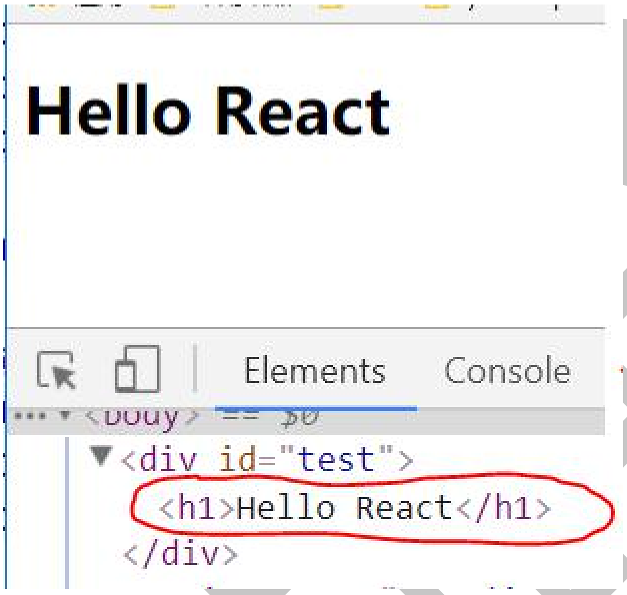
1.1. React

1.2. React

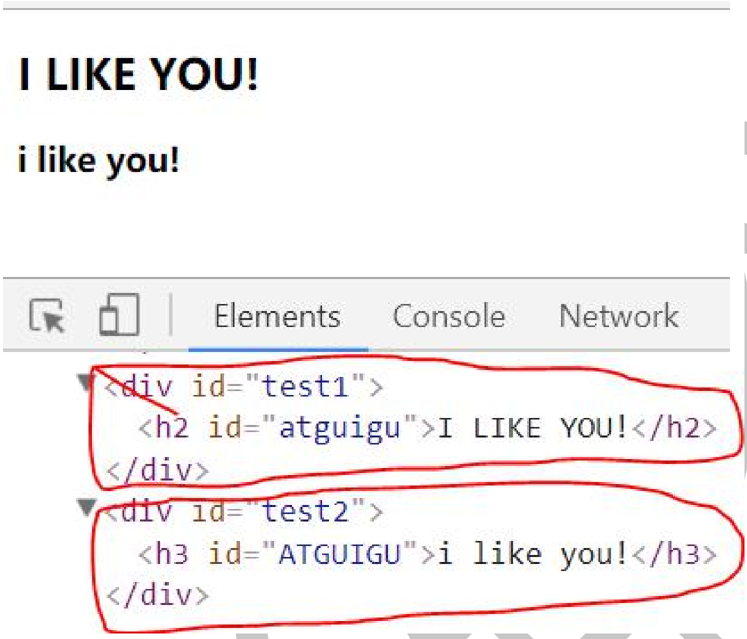


Hello React

```
••• ▼ <body> == $0
  ▼<div id="test">
     <h1>Hello React</h1>
   </div>
```

```
< c                   a a c       c         ac  d                ></ c   >
< c                   a a c       c         ac  d   d                ></ c   >
< c                   a a c       c       bab            ></ c   >
< c                     bab   > //        babel
  //         DOM
  c     vDom = <   >Hello React</   > //
  //         DOM           DOM
  R ac DOM.render(vDom, d c     .getElementById(      ))
</ c    >
```

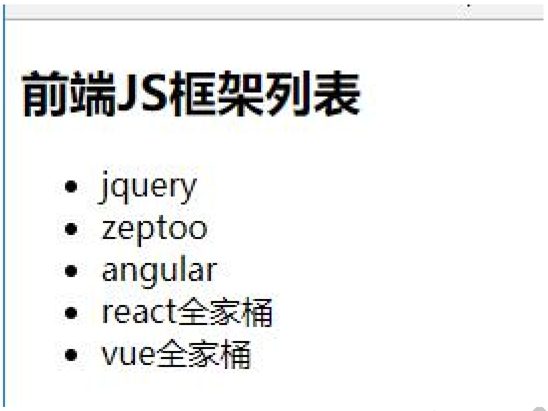
1.3. React JSX



I LIKE YOU!

i like you!

