자바스크립트
비동기

2017.08.31

인간중심

윤주형 김성진 정구범

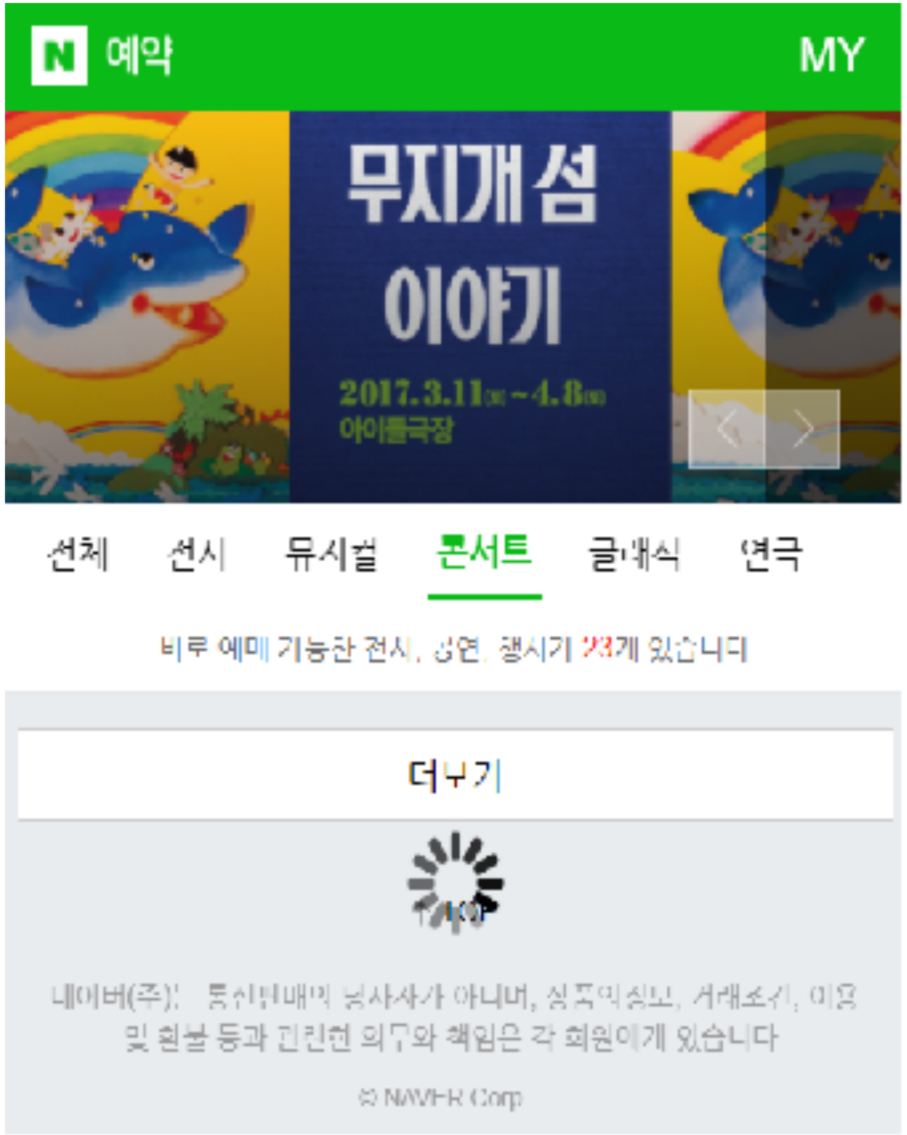
index

- 예약 시스템의 비동기
- promise
- generator / iterator
- async / await
- 결론

