

Практическое задание

Задание 1

Создать SPA (одностраничный сайт с хеш-навигацией) используя AngularJS фреймворк.

Необходимые инструменты: AngularJs, Bootstrap, AngularUI-Bootstrap

Весь бэкенд строится на эмуляровании выполнения запросов и ответов внутри angular, с помощью модуля ngMock сервис \$httpBackend.

Этапы выполнения:

- 1) Верстка страниц со статическим контентом. (5 дней)

Страницы:

- Login – страница входа
- Courses – страница список курсов
- Add\Edit Course – страница добавления/редактирования профиля курса

Максимально использовать стили Bootstrap.

- 2) Настроить роутинг между страницами, используя ngRoute. (3 дня)

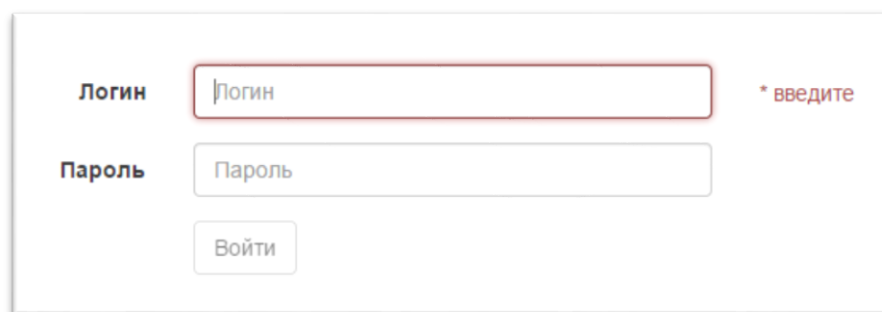
Соответствия:

- Login - # /login
- Courses - #/courses
- Add\Edit Course
 - Add - #/courses/new
 - Edit - #/courses/12 (12 – это id курса)

При вводе не существующего пути, если пользователь вошел в систему перенаправлять его на страницу courses, если не вошел, то на страницу login.

- 3) Реализация страницы Login (3 дня)

Валидация полей по потере фокуса. Поле выделяется красным, и справа выводится надпись «* введите».



The image shows a login form with the following elements:

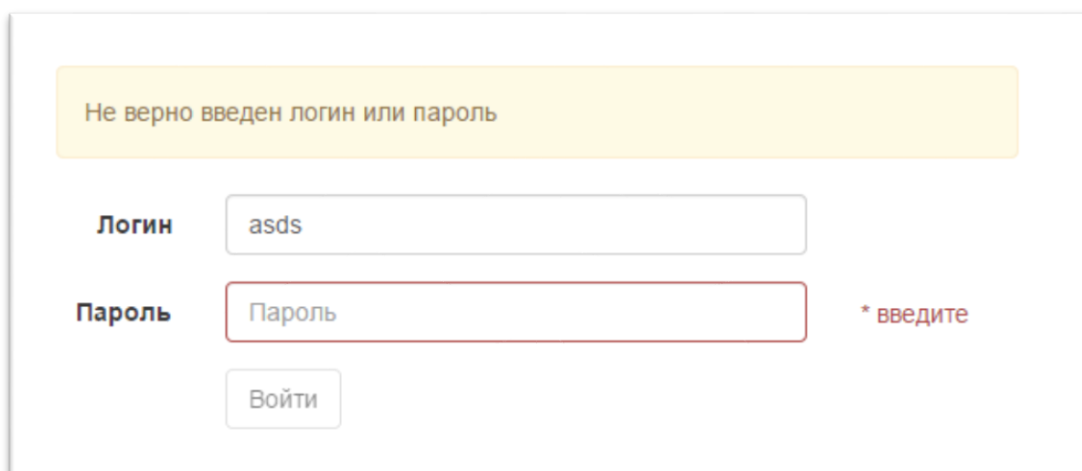
- A label "Логин" (Login) next to a text input field containing the placeholder "Логин".
- A label "Пароль" (Password) next to a text input field containing the placeholder "Пароль".
- A button labeled "Войти" (Login).
- A red asterisk and the text "* введите" (enter) are positioned to the right of the login field, indicating a validation error.