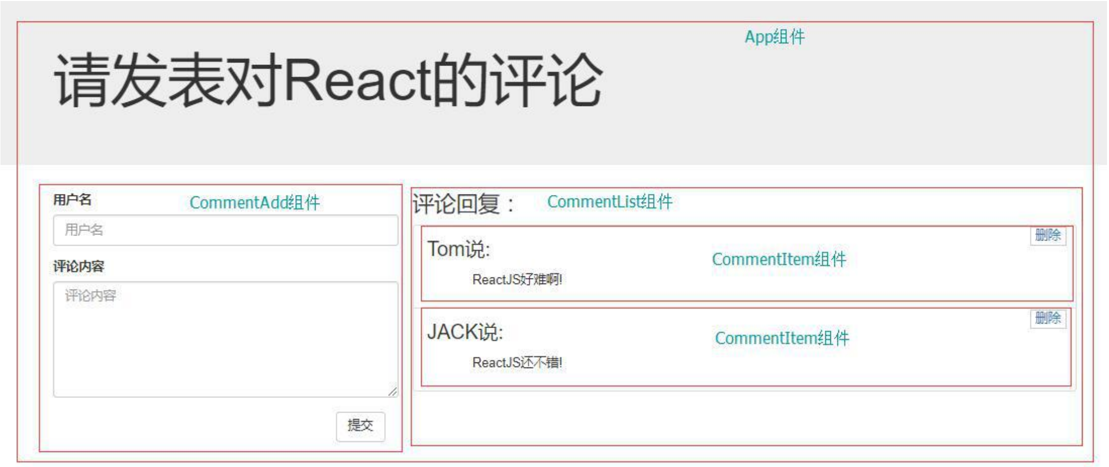
```
<div id="test1">
  <h2 id="atguigu">I LIKE YOU!</h2>
</div>
<div id="test2">
  <h3 id="ATGUIGU">i like you!</h3>
</div>
```



前端JS框架列表

- jquery
- zeptoo
- angular
- react全家桶
- vue全家桶

1.4.



App组件

请发表对React的评论

用户名 CommentAdd组件

用户名

评论内容

评论内容

提交

评论回复： CommentList组件

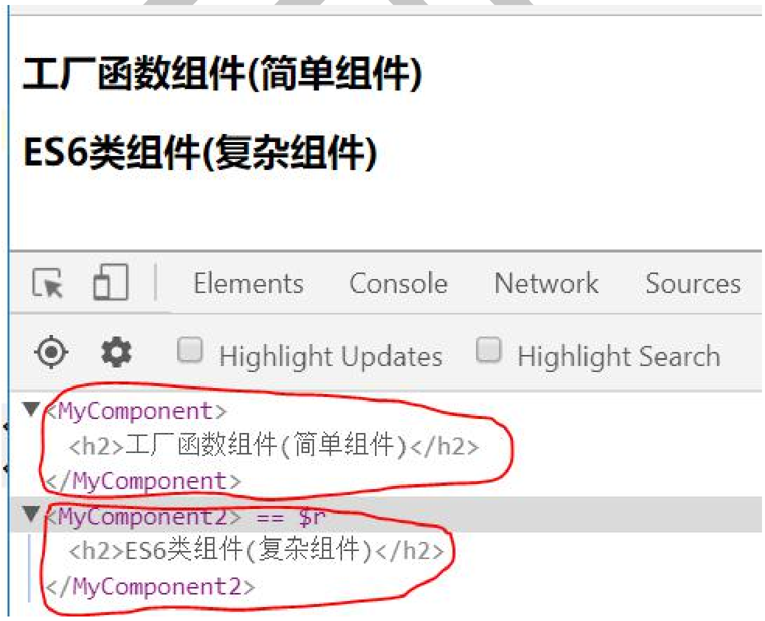
Tom说: CommentItem组件 删除

ReactJS好难啊!

JACK说: CommentItem组件 删除

ReactJS还不错!

2.1.



工厂函数组件(简单组件)

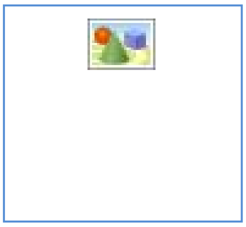
ES6类组件(复杂组件)

```
▼<MyComponent>
   <h2>工厂函数组件(简单组件)</h2>
 </MyComponent>
▼<MyComponent2> == $r
   <h2>ES6类组件(复杂组件)</h2>
 </MyComponent2>
```

```
   c      MyComponent () {
          <  >          (       )</  >
}


c a   MyComponent2      d  R ac .Component {
  render () {
    c       .log(    ) // MyComponent2
          <  >ES6      (        )</  >
  }
}
```

```
R ac DOM.render(<M C          />, d c      .getElementById(   a      ))
```

