

Личный проект «Киноман»

Техническое задание

Проект: Киноман

О проекте

Киноман — сервис для фанатов большого кино. Подробная информация о горячих новинках кино, возможность выбрать и сформировать собственный список фильмов к просмотру, обсуждение кинофильмов и многое другое. Киноман поможет провести время интересно.

1. Описание функциональности

Приложение состоит из двух экранов: Фильмы и Статистика .

Обратите внимание, оживление блоков `T a ed` и `M c e ed` часть дополнительного задания. Если вы не планируете его выполнять, удалите эти блоки из разметки в вашем проекте.

1.1 Общий контейнер

В правом верхнем углу шапки отображается звание пользователя. Звание зависит от количества просмотренных фильмов:

0	звание не отображается;
от 1 до 10	се;
от 11 до 20	fa ;
от 21 и выше	e b ff;

В правом углу подвала выводится информации о количестве фильмов в сервисе. Информация обновляется один раз при загрузке приложения.

1.2 Фильмы

После загрузки приложения в списке отображается не более 5 карточек фильмов.

Показ оставшихся фильмов выполняется нажатием на кнопку **S** **e** . При нажатии показываются очередные 5 фильмов или оставшиеся фильмы, если их количество меньше 5.

После показа всех карточек с фильмами, кнопка **С** **е** скрывается.

Любое изменение фильтра или сортировки, а так же переключение на экран статистики и обратно, сбрасывает счётчик показанных фильмов и отсчёт начинается заново.

В случае отсутствия фильмов вместо списка
отображается текст: Т е е а е е da aba е .

1.3 Карточка фильма

Карточки фильмов представлены в двух вариантах: стандартный (на главном экране, в блоках **Т а е д** **е и М с е е d**) и расширенный (попап с описанием фильма).

В стандартном варианте карточка с фильмом содержит следующую информацию:

Постер (картинка);

Название фильма;

Рейтинг;

Год производства;

Продолжительность в формате

(например 1 36);

Жанр;

Краткое описание (не более 140 символов);

Количество комментариев;

Если описание фильма больше 140 символов, то в карточке отображается 139 символов описания и знак многоточие ().

При наведении курсора мыши на блок карточки фильма появляются дополнительные кнопки управления:

Add а с добавляет фильм в список к просмотру;

Added а с ed помечает фильм как просмотренный;

Add fa e добавляет/удаляет фильм в избранное;

Клик по обложке фильма, заголовку, количеству комментариев открывает попап с подробной информацией о фильме;

Попап содержит расширенную информацию о фильме:

Полноразмерная обложка;

Название фильма;

Оригинальное название фильма;

Рейтинг;

Режиссёр;

Сценаристы;

Актёрский состав;

Дата и год релиза в формате

(например: 01 А 1995);

Продолжительность в формате

(например 1 36);

Страна;

Жанр;

Полное описание;

Возрастной рейтинг;

Фильм может относиться к нескольким жанрам. Если фильм относится к нескольким жанрам, выводите Ge e , иначе Ge e .

В попапе отображается блок с кнопками управления:

Add а с добавляет фильм в список к просмотру;

A ead а с ed ,1K

,

Одновременно может быть открыт только один попап. При открытии нового попапа прежний закрывается, например при клике на другую карточку при открытом попапе. Несохранившиеся изменения (неотправленный комментарий) пропадают.

1.4 Комментарии

Список комментариев к фильму и форма добавления нового комментария доступны в попапе.

Каждый комментарий состоит из:

Текст комментария;

Эмоция;

Автор комментария;

Дата комментария;

Кнопка удаления;

Дата комментария отображается в формате / : (например 2019/12/31 23:59).

Для добавления нового комментария пользователь заполняет текст комментария и выбирает эмоцию (один вариант из: smile, sleeping, puke, angry). Имя автора формируется случайным образом на сервере, с клиента оно не передаётся.

Введённые пользователем данные экранируются.

Отправка формы осуществляется нажатием комбинации клавиш **Ctrl/Command + Enter**.

Пользователь может удалить произвольный комментарий. Комментарий удаляется нажатием на кнопку **Delete**, расположенную в блоке с комментарием.

1.5 Рейтинг

Пользователь никак не может влиять на оценку фильма.

1.6 Фильтры

В приложении предусмотрено несколько фильтров:

- All** — все фильмы;
- Watchlist** — фильмы, добавленные в список к просмотру (**Watchlist**);
- Watched** — просмотренные фильмы (**Watched**);
- Favorites** — фильмы, добавленные в избранное (**Favorites**).