예약 시스템의 비동기

예약 시스템의 비동기

화면



N 예약 MY

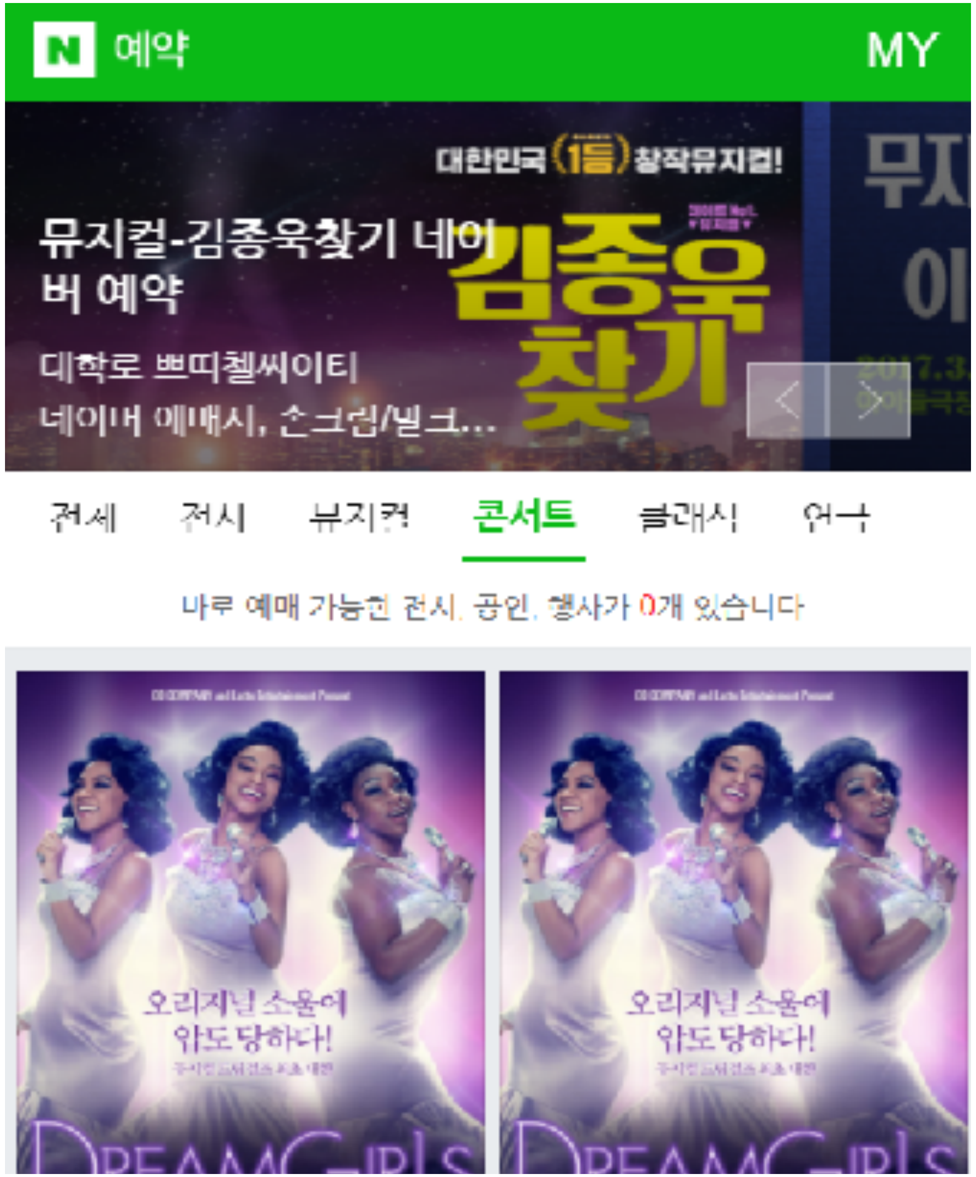
전체 전시 뮤지컬 **콘서트** 클래식 연극

바로 예매 가능한 전시, 공연, 행사가 23개 있습니다

더보기

네이버(주)는 통신판매의 당사자가 아니며, 