2.2. 1: state



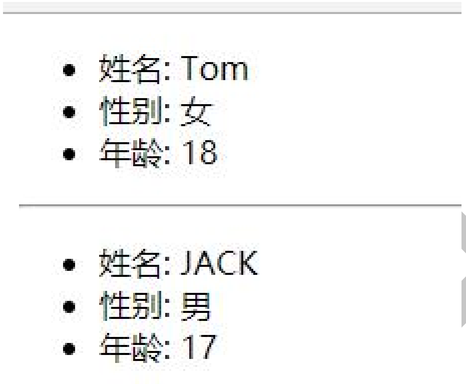
2.3. 2: props

:

1).

2). ,

3). , 18



- 姓名: Tom
- 性别: 女
- 年龄: 18

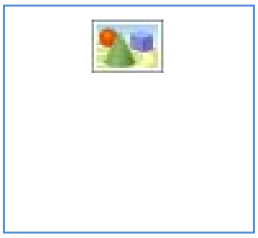
- 姓名: JACK
- 性别: 男
- 年龄: 17

2.4. 3: refs

: , :

2. ,

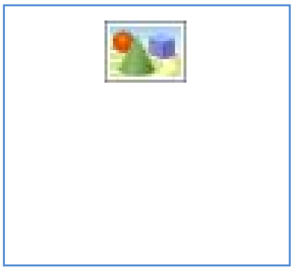
3. 2 ,



2.5.

:

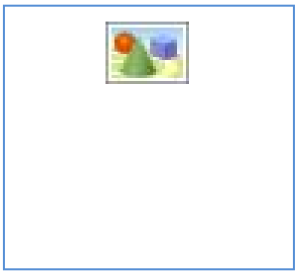
1. todo
2. , ,



2.6.

:

1. ,
3.



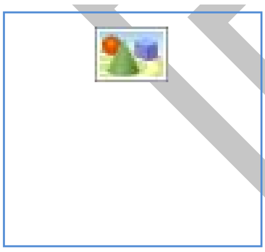
2.7.

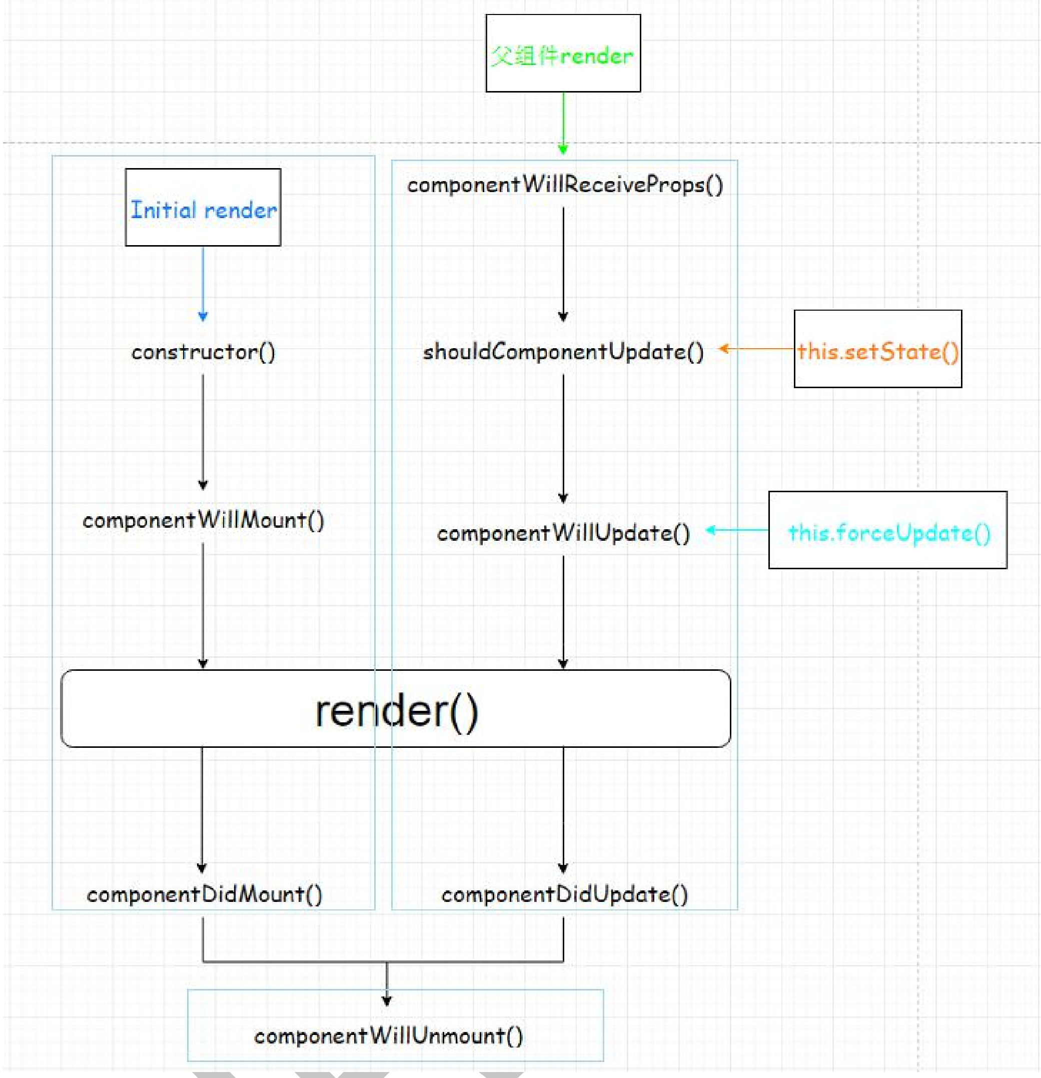
:

