



Microsoft Partner
Silver Learning



JavaScript. Расширенные ВОЗМОЖНОСТИ

Библиотека RxJS. Часть 2



ITVDSN
IT VIDEO DEVELOPERS NETWORK

JavaScript. Расширенные возможности

Автор курса



Патёха Сергей



MISTER SERGII PATOKHA

Has successfully completed the requirements to be recognized as a Microsoft Certified Professional.

Date of achievement: 06/07/2018
Certification number: GB48-0117

Satya Nadella
Chief Executive Officer

Microsoft
CERTIFIED
Professional

MCID: 16014173

JavaScript. Расширенные возможности

После урока обязательно



Повторите этот урок в видео формате на
[ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на
[TestProvider.com](http://testprovider.com)

Библиотека RxJS. Часть 2



JavaScript. Расширенные возможности

План

1. Что такое RxJS?
2. Что такое Observable, Observer?
3. Для чего применяется RxJS?
4. Шаблон Observable.
5. Push/pull стратегии.
6. Понятие потоков.
7. Создание потоков.
8. Знакомство с операторами.
9. Операторы комбинирования.
10. Определение Subject.

JavaScript. Расширенные возможности

RxJS operators

Операторы слияния потоков – используются для соединения 2 и более разных потоков в один для удобной обработки входящих параметров из источника.

- Concat;
- Merge;
- Zip;
- CombineLatest;
- WithLatestFrom;
- ConcatMap;
- MergeMap;
- SwitchMap.



JavaScript. Расширенные возможности

RxJS Subject

Subject – объект, объединяющий одновременно Observable и Observer. Он позволяет подписываться на изменения потока и одновременно может задавать эти изменения (выдавать).

Одно из преимуществ Subject – возможность предоставлять новую информацию подписчикам (пользователям) без ограничения или привязки к какому либо потоку данных извне.

Создавая Observable – необходимо задать функцию, отображающую каким образом нужно создать значения.

Создавая Subject – также создается наблюдаемый объект, и потом из любого места кода можно генерировать (отдавать) ему любое значение.



JavaScript. Расширенные возможности

RxJS Subject

Subject используется для создания одного источника для нескольких подписчиков (аналогия EventEmitter).

Subject – позволяет работать извне с методами `next()`, `error()`, `complete()`. Также вызывая метод `pipe` - работать со всеми известными операторами.

