

# ReactJS - Component life cycle

Easy Frontend - Nơi kiến thức Frontend được chia sẻ một cách đơn giản, dễ hiểu và đặc biệt là phải vui ☺

Mình nói gì trong video này

- . Component life cycle là gì?
- . Có những life cycle nào? Sử dụng ra sao?
- . Lỗi can't setState() on unmounted component

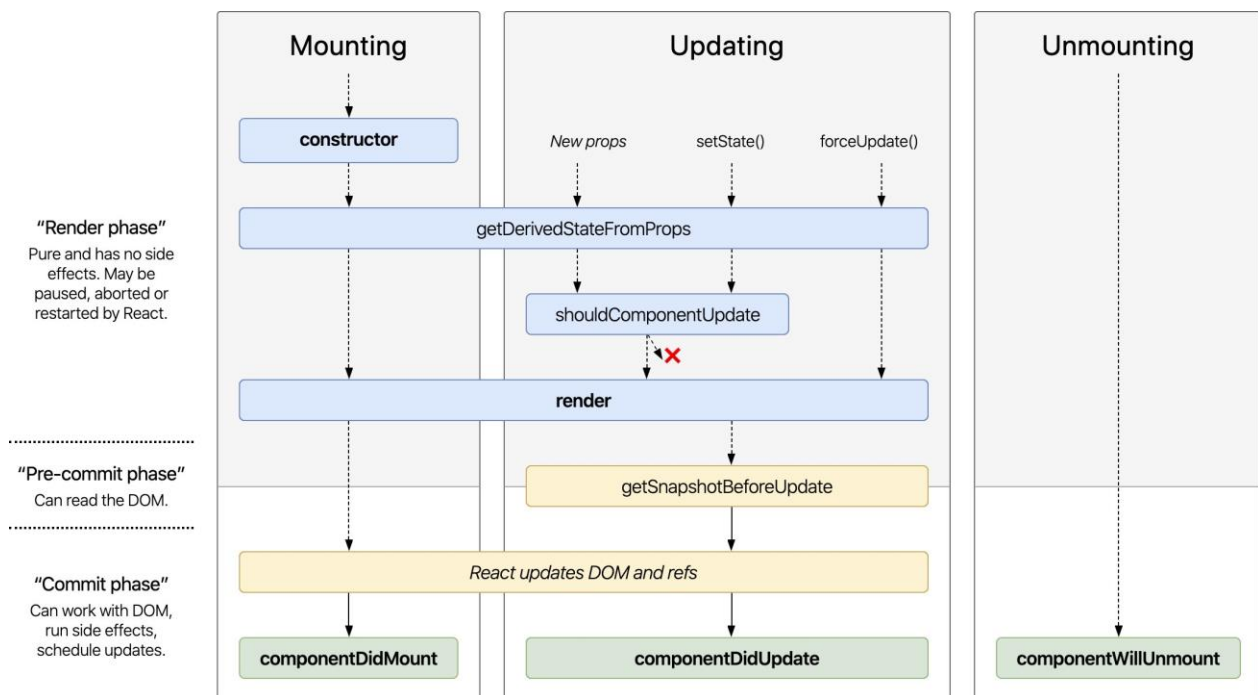
Life cycle = Vòng đời

## Life cycle của component trong ReactJS là gì?

Với component trong ReactJS, life cycle gồm 3 giai đoạn:

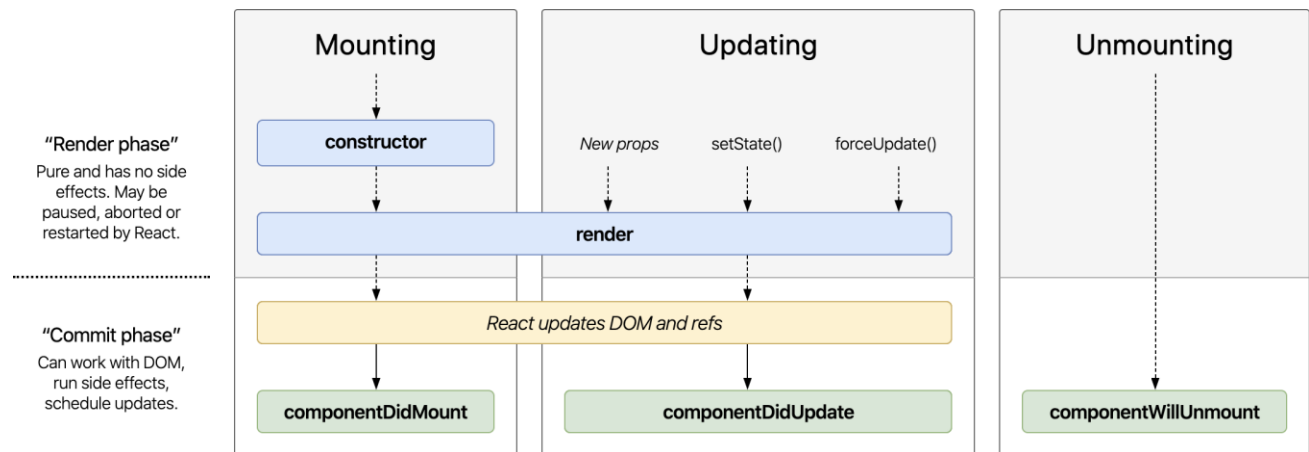
- Được tạo ra (Mounting)
- Qua nhiều thay đổi (Updating)
- Và bị huỷ bỏ (Unmounting)