1. /

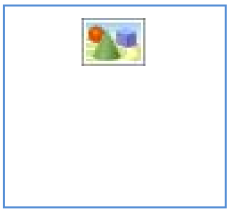
2. 2S

3.

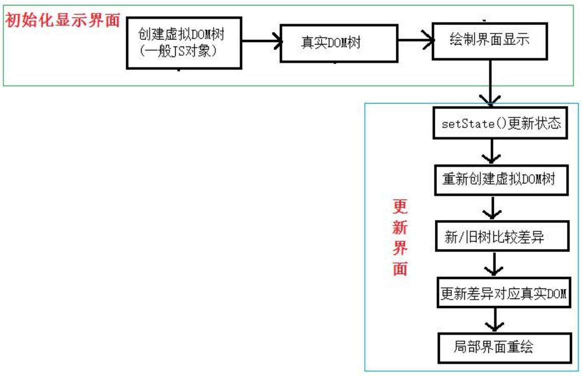


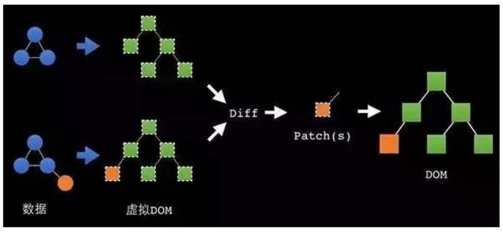

父组件render
Initial render
constructor()
componentWillMount()
componentWillReceiveProps()
shouldComponentUpdate()
this.setState()
componentWillUpdate()
this.forceUpdate()
render()
componentDidMount()
componentDidUpdate()
componentWillUnmount()

2.8.   DOM   DOM Diff



```
c a   HelloWorld      d  R ac .Component {
  c       c    (props) {
          (props)
         .  a    = {
        da   :      Date()
    }
  }

  componentDidMount () {
    setInterval(() => {
          .setState({
        da   :      Date()
      })
    }, 1000)
  }

  render () {
    c       .log(    d      )
          (
      < >
        Hello, <                    ac     d     Y      a         />!  b
        It is {    .  a   .da   .toTimeString()}
      </ >
    )
  }
}

R ac DOM.render(
  <H    W    d/>,
  d c      .getElementById(    a      )
)
```



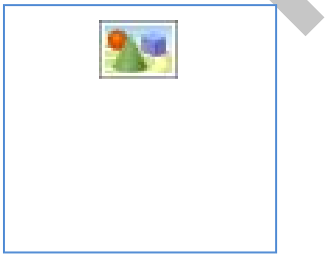


3.1. create-react-app react

```
npm install -g create-react-app
create-react-app hello-react
cd hello-react
npm start
```

```
ReactNews
   |--node_modules---
   |--public
       |-- index.html-----------------
   |--scripts
       |-- build.js-------------------build
       |-- start.js-------------------start
   |--src------------
       |--components-----------------react
       |--index.js-------------------        js
   |--.gitignore------git
   |--package.json----
   |--README.md-------             readme
```

3.2. demo:



4.1.