Количество фильмов, соответствующих фильтру отображается справа в элементе с фильтром. Для фильтра **All** количество не отображается.

Информация о количестве фильмов, соответствующих каждому фильтру доступна сразу, без необходимости применения фильтра.

Если фильтру соответствует больше 5 фильмов, действуют правила отображения как для списка фильмов

А е .

1.7 Сортировка

Пользователю доступна возможность сортировки фильмов по дате выхода (клик по ссылке [Sort by date](#)) и рейтингу (клик по ссылке [Sort by rating](#)). Сортировка работает в одном направлении от максимального к минимальному: при сортировке по дате выхода в начале списка будут самые новые фильмы, при сортировке по рейтингу с самым высоким рейтингом.

Для отмены сортировки и возвращению к исходному порядку пользователь кликает по ссылке [Sort by default](#).

1.9 Статистика

Статистика в приложении представлена в виде диаграммы, показывающей количество просмотренных фильмов в разрезе жанров за определённый период.

Помимо диаграммы, в разделе [Статистика](#) отображаются:

- Звание пользователя, если таковое имеется;

- Общее количество просмотренных пользователем фильмов;

- Затраченное время на просмотр всех фильмов;

- Любимый жанр;

Пользователь может сформировать статистику за несколько предопределённых периодов:

- А е;

- Т da ;

Wee ;

M ;

Yea ;

При смене периода диаграмма перерисовывается, а количественные показатели пересчитываются.

Если для статистики недостаточно данных (например, пользователь ничего не успел посмотреть), то для числовых значений статистики выводится 0 , а для текстовых (Любимый жанр) ничего не показывается. Диаграмма также не отображается.

1.10 Взаимодействие с сервером

Сервер расположен по адресу: `https://11.ecmascript.pages.academy/cinemaddict/`;

Все запросы, которые отправляются серверу должны содержать заголовок `Authorization` со значением `Basic ${token}`. Например, `Basic er883jdzbdw`. Случайная строка формируется однократно при старте приложения.

Интерфейс должен реагировать на отправку любого запроса к серверу. Примеры реакции описаны в соответствующих пунктах ТЗ.

При отправке комментария, форма, содержащая текст комментария должна быть заблокирована.

Если запрос на отправку комментария выполнен успешно, то комментарий должен быть добавлен в список комментариев. Форму добавления комментария нужно очистить и разблокировать.

При возникновении ошибки в момент отправки комментария, форма, содержащая текст комментария должна быть разблокирована и к ней применяется эффект

покачивание головой . Стили для эффекта есть в проекте.

Обновление любого элемента в DOM происходят только после успешного выполнения запроса на сервере.

1.10.1 Ресурсы

Структуры данных

Л с а С е :

```
{
  "comment": "a film that changed my life, a true
masterpiece, post-credit scene was just amazing
omg.",
  "date": "2019-05-11T16:12:32.554Z",
  "emotion": "smile"
}
```

С е :

```
{
  "id": "42",
  "author": "Ilya O'Reilly",
  "comment": "a film that changed my life, a true
masterpiece, post-credit scene was just amazing
omg.",
  "date": "2019-05-11T16:12:32.554Z",
  "emotion": "smile"
}
```

С е .e одно из следующих значений:

```
["smile", "sleeping", "puke", "angry"]
```

М е:

```
{
  "id": "0",
  "comments": [
    $Comment.id$, $Comment.id$
  ],
  ,
}
```

```
"film_info": {
  "title": "A Little Pony Without The Carpet",
  "alternative_title": "Laziness Who Sold
Themselves",
  "total_rating": 5.3,
  "poster": "images/posters/blue-blazes.jpg",
  "age_rating": 0,
  "director": "Tom Ford",
  "writers": [
    "Takeshi Kitano"
  ],
  "actors": [
    "Morgan Freeman"
  ],
  "release": {
    "date": "2019-05-11T00:00:00.000Z",
    "release_country": "Finland"
  },
  "runtime": 77,
  "genre": [
    "Comedy"
  ],
  "description": "Oscar-winning film, a war
drama about two young people, from the creators of
timeless classic \"Nu, Pogodi!\" and \"Alice in
Wonderland\", with the best fight scenes since
Bruce Lee."
},
"user_details": {
  "watchlist": false,
  "already_watched": true,
  "watching_date": "2019-04-12T16:12:32.554Z",
  "favorite": false
}
```

```

}
    A      a      E      :
{
  "error": 401,
  "message": "Header Authorization is not correct"
}
    N      F      dE      ,      :
{
  "error": 404,
  "message": "Not found"
}

```

Фильмы /movies

GET /movies

Код ответов

200 OK

401 Unauthorized

Пример:

Request:

URL: GET /movies

Header: Authorization: Basic T9gld2317D

Response:

Status: 200 OK

Body: массив, содержащий элементы типа Movie

Request:

URL: GET /movies/1

Response:

Status: 401 Unauthorized

Body: структура AuthorizationError

PUT /movies

,

Выставление рейтинга, добавление в избранное.

