body>
 <div id="app"></div>
 <script src="./main.js"></script>
</body>
</html>
```

#### start development

```
npm start
```

#### open localhost:1234/

#### build bundle

```
npm run build
```

verify result

```
npm i -g serve
serve dist/
```

open localhost:5000/

### 2.2. babel stage-0

parcel needed only .babelrc file and installed required presets

add .babelrc file, install and configure presets: env and stage-0

```
cp -Rf basic babel-stage-0
cd babel-stage-0/
echo '{"presets":["env","stage-0"]}' > .babelrc
npm i -D babel-preset-env babel-preset-stage-0
```

now you can use some cool JS features:

```
const obj = { ololo: 'trololo' };

document.querySelector('#app').textContent = JSON.stringify({
    ...obj,
    hey: 'ho!',
});
```

done.

#### 2.3. react

for React you need even less: just install react library and parcel will recognize everything for you!

```
cp -Rf babel-stage-0 react
cd react/
npm i -S react react-dom
npm i -D babel-preset-react
echo '{"presets":["env","react"]}' > .babelrc
```

add react component:

```
import React, { Component } from 'react';
export class EchoEhlo extends Component {
 constructor() {
    super();
   this.state = {
     message: 'Hey!',
   };
   this.toggle = this.toggle.bind(this);
 toggle() {
   this.setState({
     message: this.state.message.split('').reverse().join(''),
   })
 }
 render() {
    return <h1 onClick={this.toggle}>
      {this.state.message}
   </h1>
 }
}
```

and update code in src/main.js file:

build, run, verify...

done.

### 2.4. preact

do necessary updates from react example:

```
cp -Rf react preact
cd preact/

npm rm react react-dom babel-preset-react

npm i -S preact preact-compat
npm i -D babel-preset-preact babel-plugin-transform-class-properties

echo '{"presets":["env","preact"],"plugins":["transform-class-properties"]}' >
.babelrc
```

implement preact component:

```
import { h, Component } from 'preact';

export /* don't works without default 0.o */ default class EchoEhlo extends Component {

  state = { message: 'Hey!' };
  toggle = () => {
    const curr = this.state.message;
    const reverced = curr.split('').reverse().join('');
    this.setState({ message: reverced, });
  };
  render({}, { message }, {}) {
    return <h1 onClick={this.toggle}>
      {message}
    </h1>
  }
}
```

update entry point:

done.

#### 2.5. vue

to be able build vue apps, in addition to babel example you only need install vue and two dev packages:

```
cp -Rf babel-stage-0 vue
cd vue/
npm i -S vue
npm i -D babel-preset-vue @vue/component-compiler-utils vue-template-compiler
echo '{"presets":["env","vue"]}' > .babelrc
```

add vue component file: App.vue

```
<template lang="html">
 <div id="app">
    <h1>Hey!
              </h1>
 </div>
</template>
<script>
 export default {
    name: 'app'
 }
</script>
<style lang="css">
 html,
 body {
   height: 100%;
 }
 body {
   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Helvetica, Arial,
   sans-serif, 'Apple Color Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol';
 }
 #app {
   display: flex;
   justify-content: center;
   align-items: center;
   height: 100%;
 }
 h1 {
   font-weight: 300;
 }
</style>
```

main.js

```
import Vue from 'vue'
import App from './components/App.vue'

new Vue({
   el: '#app',
   render: h => h(App)
});
```

### 2.6. yaml

to be able to parse YAML we need js-yaml package:

```
cp -Rf basic yaml
cd yaml/
npm i -ES js-yaml
```

add YAML file:

```
app:
    map:
     key: value
    list:
    - value 1
    - value 2
```

main.js

#### 2.7. markdown

to be able to parse YAML we need parcel-plugin-markdown package:

```
cp -Rf basic markdown
cd markdown/
npm i -ED parcel-plugin-markdown
```

```
# арр
## ololo
**ololo article**
list:
- one
- two
- three
## trololololo
    _0_
    / \\
## js
<!--
title: JavaSCript Article'
metadata:
  ololo: trololo
-->
### JS article
js, ololo, trololo, javascript, olololo ololo, trololo, olololo ololo, trololo,
olololo ololo,
trololo, olololo, js, ololo, trololo, olololo, js ololo, trololo, olololo.....
```javascript
function ololo(arg) {
 arg = arg || 'trololo';
 console.log('ololo', arg);
}
· · ·
```

main.js



See also markdowns and markdowns2 projects for non single md-file parsing use case...

### 2.8. markdowns (2)

example how we can handle lazy loading with parcel for markdown files lazy processing on runtime

add ./src/posts folder with some posts in markdown format:

main.js implementation:

```
import marked from 'marked';
const years = require('./posts/**/*.md');
Object.keys(years).map(year => {
  const months = years[year];
  Object.keys(months).map(month => {
    const days = months[month];
    Object.keys(days).map(day => {
      const posts = days[day];
      Object.keys(posts)
        .map(src => posts[src])
        .forEach(uri => fetch(uri)
          .then(resp => resp.text())
          .then(markdown => marked(markdown))
          .then(html => document.querySelector('#app').innerHTML += html));
    });
  });
});
```

#### 2.9. angularjs

with no comments - I tired....