：

1.

2. github

3. ，

4.2. axios

4.3. Fetch

4.4. demo: github users

5.1.

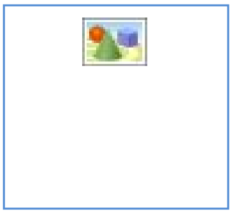
5.2.

6.1.

```
<!DOCTYPE      >
<       a       >
<  ad>
  <   a c a        UTF    >
  <      >history test</      >
</  ad>
<b d >
  < ><                    ></ >
  <a                   c  c         push(        ) >test1</a><b ><b >
  <b         C  c  push(        ) >push test2</b      ><b ><b >
  <b         C  c  back() >    </b      ><b ><b >
  <b         C  c  forword() >    </b      ><b ><b >
  <b         C  c  replace(        ) >replace test3</b      ><b ><b >

  < c                a a c
 c          cd  b   c   c                       ></ c     >
  < c                a a c     >
        history = H      .createBrowserHistory() //
    // history = History.createHashHistory() //
    // console.log(history)

       c     push (to) {
      history.push(to)
              a
    }

       c     back() {
      history.goBack()
    }

       c     forword() {
      history.goForward()
    }

       c     replace (to) {
      history.replace(to)
    }

    history.listen((location) => {
      c      .log(                  , location)
    })
  </ c     >
</b d >
</    >
```

6.2. react-router   API

6.3.



```
        React          ac
        d  a       c     About() {
          <d  >About        </d  >
}
```

```
        React          ac
        d  a       c     About() {
          <d  >Home        </d  >
}
```

```
        React          ac
        {Na L   }         ac         d

        d  a       c     MyNavLink(props) {
          <Na L     {...props} ac    C a  Na    ac    C a   />
}
```

```
     React         ac
     {R    , S   c , R d   c }         ac         d
     MyNavLink         c              a
     About                 ab
     Home

     d   a   c a   App      d  React.Component {

render () {
         (
    <d  >

      <d   c a  Na        >
        <d   c a  Na    c               c         >
          <d   c a  Na     a     ad    >
            <  >React Router Demo</  >
          </d  >
        </d  >
      </d  >

      <d   c a  Na        >
        <d   c a  Na    c        c                >
          <d   c a  Na               >
            {/*           */}
            <M Na L    c a  Na                          ab     >About</M Na L    >
            <M Na L    c a  Na                               >Home</M Na L    >
          </d  >
        </d  >
        <d   c a  Na    c        >
          <d   c a  Na     a    >
            <d   c a  Na     a    b d  >
              {/*               */}
              <S   c >
                <R      a     ab      c        {About} />
                <R      a            c        {Home} />
                <R d   c       ab     />
              </S   c >
            </d  >
            </d  >
          </d  >
        </d  >
      </d  >
    )
  }
}
```

```
.ac     C a    {
  c    :   d         a  ;
}
```

```
       React            ac
       ReactDOM            ac  d
       {B      R      , Ha  R      }          ac          d
       App          c           a


            d   c

ReactDOM.render(
  (
    <B        R      >
      <A    />
    </B        R      >
    /*<HashRouter>
      <App />
    </HashRouter>*/
  ),

  d c      .getElementById(       )
)
```

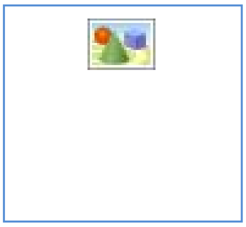
6.4.

```
       React           ac
       d  a    c a    News       d  React.Component {
    a    = {
       A  : [          ,          ,          ]
  }

  render () {
             (
      <d  >
        <  >
          {
                . a  .    A  .map((news, index) => <       {index}>{news}</  >)
          }
        </  >
      </d  >
    )
  }
}
```

```
       React           ac
       {L    , R    }         ac          d
       d  a    c a   Message      d  React.Component {
    a   = {
        a   : []
  }

  componentDidMount () {
    //        ajax
    setTimeout(() => {
      c     data = [
        { d: 1,      :  M   a      },
        { d: 3,      :  M   a      },
        { d: 6,      :  M   a      },
      ]
          .setState({
            a   : data
      })
    }, 1000)
  }


  render () {
    c     path =     .     . a c . a

           (
      <d  >
        <  >
          {
                .  a  .    a   .map((m, index) => {
                     (
                <      {index}>
                  <L            >{m.     }</L   >
                </  >
              )
            })
          }
        </  >
      </d  >
    )
  }
}
```

```
        React              ac
        {S    c , R      , R d    c }          ac          d
        MyNavLink          c               a
        News
        Message                    a

        d  a       c     Home() {
          (
    <d  >
      <  >Home          </  >
      <d  >
        <   c a  Na      a   a   ab  >
          <  >
            <M Na L                          >News</M Na L    >
          </  >
          <  >
            <M Na L                       a    >Message</M Na L    >
          </  >
        </  >
        <S    c >
          <R      a                c          {News} />
          <R      a               a     c          {Message} />
          <R d    c                  />
        </S    c >
      </d  >
    </d  >
  )
}
```