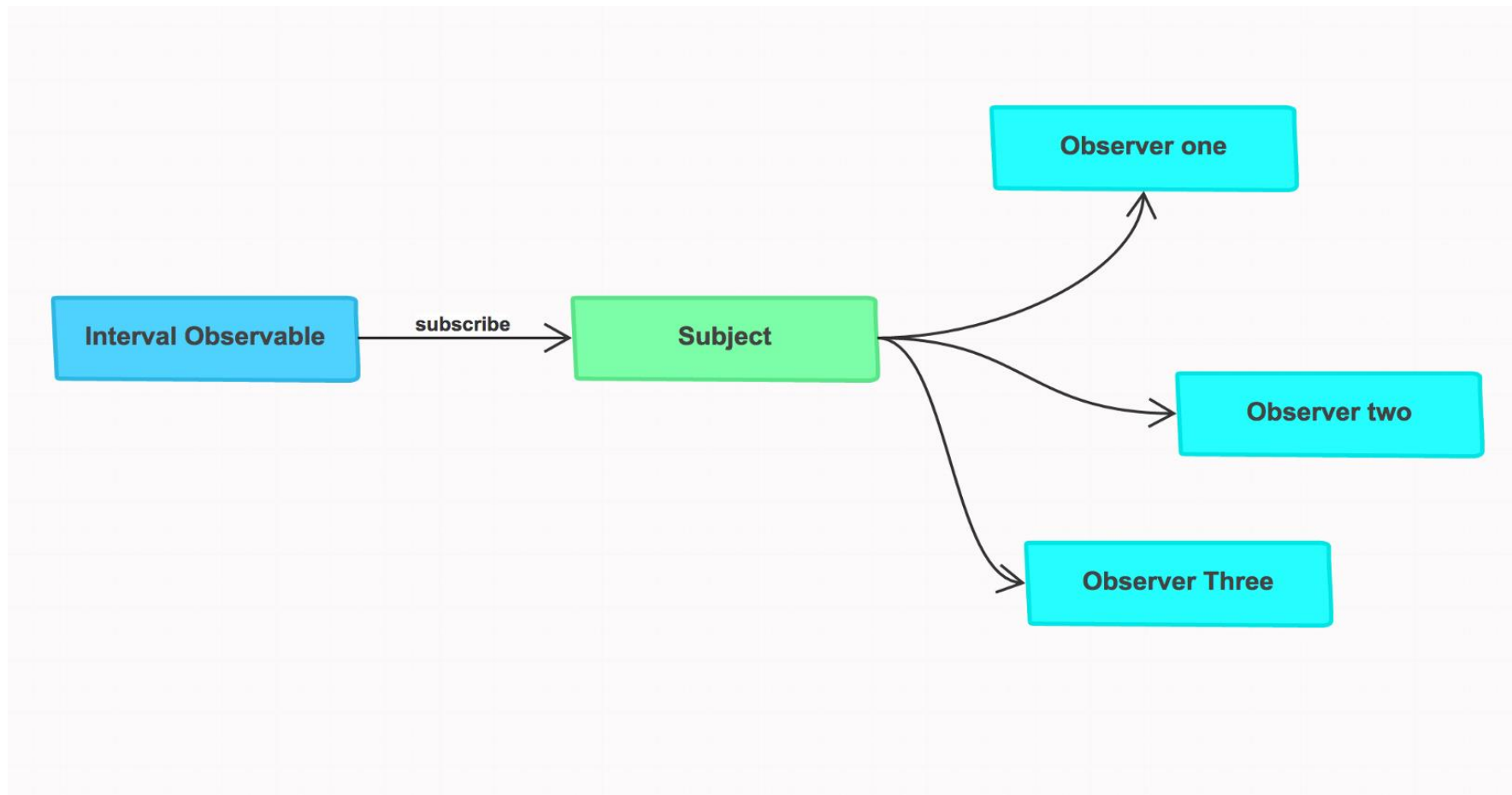


RxJS

Создавать собственные `subject` необходимо только тогда, когда некоторые методы создания `Observable` (наблюдаемого объекта `from`, `of`) – не оптимальное решение для создания готового потока.

JavaScript. Расширенные ВОЗМОЖНОСТИ

RxJS Subject



JavaScript. Расширенные возможности

Разновидности Subject

BehaviourSubject – поддерживает позднюю подписку. Может быть создан с первоначальным значением, и далее продолжать работу. Основная задача – всегда отдавать что-либо подписчикам. Даже если они опоздали на подписку потока.

