

Test case for Back-end Developer

Требования:

1. Реализовать задание на любом php framework'e, строго придерживаясь rich domain model
2. Использовать MySQL.
3. Можно использовать js библиотеки для таблицы.
4. Обязательное применение unit and functional tests

Задание:

Требуется сделать простейший обработчик клика.

1. Создать таблицу "click" и "bad_domains".

Структура "click": id | ua | ip | ref | param1 | param2 | error | bad_domain

Структура "bad_domains": id | name

2. Реализовать сценарий для обработки клика по трекинг ссылке.

Пример трекинг ссылки: http://local.dev/click/?param1=some_value¶m2=some_value

- 2.1. Во время перехода по трекинг ссылке записывать user-agent, ip, referrer + param1, param2 в базу с генерацией уникального ID. param2 не учитывается в связке данных уникального клика.
- 2.2. Если связки данных уникального клика нет в базе - выводим данные клика в произвольной форме редиректим на http://local.dev/succes/ID_CLICK.
- 2.3. Если клик имеет существующую связку данных в базе - увеличиваем счетчик "error" и делаем редирект на http://local.dev/error/ID_CLICK, где выводим сообщение об ошибке в произвольной форме.
3. На <http://local.dev/> вывести список всех залогированных кликов таблицей. Написать на JS сортировку и поиск.
4. Сделать вывод и добавление плохих рефферров в "bad_domains".
- 4.1. Добавить дополнительную проверку реффера по списку доменов в таблице "bad_domains". Если найден домен, то повторяются действия пункта 2.3 и значение "bad_domain" становится равно 1 и после редиректа на http://local.dev/error/ID_CLICK с выводом ошибки, сделать через 5 секунд редирект на <http://google.com>.

Примечания:

1. <http://local.dev/> пример домена.
2. Для генерации уникального айди клика не использовать автоинкремент
3. Не использовать ActiveRecord.
4. ID_CLICK нужно подменить на сгенерированный айди клика.