6.5.

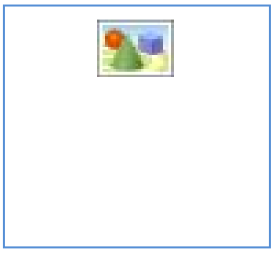


```
       React         ac
c     messageDetails = [
  { d: 1,      :  M   a      , c      :               },
  { d: 3,      :  M   a      , c      :               },
  { d: 6,      :  M   a      , c      :               },
]
       d  a        c     MessageDetail(props) {

  c     id = props. a c .params. d
  c     md = messageDetails.find(md => md. d===id*1)

          (
    <  >
      <  >ID: {md. d}</  >
      <  >TITLE: {md.     }</  >
      <  >CONTENT: {md.c       }</  >
    </  >
  )
}
```

```
       React         ac
       {L    , R     }          ac          d
```

```
     MessageDetail                        a   d  a

     d  a    c a   Message      d  React.Component {
  a   = {
      a   : []
}

componentDidMount () {
  //          ajax
  setTimeout(() => {
    c      data = [
      { d: 1,       :  M   a      },
      { d: 3,       :  M   a      },
      { d: 6,       :  M   a      },
    ]
        .setState({
          a   : data
    })
  }, 1000)
}

render () {
  c      path =      .     . a c . a

         (
    <d  >
      <  >
        {
              .  a  .    a   .map((m, index) => {
                   (
              <        {index}>
                <L        { ${path} ${m. d} }>{m.     }</L   >
              </  >
            )
          })
        }
      </  >
      <  />
      <R       a   { ${path}   d } c         {MessageDetail}></R     >
    </d  >
  )
  }
}
```

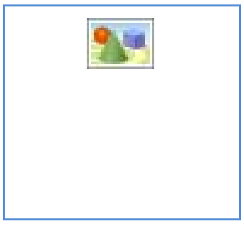
6.6.



```
         React           ac
         {L    , R    }          ac          d
         MessageDetail                         a   d  a

         d  a    c a   Message      d  React.Component {
      a    = {
          a    : []
    }

    componentDidMount () {
      //          ajax
      setTimeout(() => {
        c     data = [
          { d: 1,      :  M   a      },
          { d: 3,      :  M   a      },
          { d: 6,      :  M   a      },
        ]
            .setState({
              a    : data
        })
      }, 1000)
```

```
  }

  ShowDetail = (id) => {
        .     .       .    (           a   ${id} )
  }

  ShowDetail2 = (id) => {
        .     .       .    ac (           a   ${id} )
  }

  back = () => {
        .     .       .  Bac ()
  }

  forward = () => {
        .     .       .  F   a d()
  }

  render () {
    c      path =      .     . a c . a

             (
      <d  >
        <  >
          {
                .  a  .     a   .map((m, index) => {
                     (
                <       {index}>
                  <L        { ${path} ${m. d} }>{m.     }</L   >
                    b
                  <b         C  c  {() =>     .ShowDetail(m. d)}>
(push)</b      >  b
                  <b         C  c  {() =>     .ShowDetail2(m. d)}>
(replace)</b      >
                </  >
              )
            })
          }
        </  >
        < >
          <b        C  c  {    .back}>   </b      >  b
```

