

# Прочтите

- <https://rsdn.org/article/inet/jquery.xml> - устарело, но написано хорошо
- <https://webref.ru/dev/jqfundamentals/javascript-basics> - справа в сайдбаре есть меню по разделам
- <https://jquery.com/> - официальный сайт

## Задания

### Задание 0 (это задание показывать преподавателю не надо).

Скачайте проект с <http://assets.codeforces.com/files/695354fbbfd96ca7/de34be3c312ce2dd.7z>. Запустите проект (поправьте profile.properties, пересоберите с помощью mvn package), убедитесь, что всё работает - регистрация+вход+выход. Возможно, понадобится небольшая адаптация так как вы уже выполнили предыдущее задание и ваша база данных чуток другая.

### Задание 1

Переделайте форму входа в систему на использование AJAX.

### Задание 2

Обратите внимание, что использование \$.ajax довольно многословно. Сделайте в файле app.js свою функцию ajax, которая лучше адаптирована к нашему проекту. Добавьте в неё удобные значения по-умолчанию, автоматическое следование (location.href=...) при использовании redirect (то есть теперь в RegisterPage и EnterPage не надо будет проверять "if (response.redirect) ..."). Измените код страниц, чтобы везде пользоваться новой функцией. Убедитесь, то стало меньше кода и он стал чище.

### Задание 3

Создайте сущность Article (статья/пост) с полями id, userId, title, text, createTime (помним про внешний ключ и прочие нужные индексы) и соответствующие ей ArticleRepository, ArticleRepositoryImpl, ArticleService. По ссылке /article добавьте форму создания статьи (два поля: title типа input и text типа textarea, кнопка Create), которая будет работать посредством AJAX. Добавьте ссылку на создание статьи в меню, если пользователь авторизован.

### Задание 4

На главной странице (IndexPage) добавьте вывод всех статей в обратном хронологическом порядке. Используйте AJAX для асинхронной подгрузки статей (используйте элемент template для задания шаблона статьи). Если вы справились с заданием из 2-го дз про разметку статьи, то используйте разметку/css из того задания.

### Задание 5

Добавьте в Article поле/колонок hidden. Отображайте на главной только нескрытые статьи. Сделайте страницу /myArticles с таблицей своих статей (только id, title и колонка с кнопкой Hide/Show). Пусть таблица загружается без AJAX, просто выводится в шаблоне. Третий столбец должен содержать либо кнопку Hide, либо кнопку Show. При нажатии на кнопку должен уходить AJAX-запрос на изменение поля hidden и при успешном выполнении операции текст на кнопке и её действие должно актуализироваться. То есть сама страница должна работать без перезагрузок. Не забудьте проверить в бэкенде, что запрос пришел от автора статьи.

## Задание 6

Добавьте пользователю булевское свойство/колонку `admin` (по умолчанию `false`). Если страницу `/users` просматривает админ, то добавьте в таблицу колонку `Admin`. В этой колонке должен отображаться текущее значение `admin` пользователя и ссылка либо `enable` (то есть включить админа, для текущего `false`), либо `disable` (то есть выключить админа, для текущего `true`). По нажатию на такую ссылку в бэкенд должен уходить AJAX-запрос с `id` пользователя, менять значение `admin` у этого пользователя, а в обработчике на успех надо перерисовать `false` $\leftrightarrow$ `true` и `enable` $\leftrightarrow$ `disable`. Иными словами управление добавить/удалить из админов должно происходить без перезагрузки страницы. Не забудьте проверить в бэкенде, что запрос пришел от админа.