Если хотя бы одно поле не заполнено кнопка «Войти» заблокирована.

Поле «Логин» может содержать, только латинские буквы.

Поле «Пароль» может содержать, латинские буквы и цифры.

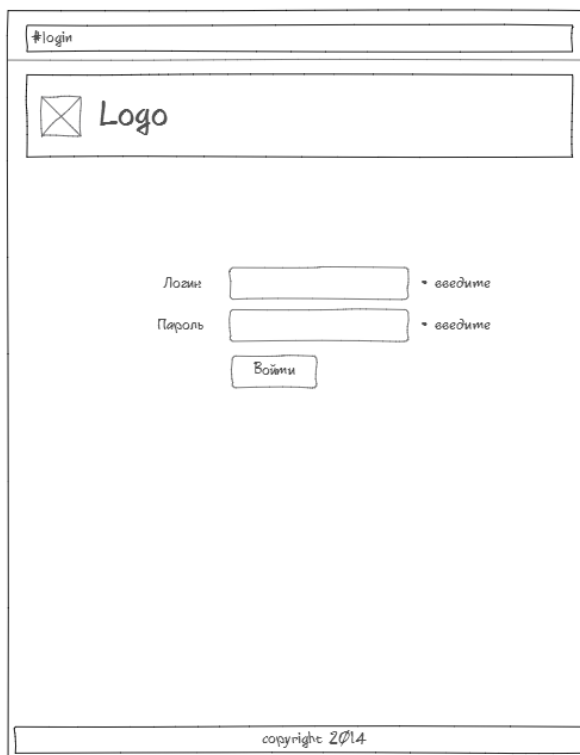
Если пользователь ввел неправильный логин или пароль, то вверху выводить сообщение «Не правильный логин или пароль» и обнулять поле «пароль».



The image shows a login form with a yellow error banner at the top that reads "Не верно введен логин или пароль". Below the banner, there are two input fields: "Логин" (Login) containing the text "asds" and "Пароль" (Password) containing the text "Пароль". The "Пароль" field has a red border and a red asterisk followed by the text "* введите" to its right. Below the password field is a button labeled "Войти" (Login).

После заполнения полей и входа пользователя на сайт, его перенаправляет на страницу courses.

Для авторизации использовать отправку запроса через сервис \$http. В качестве обработки этого запроса настроить перехват с помощью сервиса \$httpBackend.



The image shows a login form within a larger container. At the top is a search bar with the placeholder text "#login". Below it is a logo placeholder consisting of a square with an 'X' and the text "Logo". The login form itself has two input fields: "Логин" (Login) and "Пароль" (Password), both with red asterisks and the text "* введите" to their right. Below the password field is a button labeled "Войти" (Login). At the bottom of the container is a footer bar with the text "copyright 2014".

4) Реализация страницы Courses (6 дней)

Страница для вывода списка всех курсов. Работа с курсами осуществляется по принципу работы с REST. Завести сервис Courses для

получения/создания/сохранения/удаления курсов, **использовать сервис \$resource, необходимо подключение модуля ngResource.** Получение списка курсов или

конкретного курса (get), создание курса(post), обновление полей курса (put), удаление курса (delete), эмулировать через сервис \$httpBackend.

Курсы имеют поля: название, дата создания, длительность (в минутах), список авторов.

В шапке страницы отображать «логин» пользователя, также ссылку «выход». При нажатии на «выход», пользователь выходит и его автоматически перекидывает на страницу Login.

При вводе в поле «поиска» и нажатии на кнопку «Найти». Список курсов фильтруется по названию (создать для этого фильтр).

Создать фильтр для отображения длительности курса в формате «1 час 30 мин».

«Дата» должна выводиться в формате «дд.мм.гггг». (использовать фильтр date).


При нажатии на кнопку «Добавить курс» перенаправлять пользователя на страницу `#/courses/new`.

При нажатии на кнопку «Редактировать» перенаправлять на страницу `#/courses/id` (Id – это id курса).