상품의 정보, 거래조건, 이용 및 환불 등과 관련한 의무와 책임은 각 회원에게 있습니다

© NAVER Corp



N 예약 MY

뮤지컬-김종욱찾기 네이버 예약

대학로 쁘띠첼씨어터

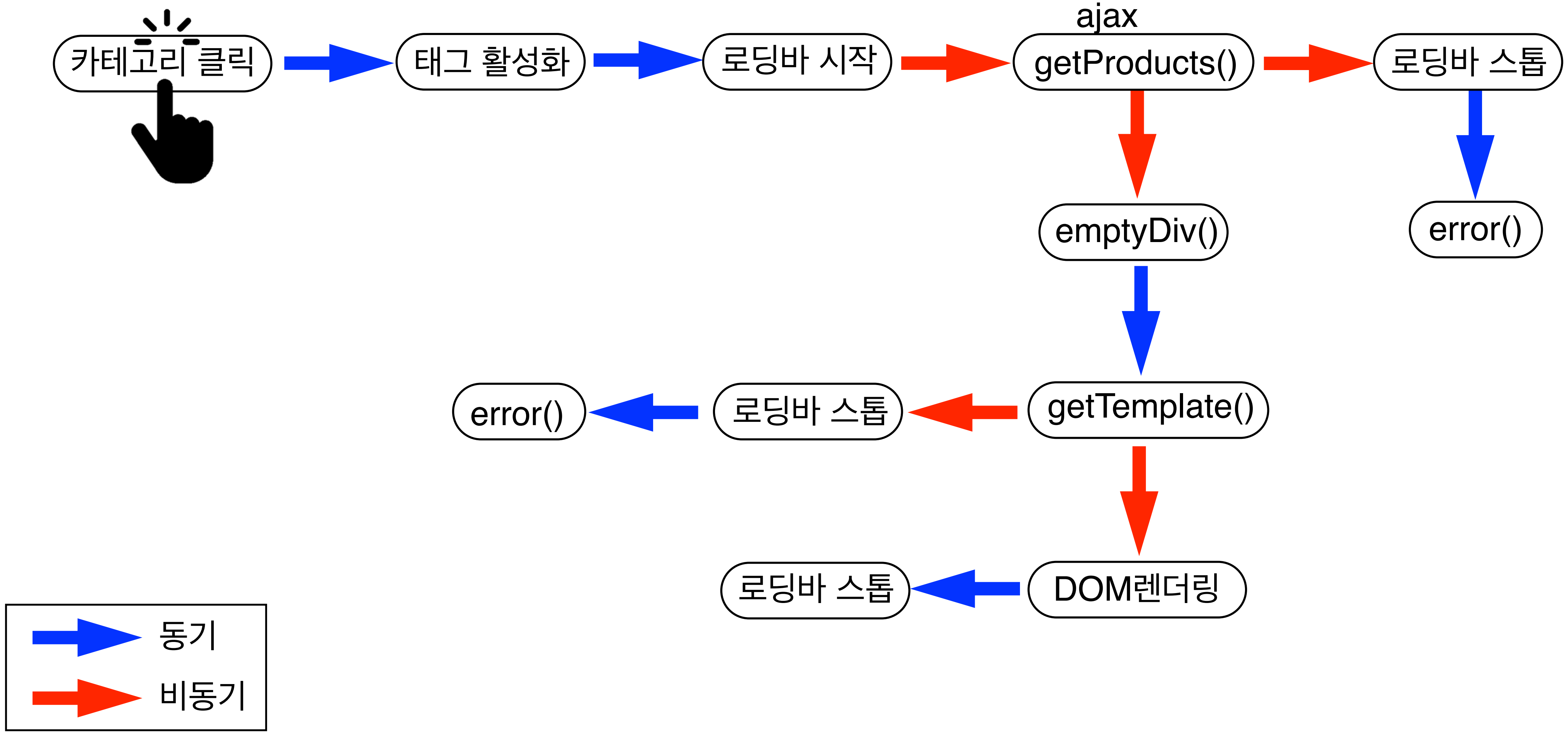
네이버 예매시, 스크린/빌크...