./src/index.html

Main application bootstrap entrypoint: ./src/main.js

```
import 'angular/angular-csp.css';
import './styles.css';
//
import angular from 'angular';
import applicationModule from './app';

angular.bootstrap(document, [applicationModule.name], {
   strictDi: true, // data-ng-strict-di=""
   cloak: true, // data-ng-cloak=""
});
```

Application module config: ./src/app/index.js

```
/**
 * Application module configuration.
import angular from 'angular';
import uiRouter from 'angular-ui-router';
//
import ComponentsModule from './components';
const applicationModule = angular.module('application.module', [
  uiRouter,
  ComponentsModule.name,
]);
const Config = ($urlRouterProvider, $locationProvider) => {
  $urlRouterProvider.otherwise('/');
  $locationProvider.hashPrefix('!');
};
applicationModule.config(['$urlRouterProvider', '$locationProvider', Config]);
export default applicationModule;
```

#### Components module config: ./src/app/components/index.js

```
/**
  * Components module configuration.
  */

import angular from 'angular';
//
import AppComponentModule from './app';

const componentsModule = angular
  .module('components.module', [
    AppComponentModule.name,
  ]);

export default componentsModule;
```

#### App module config: ./src/app/components/app/index.js

```
/**
 * App component module configuration.
 */
import angular from 'angular';
import uiRouter from 'angular-ui-router';
//
import Config from './config.js';
import Component from './component.js';

const appComponentModule = angular
   .module('app.component.module', [uiRouter])
   .component('app', Component)
   .config(['$stateProvider', Config]);

export default appComponentModule;
```

#### App component config: ./src/app/components/app/config.js

```
export default ($stateProvider) => $stateProvider.state({
  url: '/',
  name: 'app',
  template: `<app></app>`,
});
```

App component controller: ./src/app/components/app/controller.js

```
export default class Controller {
  constructor() {
    this.$ctrl = this;
}

$onInit() {
    this.greeting = this.first = 'hi';
    this.second = 'yay!';
}

toggleGreeting() {
    this.greeting = this.greeting === this.first
    ? this.second
    : this.first;
}
```

### 2.10. react + bootstrap (4)

prepare

```
cp -Rf react react-bootstrap
cd react-bootstrap/
npm i -S bootstrap
npm i -ED babel-preset-env babel-preset-react babel-preset-stage-0
```

```
{
    "presets": [
        ["env", {
             "browsers": ["last 2 versions", "safari >= 7"]
        }
    }],
    "react",
    "stage-0"
]
}
```

add bootstrap import and add some bootstrap class for styling:

```
import 'bootstrap/dist/css/bootstrap.css';
import React, { Component } from 'react';
export class EchoEhlo extends Component {
 constructor() {
    super();
    this.state = {
      message: 'Hey!',
   };
    this.toggle = this.toggle.bind(this);
 }
 toggle() {
   this.setState({
      message: this.state.message.split('').reverse().join(''),
   })
 }
 render() {
    return <h1 onClick={this.toggle} className='alert-heading'>
      {this.state.message}
   </h1>
 }
}
```

done.

### 2.11. typescript

you don't needed anything special. parcel compiling typescript working out of the box

done.

### 2.12. rx.js + typescript

```
cp -Rf ts rxts
cd rxts/
npm i -E rxjs
```

file main.ts:

done.

### 2.13. bootstrap.scss

```
cp -Rf react-bootstrap sass-bootstrap
cd sass-bootstrap/
npm i
echo '$primary: #c63 !default;' > src/_variables.scss
mv src/styles.css src/styles.scss
# similary fix index.html: styles.css -> styles.scss
```

remove import from file src/components/echo-ehlo.js:

```
// import 'bootstrap/dist/css/bootstrap.css';
```

update file styles.scss:

```
@import "./variables.scss";
@import "../node_modules/bootstrap/scss/bootstrap.scss";
```

done.

#### 2.14. react-hooks

edit file app.js:

```
import 'babel-polyfill';
import { render } from 'react-dom';
import React, { useState } from 'react';

function App() {
  const [message, setMessage] = useState('Hello');
  const reversed = str => str.split('').reverse().join('');
  const toggleReverse = () => setMessage(reversed(message));

  return <h1 onClick={toggleReverse}>
      { message }
      </h1>
}

render(
  <App/>,
      document.querySelector('#app')
);
```

done.

### 2.15. preact-hooks

edit file app.js:

```
import { h, render } from 'preact';
import { useState } from 'preact/hooks';
import React from 'preact/compat';

function App() {
   const [message, setMessage] = useState('Hello');
   const reversed = str => str.split('').reverse().join('');
   const toggleReverse = () => setMessage(reversed(message));

   return <h1 onClick={toggleReverse}>
      { message }
      </h1>
}

render(
   <App/>,
      document.querySelector('#app')
);
```

done.

## Chapter 3. Links

- parcel
- parcel examples
- react
- preact
- vue
- GitHub: parcel-bundler/awesome-parcel
- GitHub repo
- GitHub pages