Обратите внимание, изменять можно только пользовательскую информацию. То есть то, что находится внутри поля `user_details`.

Код ответов

200 OK

401 Unauthorized

404 Not found

Пример:

Response:

URL: PUT /api/users/11

Header: Accept: application/json

Response:

Status: 200 OK

Body: Структура вида Movie

Синхронизация с сервером `/movies/sync`

Этот метод потребуется для реализации дополнительного задания лекции 9.

Пример:

Response:

URL: POST /api/movies/sync

Header: Accept: application/json

Body: [{"Movie": "Movie1"}, {"Movie": "Movie2"}]

Response:

Status: 200 OK

Body:

```
{
  "updated": [{"Movie": "Movie1"}, {"Movie": "Movie2"}]
}
```

Комментарии `/comments/:film_id`

GET /comments/: film_id

Получение комментариев к фильму.

Код ответов:

200 OK

401 Unauthorized

404 Not found

Пример:

Request:

URL: GET /comments/11

Headers: Accept: application/json

Response:

Status: 200 OK

Body: Массив, содержащий элементы
вида Comment

POST /comments/: film_id

Создание нового комментария к фильму.

Код ответов:

200 OK

401 Unauthorized

404 Not found

Пример:

Request:

URL: POST /comments/11

Headers: Accept: application/json

Body: Структура вида LocalComment

Response:

Status: 200 OK

, , ,

Body : Структура вида {type: Movie, comment: [Comment, Comment]}

DELETE /comments/: comment_id

Удаление существующего комментария.

Код ответов:

200 OK

401 Unauthorized

404 Not found

Пример:

Request:

URL: DELETE /comments/42

Headers: Accept: application/json

Response:

Status: 200 OK

1.11 Обратная связь интерфейса

На время загрузки вместо карточек фильмов нужно вывести сообщение Loading.

При нажатии на кнопку удаления комментария Delete её заголовок изменяется на Delete, а сама кнопка блокируется. После выполнения запроса к серверу, или возникновению ошибки, заголовок нужно вернуть к изначальному Delete, а кнопку разблокировать.

В момент отправки запроса на создание комментария форма блокируется от внесения изменений. Разблокировка формы происходит после завершения выполнения запроса (неважно, успешно выполнен запрос

или нет).

Если запрос не удалось выполнить (сервер недоступен, произошла ошибка), форма создания остаётся открытой, к ней применяется эффект покачивание головой .

Обновление элементов (удаление комментариев, обновление информации о фильме и так далее) в DOM происходит после успешного выполнения запроса к серверу.

2. Дополнительные задания

На главном экране в блоке `T` `а ed` `е` и `М` `с` `е ed` отображаются по две карточки фильмов. В блоке `T` `а ed` `е` фильмы с наивысшим рейтингом. В блоке `М` `с` `е ed` фильмы с наибольшим количеством комментариев. Если у всех фильмов одинаковый рейтинг или одинаковое количество комментариев, берутся два случайных фильма соответственно.

Блок `T` `а ed` `е` не отображается, если у всех фильмов рейтинг равен нулю.

Блок `М` `с` `е ed` не отображается, если отсутствуют фильмы с комментариями.

Блок `М` `с` `е ed` обновляется во время работы с приложением при добавлении или удалении

пользователем комментариев.

Offline режим. Реализуйте в приложении поддержку оффлайн режима. Приложение должно иметь возможность запускаться без интернета и сохранять пользовательский функционал, кроме работы с комментариями. Она должна блокироваться. При появлении интернета все изменения должны отправляться на сервер.

Реализуйте вывод даты публикации комментария в человеческом формате (например, 12.05.2018, а также 12 мая 2018 года и так далее) с помощью библиотеки `moment.js`.

3. Разное

В зависимости от состояния, некоторым элементам управления применяются соответствующие классы оформления. Например, активный фильтр и т. д. Примеры доступны в директории с вёрсткой (`markup`).