전체 전시 뮤지컬 **콘서트** 클래식 연극

바로 예매 가능한 전시, 공연, 행사가 0개 있습니다

예약 시스템의 비동기

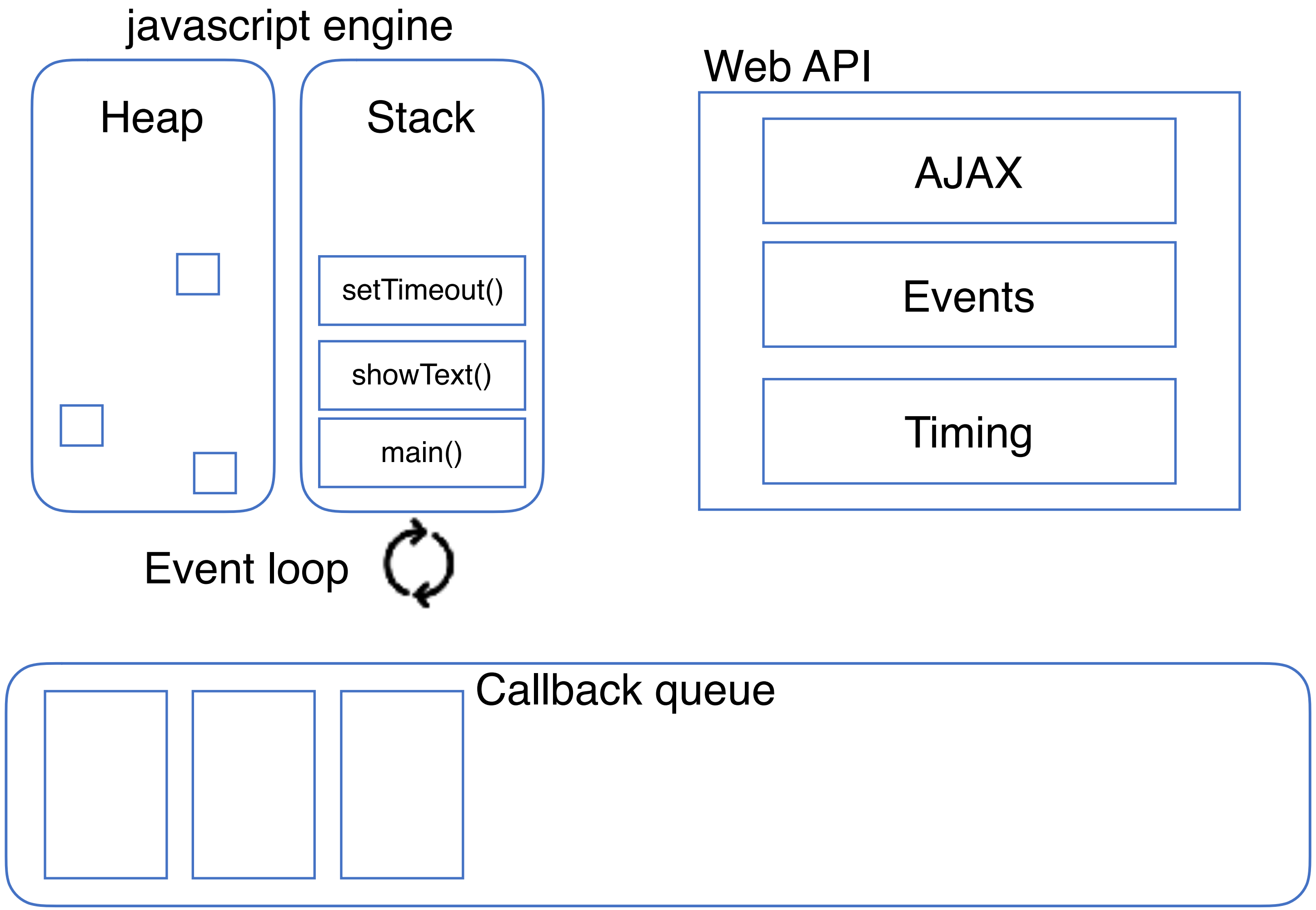
비동기 flow



Callback

Callback

콜백 동작



Callback

콜백으로 구현한 코드

```
function setEvent(){
    $(categories).on("click", "a", function () {
        activeCategory(e.currentTarget);
        let page = 1;
        spinner.spin(spinTarget, function () {
            getProducts(activeCategory, page, function (data) {
                if (data.length == 0) {
                    spinner.stop();
                    error();
                } else {
                    emptyTarget();
                    getProductTemplate.putData(...params, function (result) {
                        if (result.length == 0) {
                            spinner.stop();
                            error();
                        } else {
                            productCount = categories.data("productcount");
                            productCountTarget.html(productCount);
                        }
                        spinner.stop();
                    });
                }
            );
        }
    }
}
```

```
function setEvent(){
    $(categories).on("click", "a", function () {
        activeCategory(e.currentTarget);
        let page = 1;
        spinner.spin(spinTarget, function () {
            getProducts(activeCategory, page, function (data) {
                if (data.length == 0) {
                    spinner.stop();
                    error();
                } else {
                    emptyTarget();
                    getProductTemplate.putData(...params, function (result) {
                        if (result.length == 0) {
                            spinner.stop();
                            error();
                        } else {
                            productCount = categories.data("productcount");
                            productCountTarget.html(productCount);
                        }
                        spinner.stop();
                    });
                }
            );
        }
    }
}
```