При нажатии на кнопку «Удалить», вывести окно подтверждения, используя сервис \$modal из angular-ui. Удалить курс при подтверждении.

Под описанием курса вывести список авторов, через запятую.

#courses


Logo

Курсы

user login
[logout](#)

фрагмент имени или дата

найти

☒ Добавить курс

Видеокурс 1
1 час 28 мин

дата

Редактировать

Удалить

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna

Видеокурс 2
15 мин

дата

Редактировать

Удалить

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna

Видеокурс 3
2 часа 15 мин

дата

Редактировать

Удалить

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna

copyright 2014

5) Реализация страницы Course (5 дней)

Страница редактирования и создания курса. Если id курса равен new, то это создание курса. Если пришел id отличный от new, то редактирование курса. Для получения и сохранения изменений использовать сервис Courses (созданный на шаге 4).
Всю работу с курсом эмулировать через сервис \$httpBackend (расширение работы эмуляции п. 4).

В шапке должен появиться breadcrumbs «Курсы->Курс 12». «Курсы» должен быть ссылкой при клике, нас перекидывает на страницу списка курсов.

При изменении названия курса, имя курса также должно меняться в breadcrumbs.

Все поля являются обязательными.

Поле «Дата» сделать директивой допускающей ввод только в формате dd.mm.yyyy.

Т.е. Если пользователь кликает в текстовое поле с датой и начинает ввод «22», то текстовое поле позволит дальше вводить только ".", а после точки только цифру (упростим проверку даты до следующего формата \d\d.\d\d.\d\d\d\d).

«Продолжительность» сделать директивой (которая применяется на текстовое поле и на изменение переводит число в минутах – в часы и минуты отображая их справа). Директива должна использовать тот же фильтр преобразования минут в часы минуты, что и на странице курсов.

Если поля заполнены некорректно – то при нажатии “Сохранить” должно выдаваться сообщение об ошибке (использовать angular-ui сервис \$modal).

Кнопка «Отмена» переходим на страницу курсов, изменения не сохраняются.

The wireframe shows a web interface for editing a course. At the top, there is a breadcrumb navigation bar with the text "#courses/12". Below this is a header section containing a placeholder for a logo (a square with an 'X'), the word "Logo", a button labeled "Курсы", a right-pointing arrow, and the text "Курс 12". On the right side of the header, there is a user status area with "user login" and a blue "logout" link. The main form area contains several input fields: "Название" (Name) with a "Text input" field, "Описание" (Description) with a "Text area" field, "Дата" (Date) with a date field containing "25.12.2014", and "Продолжительность" (Duration) with a field containing "183" followed by the text "3 часа 3 мин". Below these is a section for "Список авторов" (List of authors) featuring two lists of names. The first list contains "Иванов", "Петров", and "Сидоров", with "Петров" highlighted. The second list contains "Пушкин" and "Лермонтов", with "Лермонтов" highlighted. Between the lists are ">" and "<" buttons. At the bottom of the form are two buttons: "Сохранить" (Save) and "Отмена" (Cancel). A footer bar at the very bottom contains the text "copyright 2014".

- 6) Разграничение доступа к страницам. Если пользователь залогинен на сайт ему доступны все страницы. Если нет то перенаправлять его на страницу «Входа»(Login).

Рекомендации:

- Использовать ngView и подменять только центральную часть страниц (header и footer неизменны). Хорошей идеей будет повесить на <body> 1 общий контроллер mainCtrl (который будет управлять шапкой сайта на всех страницах) + для каждой страницы свой собственный контроллер.
- Для хранения логина пользователя можно использовать AngularJS сервис (например, factory с методами login(login,password) сохраняющим эту информацию в localStorage или просто переменную, методом logoff(), и методом getLogin()). Можно посмотреть в сторону angular-local-storage.
- Эмуляция работы бэкенда производить с помощью сервиса \$httpBackend.
- jQuery манипуляция с DOM происходит только в директивах (не в контроллерах!).
- Для вывода сообщений использовать сервис \$modal (angular-ui).
- Выполнять задание следует строго по этапам.

Задание 2

Реализовать jasmine тесты для одного из контроллеров (несколько штук)