

[КАК СТАТЬ АВТОРОМ](#)[Поехали в гик-трип по Календарю IT-ивентов?](#)

gust_of_wind

3 часа назад



Тестовое задание по глубокому копированию сделок в amoCRM или Composition over inheritance. Часть 1

Средний

Разработка веб-сайтов*, CRM-системы*, PHP*

Ожидает модерации

Дано

Fullstack PHP developer (я), нет опыта с amoCRM, и тестовое задание на позицию backend-разработчика из двух задач.

1. Перенести сделки, удовлетворяющие условиям фильтра, на заданный этап воронки продаж.
2. Скопировать сделки, удовлетворяющие условиям другого фильтра, на следующий этап воронки, создав при этом копии примечаний и задач, присоединённых к сделкам.

Под фильтрами здесь понимается зашитое в код условие на бюджет сделки, воронку продаж и этап. Let's do it!!

(Вам нравятся длинные предложения в русском языке? Русский язык он такой: с суффиксами и окончаниями! :-))

Кратко об amoCRM

Т.к. опыта с amoCRM у меня никакого нет, вначале разберёмся, как тут всё устроено, и как всё будет работать в общих чертах. Итак, amoCRM - это SaaS решение, которое автоматизирует работу отдела продаж компании и all the stuff around it, например, звонки. Ключевое понятие (сущность) вокруг которого строится работа - это заявки клиента. Заявки поступают в воронку

продаж, их перемещают с этапа на этап, к заявкам добавляются контакты клиента, создаются примечания, задачи и тому подобное.

Также в приложении amoCRM есть экран (dashboard) со всевозможной аналитикой по заявкам. Пока отметим для себя раздел "События" в этом dashboard. Очень интересная вещь.

Авторизация по протоколу OAuth2

Если вы когда-нибудь авторизовались на сайте через кнопку соцсетей, то вы, в принципе, поймёте, о чём идёт речь далее.

(А как вам три слова подряд с буквой "ё"? Do you like it?)

Чтобы на своём сайте, который мы создадим, иметь возможность перемещать заявки в нашем аккаунте amoCRM, надо пройти подобную авторизацию. Речь идёт о протоколе OAuth2, суть которого в том, что мы меняем авторизационные данные (идентификатор клиента, секретный ключ и код авторизации) на не менее секретный токен (временный "пароль"). Затем мы используем полученный токен при каждом обращении к API amoCRM. Так система будет узнавать, что пользователь - это именно тот пользователь, за которого он себя выдаёт. (Ещё системе авторизации amoCRM есть понятие refresh token, но тестовое задание можно сделать и без разбирательств с ним. Не будем заморачиваться с деталями!) Компания Microsoft (и Github) пошли дальше: к выданому токenu можно задавать допустимые действия пользователя в системе, например: работу с репозиториями, workflow и т.д.

Технически авторизация с использованием OAuth2 делается довольно просто. Идём в раздел amoMarket web-приложения, создаём т.н. Внешнюю интеграцию с сайтом и прописываем URL скрипта нашего сайта для редиректа (бэкенд amoCRM отдаст заголовок Location с этим URL). Т.к. переадресация будет происходить локально в браузере, мы вполне можем использовать адрес вида http://localhost/my_script_to_redirect.php, ну, или <http://localhost/redirect>, если у вас роутинг в приложении. :-)

На стороне клиента (наш сайт) поступим максимально просто. Отдадим скрипту код авторизации, который УЖЕ доступен в web-интерфейсе amoCRM (amoMarket -> Установленные -> (Ваша интеграция) -> Ключи и доступы). После этого обменяем код авторизации на токен доступа, сохраним его в файл и будем заниматься более интересными вещами.

(Строго говоря, при таком сценарии редирект с бэкенда amoCRM даже не зовётся, т.е. технически он не нужен. Но при создании класса AmoCRMApiClient для работы с API мы этот адрес передаём в конструктор. Так что пусть будет. Это не предмет нашего исследования at the moment :-))

Инструменты для разработки

В силу отсутствия у меня mini PC с Linux, а также некомфортностью работы со связкой Vagrant, Laravel Homestead и docker (Опять этот маппинг директорий настраивать... SSH authorization... Подсети... Где оно всё там находится?! OMG!), для этого тестового задания воспользуемся ОС Windows. В качестве редактора кода с лихвою хватит уютненького и лёгкого Sublime Text. Для web-сервера и "публикации" нашего приложения будем использовать Laragon, а также такую фичу файловой системы NTFS как symbolic links. Они очень удобны: можно "на лету" менять параметр DocumentRoot в конфиге Apache.

Папки проекта

Всё относящееся к тестовому заданию будет храниться в директории с незамысловатым именем api-test. Внутри создадим папки с self-explained названиями app и project-management. В папке app создадим каталог webroot, на который будет указывать symbolic link из Apache. А также директории src и config, в которой будут храниться (sensitive) настройки приложения и токен доступа. (По-хорошему, последний файл надо хранить в каком-нибудь хранилище секретов, зашифрованным или хотя бы в другой папке. Но не в этот раз! ;-))

А пока создадим в папке webroot файл index.php, в папке config файл .env.sensitive.data и директорию token-storage для хранения токена доступа к

бэкенд amoCRM. Вот что получилось:

```
habr>tree /F
Folder PATH listing
Volume serial number is B573-1E87
I:..
|---config
|   |---.env.sensitive.data
|   |---token-storage
|---src
|---webroot
|       index.php
habr>
```

Дерево проекта в файловой системе

Теги: php, oauth2

Хабы: Разработка веб-сайтов, CRM-системы, PHP

Данная статья не подлежит комментированию, поскольку её автор ещё не является [полноправным участником сообщества](#). Вы сможете связаться с автором только после того, как он получит [приглашение](#) от кого-либо из участников сообщества. До этого момента его username будет скрыт псевдонимом.

О ПЕСОЧНИЦЕ

Это [«Песочница»](#) — раздел, в который попадают дебютные публикации пользователей, желающих стать полноправными участниками сообщества.

Если у вас есть [приглашение](#), отправьте его автору понравившейся публикации — тогда её смогут прочитать и обсудить все остальные пользователи Хабра.

Чтобы исключить предвзятость при оценке, все публикации анонимны, псевдонимы показываются случайным образом.

О МОДЕРАЦИИ

Не надо пропускать:

- рекламные и PR-публикации
- вопросы и просьбы (для них есть [Хабр Q&A](#));
- вакансии (используйте [Хабр Карьеру](#))
- статьи, ранее опубликованные на других сайтах;
- статьи без правильно расставленных знаков препинания, со смайликами, с обилием восклицательных знаков, неоправданным выделением слов и предложений и другим неуместным форматированием текста;
- жалобы на компании и предоставляемые услуги;
- низкокачественные переводы;
- куски программного кода без пояснений;
- односложные статьи;
- статьи, слабо относящиеся к IT-тематике или не относящиеся к ней вовсе.

Ваш аккаунт	Разделы	Информация	Услуги
Профиль	Статьи	Устройство сайта	Корпоративный
Трекер	Новости	Для авторов	блог
Диалоги	Хабы	Для компаний	Медийная реклама
Настройки	Компании	Документы	Нативные проекты
ППА	Авторы	Соглашение	Образовательные
	Песочница	Конфиденциальность	программы
			Стартапам
			Спецпроекты



Настройка языка

[Техническая поддержка](#)

[Вернуться на старую версию](#)

© 2006–2023, Habr