7.2. ant-design-mobile



```
       React, {Component}        ac
//
       Button      a  d   b      b b
       Toast      a  d   b      b   a

       d  a    c a   App      d  Component {
  handleClick = () => {
    Toast.info(         , 2)
  }

  render() {
           (
      <d  >
        <B                   a      C  c  {     .handleClick}>    </B        >
      </d  >
    )
  }
}
```

```
        React          ac  ;
        ReactDOM          ac  d
        App          A
//          css
         a  d   b    d    a  d   b    c



ReactDOM.render(<A   />, d c      .getElementById(      ))
```

```
<   a  a              c         d   d   c    d         a   ca        a       ca
         ca             ca ab       />

< c
  c          a  a   a  b  c   c     c         a  c  c          a  c  c     ></ c
 >
< c   >
     ( addE     L          in d c      ) {
   d c      .addEventListener( DOMC      L ad d ,     c    () {
     FastClick.attach(d c     .b d );
   },  a    );
  }
   (!   d  .P       ) {
   d c      .writeln(   c
  c          a  a   a  b  c   c     c
 +   +   +   + c      );
  }
</ c    >
```

●

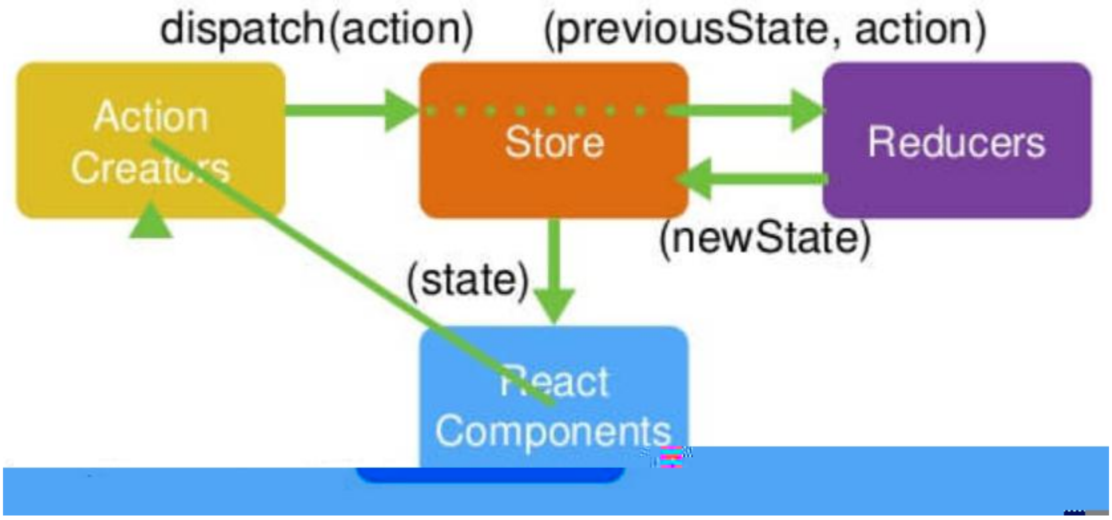
```
  c      : {
     a   :    ac  a          d   a   ,
  b   d :    ac  a          d b   d ,
       :    ac  a          d                d
}
```

●

```
c     {injectBabelPlugin} = require(   ac  a          d );
  d   .exports =    c     override(config, env) {
  config = injectBabelPlugin([         , {  b a  Na  :  a  d   b    ,       :  c   }],
config);
         config;
};
```

```
// import 'antd-mobile/dist/antd-mobile.css'

// import Button from 'antd-mobile/lib/button'
// import Toast from 'antd-mobile/lib/toast'
       {Button, Toast}       a  d   b
```

8.1. redux


dispatch(action)
(previousState, action)
Action Creators
Store
Reducers
(newState)
(state)
React Components