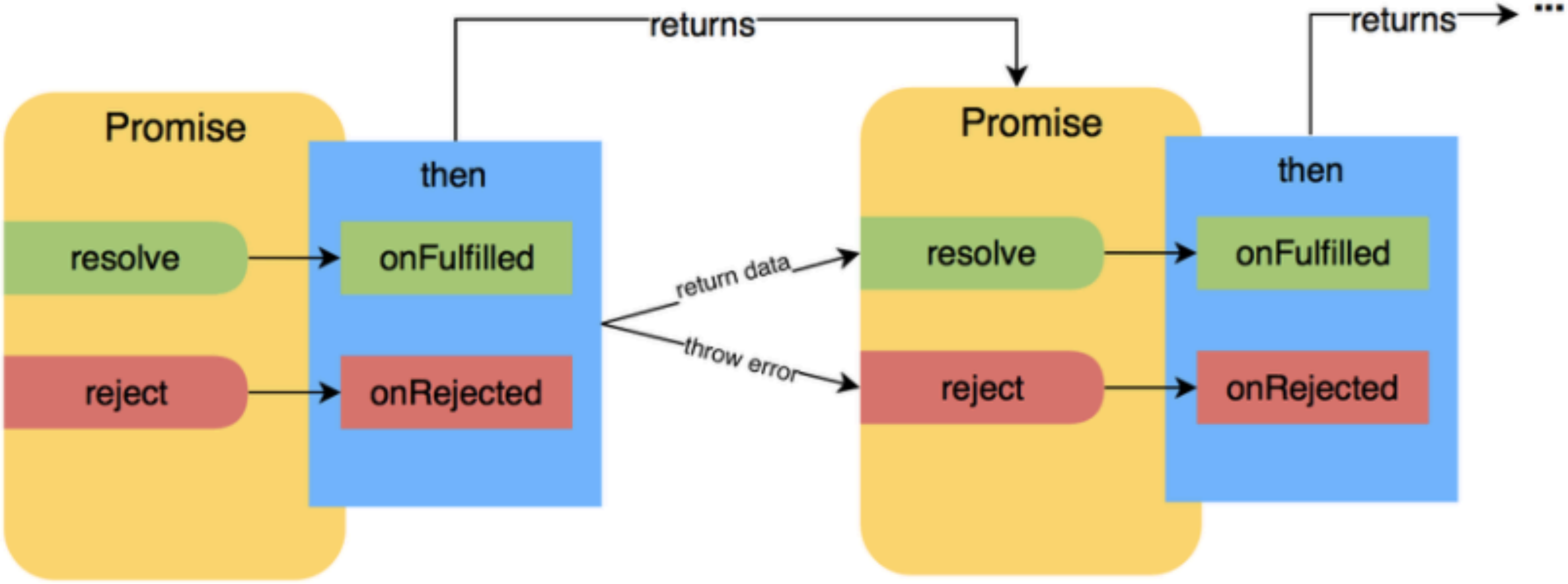
```
function setEvent(){
    $(categories).on("click", "a", function () {
        activeCategory(e.currentTarget);
        let page = 1;
        spinner.spin(spinTarget, function () {
            getProducts(activeCategory, page, function (data) {
                if (data.length == 0) {
                    spinner.stop();
                    error();
                } else {
                    emptyTarget();
                    getProductTemplate.putData(...params, function (result) {
                        if (result.length == 0) {
                            spinner.stop();
                            error();
                        } else {
                            productCount = categories.data("productcount");
                            productCountTarget.html(productCount);
                        }
                        spinner.stop();
                    });
                }
            );
        }
    }
}
```

```
function setEvent(){
    $(categories).on("click", "a", function () {
        activeCategory(e.currentTarget);
        let page = 1;
        spinner.spin(spinTarget, function () {
            getProducts(activeCategory, page, function (data) {
                if (data.length == 0) {
                    spinner.stop();
                    error();
                } else {
                    emptyTarget();
                    getProductTemplate.putData(...params, function (result) {
                        if (result.length == 0) {
                            spinner.stop();
                            error();
                        } else {
                            productCount = categories.data("productcount");
                            productCountTarget.html(productCount);
                        }
                        spinner.stop();
                    });
                }
            );
        }
    }
}
```

