

Интересные камеры (тестовое задание)

Требуется разработать одностраничное веб-приложение, которое позволяет просматривать список публичных камер, подключенных к Ivideon, и добавлять их в избранное.

Общие требования

Задание принимается в виде zip-архива, содержащего все исходные файлы. В архиве также должен быть текстовый файл README, описывающий, как запустить результат выполнения задания.

Если вам хотелось бы что-то добавить сверх задания, или что-то улучшить в предложенной реализации, но вам не хватает времени — просто напишите нам об этом.

Требования к представлению

Размеры элементов на странице должны пропорционально адаптироваться под ширину окна браузера; максимальная ширина страницы — 600px.

Требования к технологиям

JavaScript

ReactJS, Redux.

Нельзя использовать языки, компилируемые в JavaScript (CoffeeScript, TypeScript и т.п.). Использование ES2015+ приветствуется.

CSS

Можно использовать любые методологии, но нельзя применять готовые CSS-фрэймворки.

Требования к архитектуре

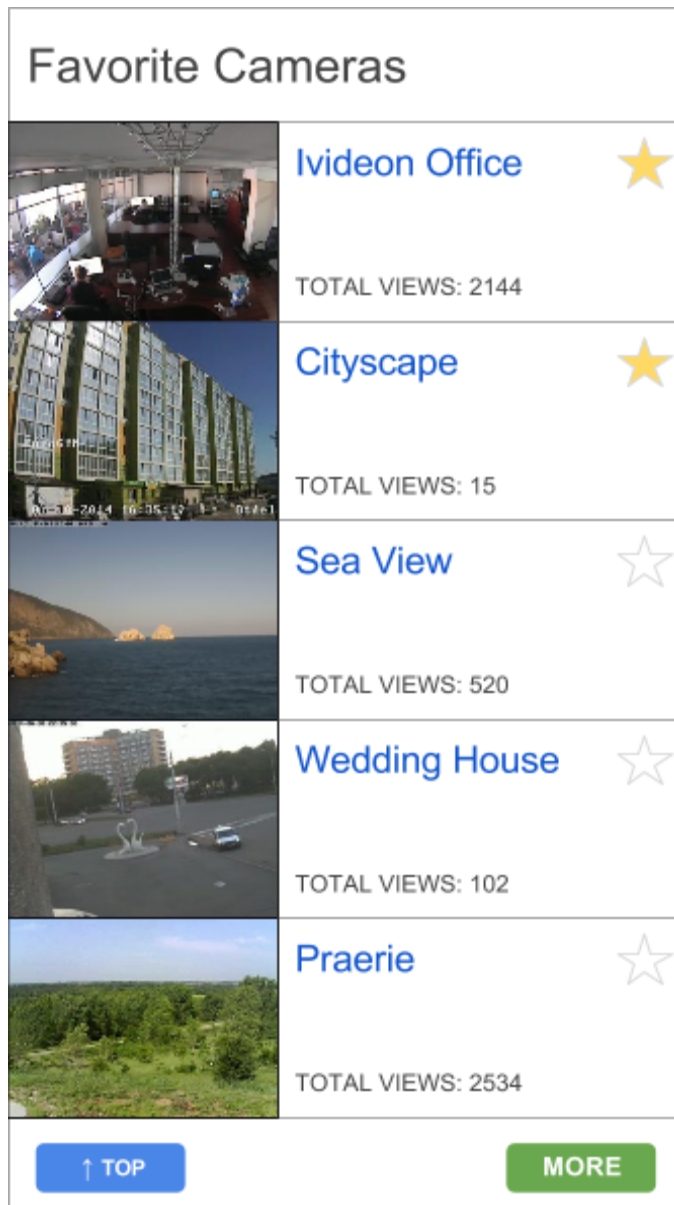
Архитектура приложения должна позволять легко развивать его (например, добавлять новые экраны, изменять поведение элементов управления).

Представьте, что разрабатываемые вами компоненты — часть большого приложения (потенциально повторно используемые). Подсказка: стандартные boilerplate для Redux не масштабируются на большие приложения.

Требования к функциональности

Приложение должно состоять из одного экрана с вертикальным списком камер.

Макет представлен ниже:



В списке должны выводиться публичные камеры, полученные через открытое API Ivideon. Документация по API получения списка камер — ниже.

Для каждой камеры в списке должно отображаться изображение предварительного просмотра с соотношением сторон 4:3. Нужно учесть, что к Ivideon подключены камеры с различным разрешением и соотношением сторон кадра. Документация по API получения изображений — ниже.

Для каждой камеры в списке должно отображаться название камеры и общее количество просмотров.

Добавление в избранное

Для каждой камеры в списке должна отображаться звездочка, которая при нажатии «добавляет камеру в избранное» (или, если камера добавлена, удаляет из избранного). Добавление в избранное должно приводить к тому, что камера после перезагрузки страницы не исчезнет из списка и окажется вверху.

Кнопка «НАВЕРХ»

Под списком камер должна располагаться кнопка «НАВЕРХ», которая прокручивает страницу к заголовку.

Кнопка «ЕЩЁ»

Под списком камер должна располагаться кнопка «ЕЩЁ», которая загружает новые камеры и добавляет их в конец списка.

