

#### 001 - index страница

Это начальная страница. С неё можно перейти на страницу Essences или Articles, а также Авторизоваться или Зарегистрироваться

#### 002 - Essences Catalog

Essences Catalog - публичная страница вывода всех созданных Сущностей и Свойств к ним. Созданная сущность может совсем не иметь Свойств, или обладать только некоторыми Свойствами

#### 003 - Articles Catalog

Articles Catalog - публичная страница вывода всех созданных Статей

#### 004 & 004-2 - Article

Article - публичная страница отдельной Статьи (плюс возможность добавить Комментарий для авторизованного Пользователя (Комментарий публикуется после прохождения Модерации))

#### 005 - Регистрация

Кроме собственно самой Регистрации, есть возможность выбрать check-box 'Auto login', и при условии, что Валидация вводимых данных пройдена успешно, Пользователь сразу будет авторизован

#### 006 - Авторизация

Кроме Авторизации, можно восстановить Пароль

#### 007 - Account

После Авторизации, Пользователь попадает на некую страницу 'посредник', это некий аккаунт, где Пользователя уведомляют о его Правах, и если аккаунт Пользователя не заблокирован, даётся ссылка на Кабинет (Панель управления)

#### 008 - Cabinet (Панель управления) ~ Essences

Для каждого Пользователя будет своя Панель Управления, в зависимости от его прав. Стартовая страница Панели Управления - это Сущности

#### 009 & 009-2 - Управление конкретной Сущностью

Созданная Сущность имеет Стандартные Свойства, которые есть всегда: это Количество, Описание (Сущности), Изображение. И также, есть возможность создать любое количество Свободных Свойств. Для создания Свободного Свойства нужно указать Имя Свойства, и его Значение.

Стандартные Свойства не удаляются, но очищаются. Свободные Свойства можно удалить. (При загрузке изображения, и далее после его изменения или удаления, старое изображение физически удаляется с сервера. Также при удалении самой Сущности, если в качестве Свойства для неё было добавлено изображение, то оно будет удалено).

#### 010 - Пользователи

Приложение имеет Root-Администратора, который может всё, т.е. кроме всего прочего, что может обычный Пользователь, Root-Администратор (и только он) может управлять Пользователями, а именно удалять Пользователей, назначать Администратора (Администраторов), и блокировать Пользователя, в случае если это потребуется.

Логика построена так, что обычный Пользователь или Администратор не попадут на Страницу управления Пользователями.

Администратор - соответственно не имеет прав управления Пользователями, но имеет права одобрять Комментарии.

Также Root-Администратор и Администратор - могут работать со всеми Сущностями, Статьями, Категориями, Комментариями, включая те, которые создавали не они. А обычным Пользователям доступны только их собственные Сущности, Статьи, Категории, Комментарии, и получить доступ к не своим данным у Пользователя не получится.

#### 011 - Категории

Категории, это некое отношение, которое указывается при создании Статьи (как группа). При создании Статьи может быть выбрано несколько Категорий, но обязательно должна быть выбрана хотя бы одна Категория. Категорию можно удалить, только если она не используется ни в одной Статье.

#### 012 - Статьи

При удалении Статьи, удаляется и Комментарий к Статье (используются Внешние Связи)

#### 013 - Добавление статьи & 014 - Редактирование статьи

Также как и на других прочих страницах приложения, при Добавлении, Редактировании, Удалении, - осуществляется Валидация данных, и выводится соответствующее сообщение.

## 015 - Комментарии

Комментарии может одобрять Root-Администратор и Администратор.

Добавление Комментария - происходит на публичной странице Статьи, и только авторизованным Пользователем.

### Прочее:

- на сколько это было возможно в рамках тестового проекта, была реализована защита от удаления Сущностей для не их авторов, т.е. не авторы не смогут это сделать, кроме соответственно Root-Администратора и Администратора (то же самое реализовано при удалении Свободных Свойств)
- также реализована защита от подмены данных при GET Запросе (подмене данных в адресной строке), при Удалении Сущности, а также Редактировании Стандартных и Свободных Свойств, а также при Удалении Свободных Свойств.
- реализован вывод сообщений успешных / не успешных действий - практически на все действия совершаемые Пользователем, соответственно реализованы соответствующие проверки (Валидация данных происходит как в отдельных Request-ax, так и в Контроллере).
- предусмотрены поля в БД как: `user\_id` и `first\_author`, где значение созданное впервые - не изменяется в Поле `first\_author`, а при Редактировании (например Администратором) - Поле `user\_id` изменит своё значение
- закрыт доступ к данным, для не их владельцев, или если у Пользователя нет на это прав, кроме Root-Администратора, и Администратора (не учитывая работу с Пользователями).
- для Сущностей, Свойств, Категорий, Статей, Комментариев - реализованы действия:
  - Добавить /
  - Редактировать /
  - Удалить (или Очистить для Стандартных Свойств Сущности,
  - Загрузить изображение для Стандартного Свойства Сущности,
  - и различные переходы.
- валидация данных - реализована как в отдельных Request, так и в Контроллерах - для наглядности подходов. Также, работа с данными - где-то вынесена конкретно в Модель (в результате мы получаем 'тонкий' Контроллер), а где-то описана в Контроллере (опять же для наглядности). Например, в ArticlesController - Валидация вынесена в отдельный Request - ArticleRequest, а работа с данными в Модель Article.
- активно используются Внешние Ключи, и при данной взаимосвязи, при удалении - взаимосвязанные данные удаляются каскадом (смотрите схему БД, или Миграции).