Promise

Promise

ES6의 Promise 사용



Promise

promise로 개선된 코드

```
$(categories).on("click", "a", function () {
    activeCategory(e.currentTarget);
    let page = 1;
    spinner.spin(spinTarget)
    .then(function() {
        return getProducts(activeCategory, page);
    }).then(function (result) {
        emptyTarget();
        return getProductTemplate.putData(...result);
    }).then(function(result){
        const productCount = categories.data("productcount");
        productCountTarget.html(productCount);
    }).catch(function(error){
        return errorHandler(error);
    }).then(spinner.stop());
});
```

Promise

promise로 개선된 코드

```
$(categories).on("click", "a", function () {
    activeCategory(e.currentTarget);
    let page = 1;
    spinner.spin(spinTarget)
    .then(function() {
        return getProducts(activeCategory, page);
    }).then(function (result) {
        emptyTarget();
        return getProductTemplate.putData(...result);
    }).then(function(result){
        const productCount = categories.data("productcount");
        productCountTarget.html(productCount);
    }).catch(function(error){
        return errorHandler(error);
    }).then(spinner.stop());
});
```

Promise

동기인듯 동기아닌 동기같은 비동기

```
function dreamCode(event){
    let page = 1;
    activeCategory(event.currentTarget);
    try{
        spinner.spin(spinTarget);
        let data = getProducts(activeCategory, page);
        emptyTarget();
        getProductTemplate.putData(...params);
        const productCount = categories.data("productcount");
        productCountTarget.html(productCount);
    } catch(error){
        errorHandler(error);
    }
    spinner.stop();
};
```

Generator / Iterator

Generator / Iterator

Generator / Iterator 기본 사용법

```
function* generator(){
    let a = yield 10;
    let b = yield a+20;
    let c = yield b+30;
    return a + b + c; //1 + 2 + 3
}

let iterator = generator();

console.log(iterator.next()); //done : false, value : 10;
console.log(iterator.next(1)); //done : false, value : 21;
console.log(iterator.next(2)); //done : false, value : 32;
console.log(iterator.next(3)); //done : true, value: : 6;
```

Generator / Iterator

Generator / Iterator + Promise

```
function* generator(event){
    let page = 1;
    activeCategory(event.currentTarget);
    try{
        yield spinner.spin(spinTarget);
        let data = yield getProducts(activeCategory, page);
        emptyTarget();
        yield getProductTemplate.putData(...params);
        const productCount = categories.data("productcount");
        productCountTarget.html(productCount);
    } catch(error){
        errorHandler(error);
    }
    spinner.stop();
}
```

Generator / Iterator

Generator / Iterator + Promise

```
function runner(generator, onEnd) {
    const iter = generator();
    let result = iter.next();

    function loop(result) {
        if (result.done) {
            if (onEnd !== undefined) {
                onEnd(result.value);
            }
        } else {
            if (!(result.value instanceof Promise)) {
                loop(iter.next(result.value));
            } else {
                result.value.then(value => {
                    loop(iter.next(value));
                });
                loop(result);
            }
        }
    }
```

async / await

async / await

동기같은 코드

```
async function ayncFunction(event){
    let page = 1;
    activeCategory(event.currentTarget);
    try{
        spinner.spin(spinTarget);
        let data = await getProducts(activeCategory, page);
        emptyTarget();
        await getProductTemplate.putData(...params);
        const productCount = await categories.data("productcount");
        productCountTarget.html(productCount);
    } catch(error){
        errorHandler(error);
    }
    spinner.stop();
};
```

결론

Thank you