

## API جزء (API Component) در ReactJS

در این بخش می خواهیم به بررسی API جزء در ری اکت بپردازیم. برای این منظور به سه متد `setState()`, `forceUpdate` و `ReactDOM.findDOMNode()` می پردازیم. در کلاس های جدید ES6 باید این api را به صورت دستی مقید کنیم. در این مثال از `this.method.bind(this)` استفاده خواهیم کرد.

### Set State

متد `setState()` متدی است که در بروز رسانی حالت جزء کاربرد دارد. این متد جای حالت را عوض نمی کند بلکه صرفا تغییراتی را بر حالت اصلی اعمال می کند.

```
import React from 'react';

class App extends React.Component {

  constructor() {

    super();

    this.state = {

      data: []

    }

    this.setStateHandler = this.setStateHandler.bind(this);

  };

  setStateHandler() {

    var item = "setState..."

    var myArray = this.state.data.slice();

    myArray.push(item);

    this.setState({data: myArray})

  }

}
```

```

};

render() {

  return (

    <div>

      <button onClick = {this.setStateHandler}>SET STATE</button>

      <h4>State Array: {this.state.data}</h4>

    </div>

  );

}

}

export default App;

```

ما کار خود را با یک آرایه ی خالی آغاز کرده ایم. هر بار که بر روی دکمه کلیک کنیم، حالت مورد نظر ما به روز رسانی می شود. اگر 5 بار بر روی دکمه کلیک کنیم در این صورت نتیجه زیر را شاهد خواهیم بود.



## Force Update

گاهی اوقات نیاز است که جزء خود را به صورت دستی به روز رسانی کنیم. برای انجام این کار می توانیم از متد `forceUpdate` استفاده کنیم.

```
import React from 'react';
```

```
class App extends React.Component {  
  
  constructor() {  
  
    super();  
  
    this.forceUpdateHandler = this.forceUpdateHandler.bind(this);  
  
  };  
  
  forceUpdateHandler() {  
  
    this.forceUpdate();  
  
  };  
  
  render() {  
  
    return (  
  
      <div>  
  
        <button onClick = {this.forceUpdateHandler}>FORCE UPDATE</button>  
  
        <h4>Random number: {Math.random()}</h4>  
  
      </div>  
  
    );  
  
  }  
  
}  
  
export default App;
```

می خواهیم عددی تصادفی را تنظیم کنیم که هر بار که بر روی دکمه کلیک می کنیم به روز شود.



## Find Dom Node

جهت دستکاری DOM میتوانیم از متد `ReactDOM.findDOMNode()` استفاده کنیم. ابتدا باید `react-dom` را ایمپورت کنیم.

```
import React from 'react';

import ReactDOM from 'react-dom';

class App extends React.Component {

  constructor() {

    super();

    this.findDomNodeHandler = this.findDomNodeHandler.bind(this);

  };

  findDomNodeHandler() {

    var myDiv = document.getElementById('myDiv');

    ReactDOM.findDOMNode(myDiv).style.color = 'green';

  }

  render() {

    return (

      <div>

        <button onClick = {this.findDomNodeHandler}>FIND DOME NODE</button>

        <div id = "myDiv">NODE</div>

      </div>

    );

  }

}
```

```
export default App;
```

بعد از اینکه بر روی دکمه کلیک می شود، عنصر myDiv به رنگ سبز در می آید.



از به روز رسانی 0.14 ، اغلب متدهای API اجزای قدیمی منسوخ یا حذف شده اند تا با ES6 سازگار شوند.