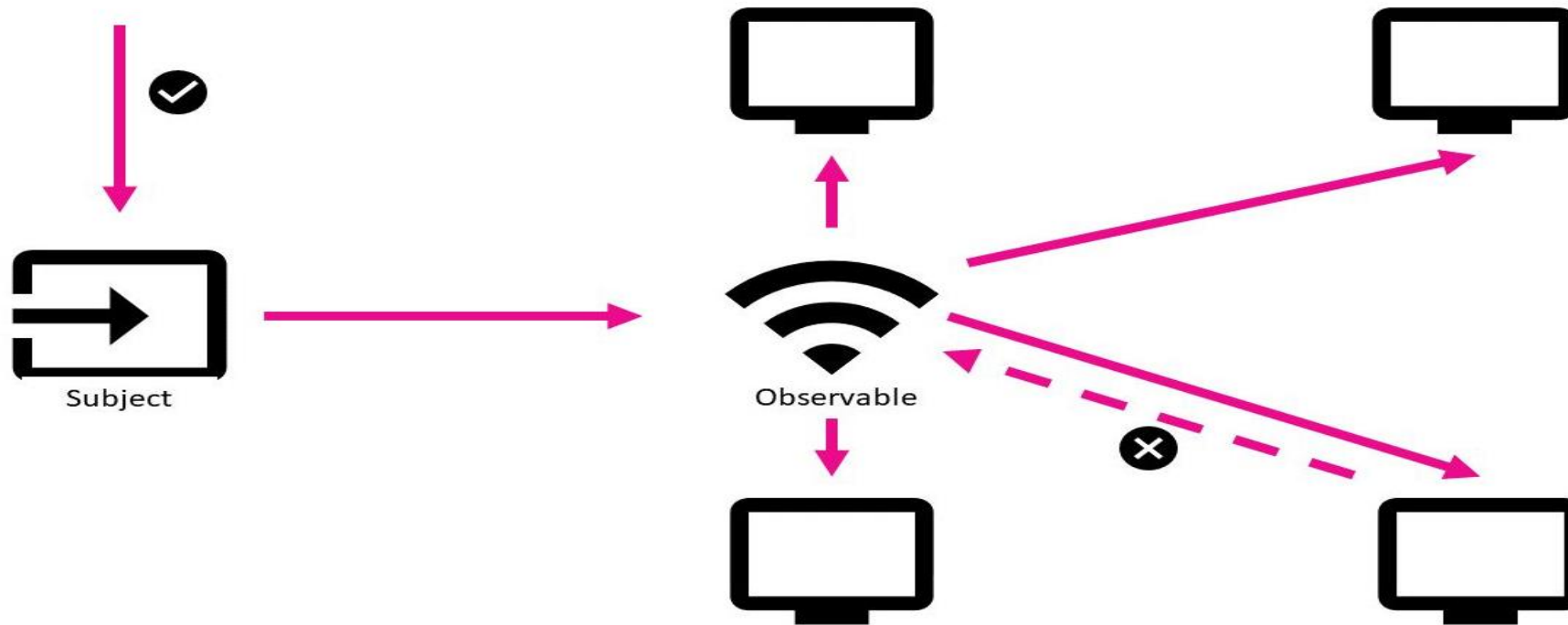
ReplaySubject – ограничивает количество триггеров (работает как буфер – с ограничениями). Позволяет более поздним подписчикам – получать все предыдущие данные из потока.

AsyncSubject – не будет выдавать значение, пока не закончится поток (complete). Забирает всегда только последнее значение для всех подписчиков (ранних, поздних). Идеально подходит для вычислительных функций.

