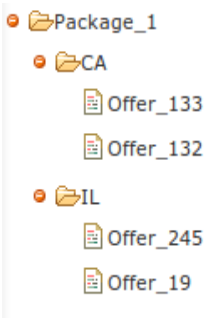
Applicant Qualification Test

For position: Web Team Developer

General Background:

The company has decided to build a new configuration system and you need to implement the crucial parts of it, the configuration is made out of packages and offers. A user is downloading a program and during the installation process, the servers need to send back a configuration to the installer made from the following parts:

A package is a group entity, which has multiple offers per country:



The package entity has the following attributes:

Name	Description	Example
`Id`	Numerical identifier of the package	5678
`name`	The name of the package, will be shown on the installer on the end user machine	"awesome package"
`created_time`	When the package was created	"2017-07-06 12:22:45"
`updated_time`	When one of the package parameters was last updated	"2017-07-07 10:35:19"

The offer entity has the following attributes:

Name	Description	Example
`Id`	Numerical identifier of the offer	1234
`name`	The name of the offer, will be shown on the installer on the end user machine	"Super Magic File Manager"
`created_time`	When the offer was created	"2017-07-06 12:22:45"
`updated_time`	When one of the offer parameters was last updated	"2017-07-07 10:35:19"
`is_enabled`	Indicates if the offer should be shown	"0"
`installs_limit_per_day`	Amount of installation allowed for the offer, if exceeded, the offer should not be shown	"4053"
`content`	HTML, CSS and JS for the actual offer	"<html> ….. </html>"

Every hour the system should create the package → country → offer mapping according to the following logic:

- For each package, for each country, create a list of the top 10 offers using the `daily_installs` table
    - Offer should be rated according to the daily installs per views (since there might not be daily info for a specific offer/country combo, use the latest known data (could be half a year ago))
    - There could be less than 10 offers per country, in this case return all of them (there could be none).
- Save the selected offers in the `package_offer` table.
- The system should create a file for each package, per country, containing the package data and all of the relevant offers and named `{PACKAGE_ID}_{COUNTRY_CODE}` alongside with the country, see example .

For example, let's assume that we have a package called "Space Monkey Pirates" with id 1843.

The system has created a list of offers for the package, for Israel it chose offers 1, 14, 15 and 40

The system should generate a json file named `1843_IL` containing the following:


```
{
    name: "Space Monkey Pirates",
    is_enabled: 1,
    id: 1843,
    country: "IL",
    file_created_at: "2017-07-26 07:56:54",
  - offers: [
      - {
            name: "Mayert Ltd",
            is_enabled: 1,
            content: "...",
            created_time: "2017-02-07 00:20:38",
            installs_limit_per_day: 8028,
            id: 13
        },
      - {
            name: "Pacocha, Kiehn and Bogan",
            is_enabled: 1,
            content: "...",
            created_time: "2016-11-16 19:42:00",
            installs_limit_per_day: 3015,
            id: 21
        },
      - {
            name: "Wehner, Schaefer and Beatty",
            is_enabled: 1,
            content: "...",
            created_time: "2016-11-22 19:43:50",
            installs_limit_per_day: 9904,
            id: 55
        },
      - {
            name: "Walter, Medhurst and Crist",
            is_enabled: 1,
            content: "...",
            created_time: "2017-07-07 00:00:23",
            installs_limit_per_day: 4048,
            id: 93
        }
    ]
}
```

Requirements:

- You are required to create to following parts of the system:
  - A page for adding, changing and deleting offers.
  - A page for adding, changing and deleting packages.
  - A page for viewing offers per package, allowing the user to select a package name and country combination.
  - A script which creates offer mapping for each package (implement it as a page in the system)
  - A script that generates json files per package, per country (implement it as a page in the system)
- You are required to use PHP, and no frameworks are allowed, you are allowed to use modules you find necessary.
- You may use any front-end framework to your liking.
- Please include a readme file, containing instructions how to run your code (i.e composer usage, SQL scripts if needed etc.)
- The system doesn't have to be a single page application or use fancy rewrites, focus on the main requirements.
- You are also provided the SQL schema for the system, alongside with demo data (attached SQL file)

Thank you for taking the time to take this test, please read the instructions carefully before starting.

Make sure you understand it all, if not, don't hesitate to contact us, it's better to ask weird questions than answering incorrectly.

You are free to implement the tasks anyway you find fit, however no usage of frameworks is allowed in order to measure all candidates equally.

After completing the test, we'll have a talk about the task and the implementation.

Good luck and see you soon ☺