Просмотр камеры

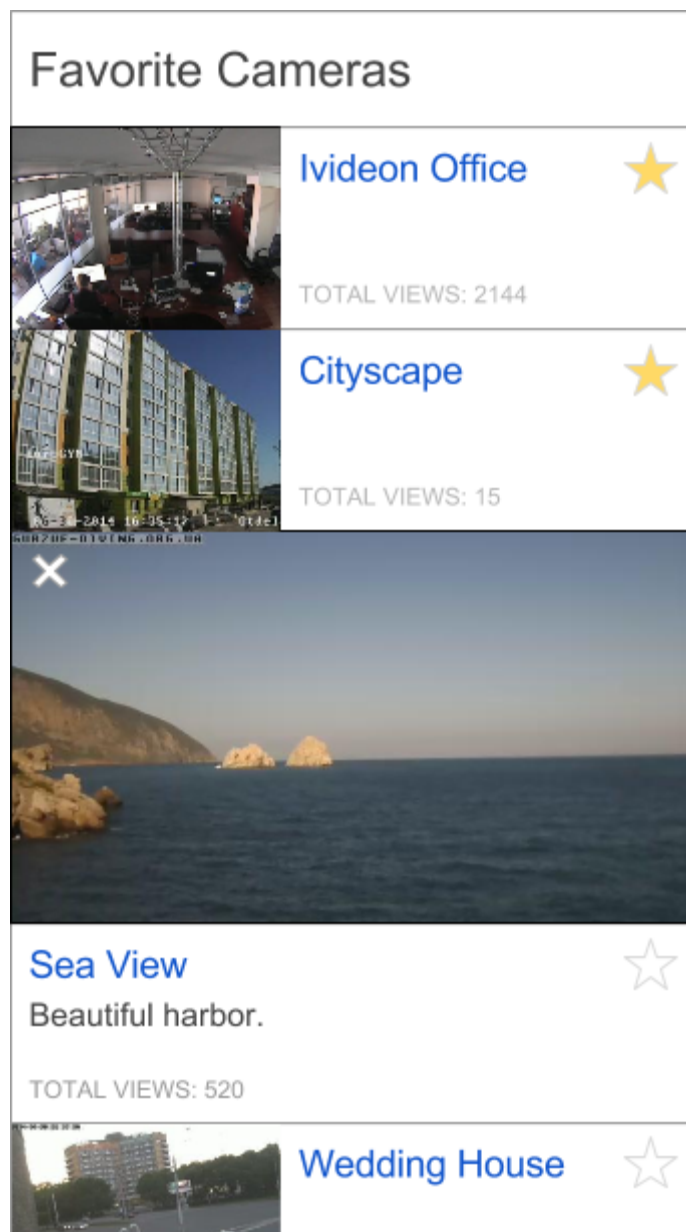
При нажатии на камеру в списке изображение предварительного просмотра должно развернуться на всю ширину списка. При этом должно быть отображено изображение целиком, в его оригинальных пропорциях (даже если в списке оно было обрезано).

Название, количество просмотров и звездочка должны оказаться под изображением. Между названием и количеством просмотров должно появиться описание камеры.

Открытое изображение должно обновляться раз в 10 секунд. По желанию: обновление изображения сделать плавным переходом (cross-fade) от старого к новому.

На открытом изображении должен появиться крестик, при нажатии на который элемент списка возвращается в свёрнутое состояние.

Макет представлен ниже:



Если изображение не удалось загрузить, вместо изображения нужно отобразить чёрный фон, сообщение о том, что камера недоступна, и кнопку «ОБНОВИТЬ».

При нажатии на кнопку «ОБНОВИТЬ» нужно повторить попытку загрузки изображения.

Макет представлен ниже.

Favorite Cameras



Ivideon Office



TOTAL VIEWS: 2144



Cityscape



TOTAL VIEWS: 15



Camera is offline.

REFRESH

Random Camera



Shows random images.

TOTAL VIEWS: 10

API для получения списка случайных публичных онлайн-камер

В представленной ниже документации отражены только те параметры и данные, которые необходимы для выполнения задания.

Поскольку API списка публичных камер выдаёт реальные онлайн-камеры пользователей, доступность которых зависит от их владельцев, возвращаемый список может изменяться со временем. По той же причине API не гарантирует полное отсутствие дубликатов при последовательных запросах — в ответах могут присутствовать камеры, которые уже были возвращены ранее.

Запрос	<code>GET</code> <code>http://api.ivideon.com/tv/cameras?jsonp={callback}&limit={limit}&seed={seed}</code>
Параметры запроса	jsonp — название JavaScript-функции, принимающей JS-объект с ответом. limit — максимальное количество камер в возвращаемом списке. Необязательный. seed — идентификатор страницы для постраничной навигации. Необязательный. Чтобы загрузить следующие камеры, в качестве <code>seed</code> нужно передать <code>response.seeds.next</code> из ответа на предыдущий запрос.
Ответ	<pre>{ // флаг успешности (при ошибке false): "success": true, // Тело ответа на запрос (при ошибке содержит подробности // ошибки) : "response": { // Список камер: "cameras": [{ // Идентификатор сервера: "server": "...", // Идентификатор камеры: "camera": "...", // Название камеры: "camera_name": "Cool Public Camera", "misc": { // Описание камеры: "description": "Yeah, Ivideon is cool!", // ... }, // Общее количество просмотров: "total_views": 2144, // Оригинальная ширина изображения с камеры: "width": 1280,</pre>

	<pre> // Оригинальная высота изображения с камеры: "height": 720, // ... }, // ...], // Идентификаторы страниц для постраничной навигации: "seeds": { // Идентификатор текущей страницы: "this": "68935489", // Идентификатор следующей страницы: "next": "68935499" } } } </pre>
--	--

API для получения изображения предварительного просмотра

В представленной ниже документации отражены только те параметры и данные, которые необходимы для выполнения задания.

Запрос	GET https://streaming.ivideon.com/preview/live?server={server_id}&camera={camera_id}
Параметры запроса	server — идентификатор сервера. camera — идентификатор камеры.
Ответ	<i>Изображение в формате JPEG</i>