ReactJS life cycle (full version)



Link component life cycle diagram: <http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/>

## Bảng life cycle rút gọn (nên dùng cái này)



□ Không sử dụng các life cycle này nữa:

- `componentWillMount()`
- `componentWillReceiveProps()`

### ? Component vs PureComponent

- Nên dùng `PureComponent`
- Vì có shallow comparison trong hàm `shouldComponentUpdate()`
- Link tham khảo: <https://stackoverflow.com/questions/41340697/react-component-vs-react-purecomponent/53740921>

## `constructor()`

- Được phép dùng.
- Nhớ có `super(props)`
- Khai báo state.
- Định nghĩa properties của component.

```
class App() extends PureComponent {
  constructor(props) {
    super(props);

    this.DEFAULT_MAX_LENGTH = 10;
    this.state = {
      productList: [],
    };
  }
}
```

## componentDidMount()

- Được phép dùng.
- Khởi tạo dữ liệu cho component: gọi API, biến đổi dữ liệu, cập nhật state.
- Gửi tracking page view (GA, FacebookPixel, ...)

```
class HomePage extends PureComponent {
  constructor(props) {
    super(props);

    this.state = {
      loading: true,
      productList: [],
    };
  }

  async componentDidMount() {
    try {
      // Send GA page view tracking
      analytics.page('Home page');

      const productList = await productApi.getAll();
      this.setState({
        productList,
        loading: false,
      });
    } catch (error) {
      console.log('Failed to fetch product list: ', error);
      this.setState({loading: false});
    }
  }

  render() {
    const {loading, productList} = this.state;
    if (loading) return <Loader />;

    return <ProductList productList={productList}>
  }
}
```

## componentWillUnmount()

- Được phép dùng.
- Clear timeout hoặc interval nếu có dùng.
- Reset dữ liệu trên redux nếu cần thiết.

```
class Countdown extends PureComponent {
  constructor(props) {
    super(props);
    this.state = {
      currentSecond: 0,
    };
  }

  componentDidMount() {
    this.timer = setInterval(() => {
      this.setState(prevState => ({
        currentSecond: prevState.currentSecond - 1,
      }));
    }, 1000);
  }

  componentWillUnmount() {
    if (this.timer) {
      clearInterval(this.timer);
    }
  }

  render() {
    const {currentSecond} = this.state;
    return <p>{currentSecond}</p>;
  }
}
```

## componentDidUpdate()

- Cục kỳ hạn chế dùng
- **ADVANCED** Chỉ dùng nếu muốn handle update component khi click nút back mà trên URL có query params.

## Lỗi can't setState() on unmounted component

```
✖ Warning: Can't call setState (or
forceUpdate) on an unmounted component. This is a no-op,
but it indicates a memory leak in your application. To
fix, cancel all subscriptions and asynchronous tasks in
the componentWillUnmount method.
    in TagCloud (at WordCloudChart.js:90)
    in div (at WordCloudChart.js:82)
    in WordCloudChart (at WordCloudAggregation.js:31)
```

? Lý do:

- Ở trang `Home`, đang lấy dữ liệu từ API, sau đó update vào state.
- Nhưng ác thay, dữ liệu chưa lấy xong, user qua trang `About`
- Thế là component `Home` bị `unmount`.
- Ngay sau đó, dữ liệu từ API được trả về và tiếp tục gọi `setState()`
- Đau lòng thay, component `Home` có còn đâu mà update.

✓ Giải pháp

- Dùng một flag `isComponentMounted` để biết trạng thái của component.

```
class Home extends PureComponent {
  constructor(props) {
    super(props);

    this.isComponentMounted = false;
    this.state = {
      productList: [],
    }
  }

  async componentDidMount() {
    this.isComponentMounted = true;

    try {
      const productList = await productApi.fetchProductList();

      if (this.isComponentMounted) {
        this.setState({ productList });
      }
    } catch (error) {
      console.log('Failed to fetch data:', error);
    }
  }

  componentWillUnmount() {
    this.isComponentMounted = false;
  }

  render() {
    // Render something here ...
  }
}
```

