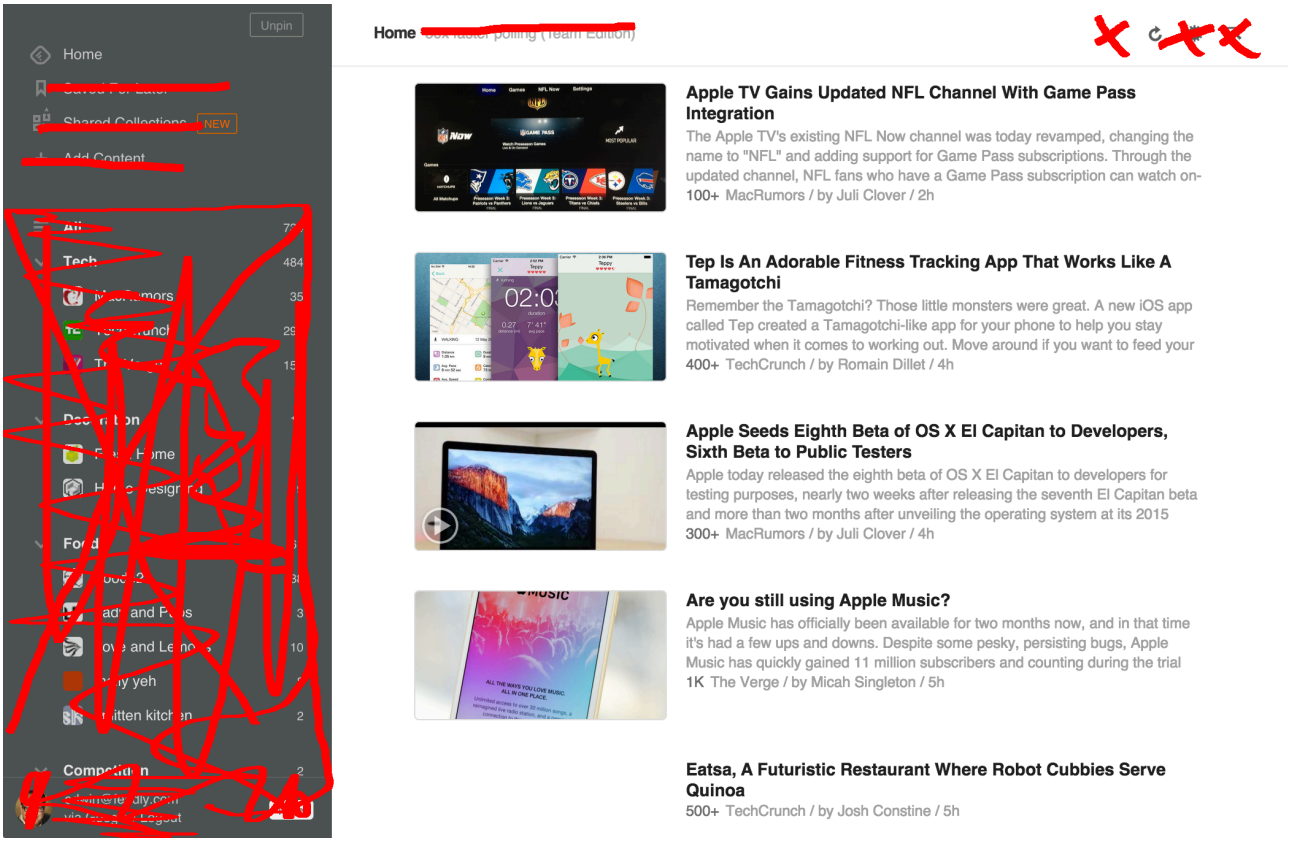
Новостной агрегатор

В результате хочется получить что-то вроде: https://s3.feedly.com/production/head/images/landing/screenshot-web@2x.png

Только значительно упростим дизайн для первой версии



То есть слева будет лишь пункт меню “Home”, справа будет список новостей и кнопка обновить.

При открытии новости будет появляться попапа на всю страницу, белый, по центру будет выводиться контент вместе с заголовком и будет крестик в верхнем правом углу для закрытия его и возвращения к главному экрану.

Инфраструктуру у нас должна состоять из двух больших составляющих. Это frontend и backend.

Frontend - тут все предельно понятно, сверстать примерно как на скриншотах, подключить API и работает. Используем react + redux. Для попапа я бы посоветовал использовать react-router, то есть попапа будет просто отдельной страницей(компонентом) и все открытие попапа сведется к переходу по ссылке.

Backend - тут все интереснее. Нам нужно три состовляющих. Это база данных, демон для парсинга новостных сайтов и API для сайта.

БД

Базу данных предлагаю выбрать mongo, так как достаточно простая и легке использовать. Для нее есть адаптер, который ещё больше упрощает жизнь, это http://mongoosejs.com/. На сайте в основном примеры с callback, но он так же работает и на промисах. У них есть вот такой гайд http://mongoosejs.com/docs/index.html и документация, если что, вопросы по нему можно писать мне, я подскажу.

Пока в базе будет одна таблица(модель) articles, в нее будешь класть спаршеные новости, и из нее будешь их получать в API.

Парсинг

Про демона для парсинга, тут все предельно просто, это будет обычная JS функция, которая будет вызываться через SetInterval, и внутри себя уже вызывать функции парсинга для каждого из доступных сайтов.

Пока не будет парсить сам HTML, а ограничимся API. Вот есть великолепный API у NYTimes https://developer.nytimes.com/, там есть пункт https://developer.nytimes.com/top_stories_v2.json#/Documentation/GET/%7Bsection%7D.%7Bformat%7D, вот его давай пока парсить, начать можно пока с одной секции, а потом парсить остальные.

API

По поводу API все предельно просто, пока что один endpoint, по запросу на который передаются 20 новостей. у него есть один параметр offset, по умолчанию разный нулю, это то, с какого отступа от начала будут браться эти 20 новостей, то есть если offset=0, то первые 20, если offset=20 то 21-40 новости и так далее.

Про то, как работать с mongoose можно почитать тут:
http://ru.smedialink.com/razrabotka/web-servis-na-node-js-i-express-js-chast-1-samoe-nachalo/
https://sohabr.net/habr/post/213931/
http://blog.rukomoynikov.ru/avtorizatsiya-polzovatelej-express-js-mongo/

В последней видно, как работать с промисами в mongoose.

А про саму базу можно почитать тут:
http://jsman.ru/mongo-book/Glava-1-Osnovy.html
https://habrahabr.ru/post/74144/