8.2. redux API

8.3. redux

8.4.   redux

```
/*
action       type
 */
       c     INCREMENT =    c
       c     DECREMENT =  d c
```

```
/*
action creator
 */
       {INCREMENT, DECREMENT}         ac

       c     increment = number => ({    : INCREMENT, number})
       c     decrement = number => ({    : DECREMENT, number})
```

```
/*
        state      action,                 state
 */
       {INCREMENT, DECREMENT}          c    a    Ac    T

       {INCREMENT, DECREMENT}         ac

          c     counter(state = 0, action) {
  c      .log( c       , state, action)
      c  (action.    ) {
    ca   INCREMENT:
             state + action.   b
    ca   DECREMENT:
             state - action.   b
    d  a   :
             state
  }
}
```

```
/*

 */
       React, {Component}         ac
       PropTypes
       * a  actions              d   ac

       d  a    c a   App       d  Component {

    a  c propTypes = {
         : PropTypes.object.isRequired,
  }

  increment = () => {
    c     number =      .    .numSelect. a    * 1
        .     .     .dispatch(actions.increment(number))
```

```
}

decrement = () => {
  c     number =     .    .numSelect. a    * 1
      .     .     .dispatch(actions.decrement(number))
}

incrementIfOdd = () => {
  c     number =     .    .numSelect. a    * 1

      count =     .     .     .getState()
     (count % 2 === 1) {
        .     .     .dispatch(actions.increment(number))
  }
}

incrementAsync = () => {
  c     number =     .    .numSelect. a    * 1
  setTimeout(() => {
        .     .     .dispatch(actions.increment(number))
  }, 1000)
}

render() {
         (
    <d  >
      < >
       click  {    .     .     .getState()} times {   }
      </ >
      <    c          S   c  >
        <        a       >1</      >
        <        a       >2</      >
        <        a       >3</      >
      </    c >{   }
      <b        C  c  {    .increment}>+</b     >
      {   }
      <b        C  c  {    .decrement}>-</b     >
      {   }
      <b        C  c  {    .incrementIfOdd}>increment if odd</b     >
      {   }
      <b        C  c  {    .incrementAsync}>increment async</b     >
      </d  >
    )
  }
}
```

```
       React        ac
       ReactDOM        ac   d
       {createStore}         d

       App         c          a
       {counter}          d     d c

//     counter        store
c     store = createStore(counter)

//
c     render = () => {
  ReactDOM.render(
    <A         {store}/>,
    d c     .getElementById(      )
  )
}
//
render()

//     (   )   ,               ,
store.subscribe(render)
```

8.5. react-redux

```
/*
UI     :            redux API
 */
       React          ac
       PropTypes

       d  a    c a   Counter      d  React.Component {

    a  c propTypes = {
    c     : PropTypes.number.isRequired,
      c       : PropTypes.func.isRequired,
    d c       : PropTypes.func.isRequired
  }
```

```
increment = () => {
  c     number =     .    .numSelect. a    * 1
      .     .  c      (number)
}

decrement = () => {
  c     number =     .    .numSelect. a    * 1
      .     .d c      (number)
}

incrementIfOdd = () => {
  c     number =     .    .numSelect. a    * 1
      count =     .     .c
     (count % 2 === 1) {
        .     .  c      (number)
  }
}

incrementAsync = () => {
  c     number =     .    .numSelect. a    * 1
  setTimeout(() => {
        .     .  c      (number)
  }, 1000)
}

render() {
         (
    <d  >
      < >
       click {     .     .c     } times {    }
      </ >
      <    c          S   c  >
        <        a      >1</        >
        <        a      >2</        >
        <        a      >3</        >
      </    c >{   }
      <b        C  c  {    .increment}>+</b     >
      {   }
      <b        C  c  {    .decrement}>-</b     >
      {   }
```

```
        <b          C  c  {    .incrementIfOdd}>increment if odd</b      >
        {   }
        <b          C  c  {    .incrementAsync}>increment async</b      >
      </d  >
    )
  }
}
```

```
/*
    Counter
 */
       React          ac
//
       {connect}          ac    d
//     action
       {increment, decrement}              d   ac

       Counter           c          c

//            App
       d  a     connect(
  state => ({c     : state}),
  {increment, decrement}
)(Counter)
```

```
       React          ac
       ReactDOM          ac  d
       {createStore}          d
       {Provider}          ac    d

       App          c   a       a
       {counter}           d     d c

//     counter         store
c      store = createStore(counter)

//
ReactDOM.render(
  (
    <P    d          {store}>
      <A   />
    </P     d  >
  ),
  d c      .getElementById(      )
```

8.6. redux

```
        {createStore, applyMiddleware}         d
        thunk           d
//      counter          store
c      store = createStore(
  counter,
  applyMiddleware(thunk) //
)
```

```
//      action creator(            )
```

```
        c     incrementAsync = number => {
          dispatch => {
     setTimeout(() => {
       dispatch(increment(number))
     }, 1000)
   }
 }
```

```
incrementAsync = () => {
  c     number =      .     .numSelect. a    *1
       .      .  c       A    c(number)
}
```

```
        {increment, decrement, incrementAsync}             d    ac
//            App
        d  a     connect(
  state => ({c     : state}),
  {increment, decrement, incrementAsync}
)(Counter)
```

8.7. redux

```
        { c       W   D  T      }         d   d

c     store = createStore(
  counter,
  c      W   D  T    (applyMiddleware(thunk))
)
```

8.8. ：