



Что такое RxJS?

“LoDash для событий”

Что такое RxJS?

“LoDash для асинхронных действий”

Типы асинхронных действий в браузере

- AJAX
- User Events (clicks, mousemoves, keyups, др.)
- Animations
- Sockets
- Workers



Callbacks

Самый примитивный способ работы с асинхронностью

```
getSomeData((data) => {  
  console log('data received', data);  
})
```

Callback Hell (aka “the flying V”)

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```

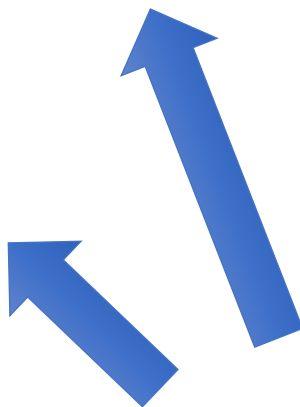


Promise - более чистое решение

```
getSomeData(id)
  .then(data => {
    doStuff(data);
    return getSomeData(data.parentId);
  })
  .then(data => {
    doStuff(data);
    return getSomeData(data.parentId);
  })
```

Promise

- Guaranteed Future
- Immutable
- Single Value



Эти два ограничения – проблема в современных WEB приложениях

Promise'ы нельзя отменить



План приёма пациентов



Печать



21 ноября 2017, Вт



Поиск по ФИО, пол...

Найти

3 Экстренные

Ильина Галина Петровна

УЗИ органов брюшной полости

Чарьев Батыр Рамазанович

УЗИ гепатобилиарной системы (печени, желчный пузырь, желчные протоки, поджелудочная железа)

Кремнева Нина Федоровна

УЗИ желчного пузыря с определением функции
[Дополнительное:](#) УЗИ органов брюшной полости

3 Ожидающие

14:00

Волнистова Марина Вячеславовна

УЗИ желчного пузыря с определением функции
[Дополнительное:](#) УЗИ органов брюшной полости

18:00

Чарьев Батыр Рамазанович

Всего: 6

Подробная информация о записи

Чарьев Батыр Рамазанович

М, 28 лет (08 декабря 1988 года), полис ОМС: 7747110841003130

Информация о пациенте

Контактные данные

Прикрепления

СМО: ООО ВТБ МЕДИЦИНСКОЕ СТРАХОВАНИЕ

ЛПУ: ГП 45 Ф 5 (ГП 154)

Тип: Терапевтический

Участок: Участок №15

Диагностическая задача

Вид исследования:

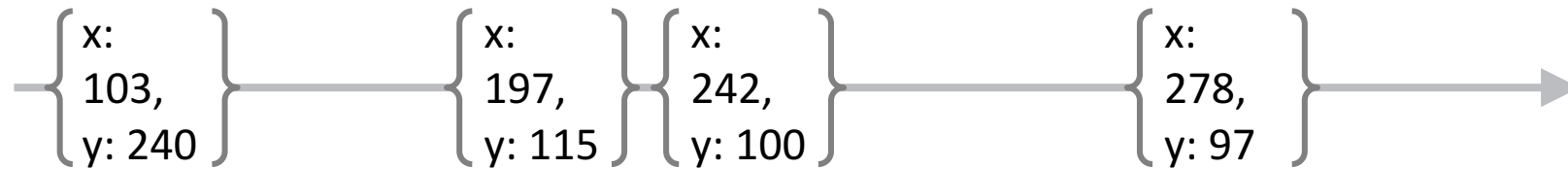
– Рентгенография черепа обзорная

Начать приём

Observable

- Любое кол-во значений
- Распределены во времени
- Могут быть отменены
- «Ленивые» - Lazy

Observable – массив событий



Observable – массив событий

$$\left[\begin{array}{l} \left\{ \begin{array}{l} x: \\ 103, \\ y: 240 \end{array} \right\}, \left\{ \begin{array}{l} x: \\ 197, \\ y: 115 \end{array} \right\}, \left\{ \begin{array}{l} x: \\ 242, \\ y: 100 \end{array} \right\}, \left\{ \begin{array}{l} x: \\ 278, \\ y: 97 \end{array} \right\} \end{array} \right]$$

Observable – массив событий

И че?

Массив можно трансформировать

map, filter, reduce

Массивы можно объединять

merge, concat, zip

Имеет отдельный канал
ошибок, как и у Promise

```
myObservable.catch(err => Observable.of('handled'))
```

Observables – Lazy, мы можем
переподписаться когда
захотим

repeat, retry

Observable добавляет другое измерение: ВРЕМЯ

combineLatest, buffer, debounce, throttle, delay, timeout, retryWhen

А теперь давайте напишем
код