

Описание потребностей

Имплементировать веб приложение для управлением задач пользователя.

Требования взяты из реальных проектов.

Требования к реализации:

- Сделать главную страницу приложения, на которой выводится список всех задач в системе с указанием идентификатора задачи, времени создания, названия и пользователя, на которого назначена задача.
- Сделать страницу авторизации пользователя в систему Task Manager.
- После авторизации пользователь попадает на страницу списка своих задач (личный кабинет). В этом списке выводятся задачи со следующими атрибутами: идентификатор задачи, название, описание, статус задачи, время создания.
- Сделать возможность добавления, редактирования и удаления задач пользователем из списка задач в личном кабинете.
- Если пользователь - admin, то в списке задач в личном кабинете он видит все задачи в системе. При этом в таблице списка задач указывается ещё пользователь, на которого назначена задача. Он может назначать, редактировать и удалять задачи других пользователей.
- Реализовать прикрепления файла к задаче. Использовать гем carrierwave.
- Сделать страницу просмотра задачи в личном кабинете. На странице выводится информация о задаче: идентификатор задачи, название, описание задачи, время создания. Если к задаче была прикреплена картинка, то отобразить картинку, иначе отобразить ссылку на скачивание файла. Если пользователь - admin, то выводить пользователя, на которого назначена задача.
- Реализовать смену состояния задачи отдельными переключателями в списке задач.

Требования к бизнес-логике:

- Модель User. Атрибуты - email (уникальное поле), password, role (роль пользователя - admin, user, можно отдельным полем(атрибутом)).
- Модель Task. Атрибуты - name, description, user, state. Возможные значения state - new, started, finished. Задача не может существовать без имени.

Реализовать связь one-to-many между User и Task. Обеспечить целостность данных при удалении объектов (задача не может существовать без пользователя).

Для реализации state machine использовать гем state_machines-activerecord или aasm.

Требования к контроллерам

- Реализовать иерархию контроллеров [3]
- Не использовать scaffold.
- Для авторизации не использовать гем devise.

Требования к вьюшкам

- Использовать гем simple_form для реализации форм.
- Шаблонизатор - haml/slim.
- Twitter Bootstrap 3 для вёрстки.

Требования к тестированию

- Тестовое покрытие кода >92 %. Проверка через гем simplecov.
- Rspec

Требования к развёртыванию

- Код разместить на github.
- Прикрутить travis ci.
- При наличие собственного внешнего сервера выложить приложение на сервер и прикрутить capistrano для деплоя. При отсутствии - выложить на Heroku.
- Написать rake-таск для создания фейковых данных. Использовать гем faker.
- Создать учётку admin и user и поместить в seeds.

Оценка задания

При оценке тестового задания будут учитываться следующие пункты:

- Выполнение требований к заданию.

- Следование style guides.
- Следование требованиям к тестированию, полнота тестов.
- DRY.

Справочная информация

1. <https://github.com/arbox/ruby-style-guide/blob/master/README-ruRU.md>
2. <https://github.com/arbox/rails-style-guide/blob/master/README-ruRU.md>
3. <https://habrahabr.ru/post/136461/>
4. <http://rusrails.ru/>