JavaScript. Расширенные возможности

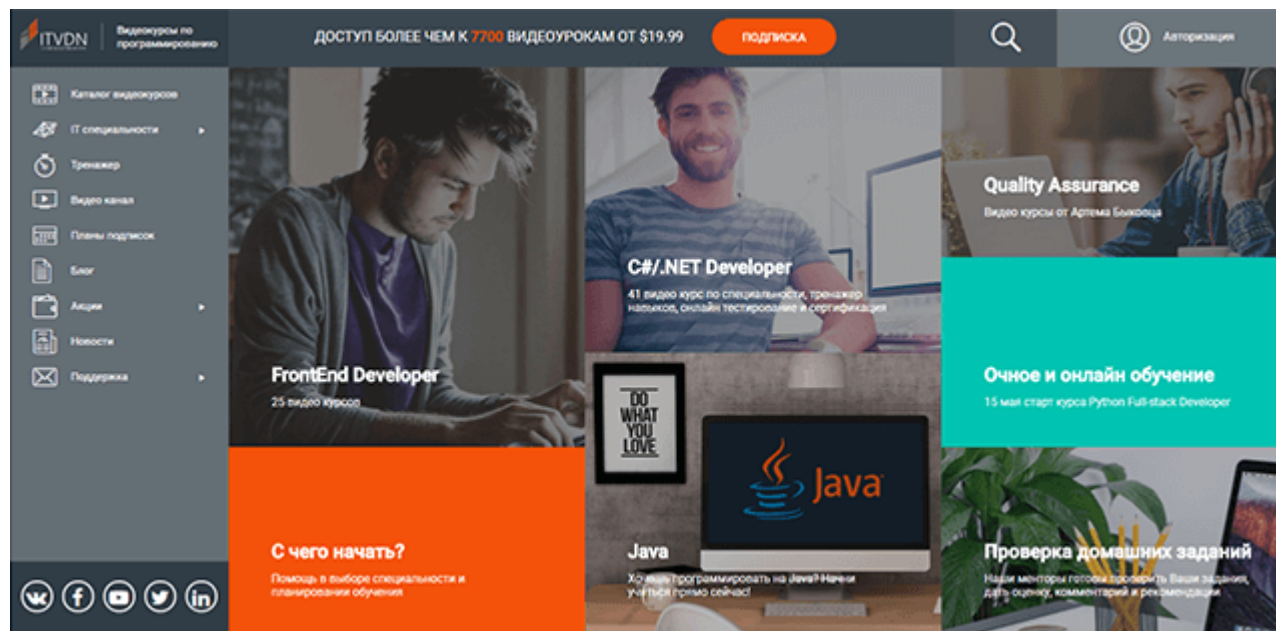
Share

Share() – позволяет вернуть новый Subject, поток которого будет транслироваться всем подписчикам. Как только не окажется ни одного – самостоятельно отменяет подписку на Subject (unsubscribe).



Смотрите наши уроки в видео формате

ITVDN.com



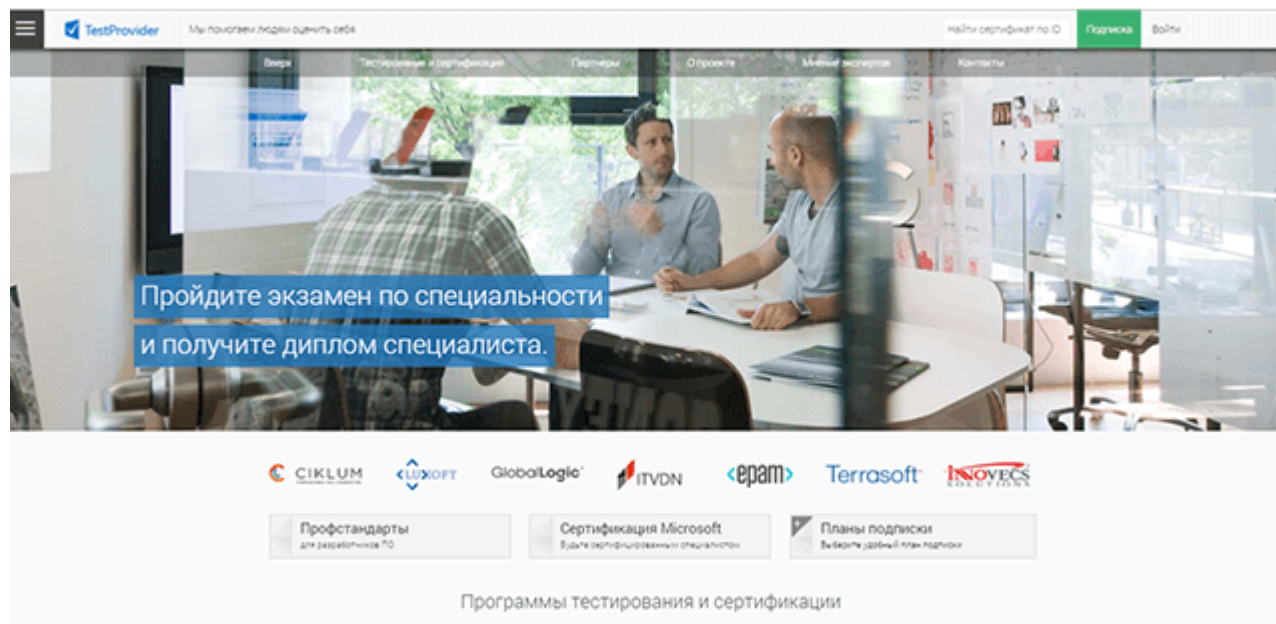
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



Q&A

Информационный видеосервис для разработчиков программного обеспечения

