ДОМАШКА 11 & 12

Завдання

Задача. OOP. MVC. “Вивести список із ТОП-500 популярних доменів.“

Необхідно реалізувати набір **PHP**-скриптів, який отримує список доменів та їх **rank** (популярність) із `txt` та `json`-файлів, перевіряє чи домени доступні по протоколу **http** та виводить ТОП-**500** найпопулярніших веб-сайтів, відсортованих по рейтингу у порядку спадання.

Під час реалізації необхідно **повною мірою використовувати** ООП-підхід: *класи, інтерфейси, наслідування, композицію класів*.

Проект необхідно реалізувати згідно патерна проектування **MVC** (`Model View Controller`).

Вхідні дані.

На диску міститься два файли однакової структури, але різним форматуванням і розширенням:

- **`top10milliondomains.csv`**
- **`top10milliondomains.txt`**

Кожен з файлів містить список "Топ-100 мільйонів найпопулярніших веб-сайтів" та Open Page Rank (популярність) для кожного із доменів.

Приклад стрічки із файла **`top10milliondomains.txt`**:

```
"150";"s-media-cache-ak0.pinimg.com";"7.45"
```

Необхідно написати окремі "класи-рідери" (`readers`), **для кожного** із розширень файлів (**`pdf`**, **`txt`**) які читають інформацію із файла та повертають її у вигляді масива. Формат масива, який повертається класами рідерами довільний.

Кожен клас-рідер необхідно реалізувати у вигляді класу та помістити в окремий **`php`**-файл. Також необхідно розробити **`Interface`** який стандартизує поведінку класів-рідерів, для того

щоб передбачити додавання нових типів вхідних файлів, наприклад: **xml**, **html**, **json** у майбутньому.

Із цього списка необхідно вибрати лише два поля: `domain`, `rank`, номер можна пропустити, ця інформація ніде не використовується.

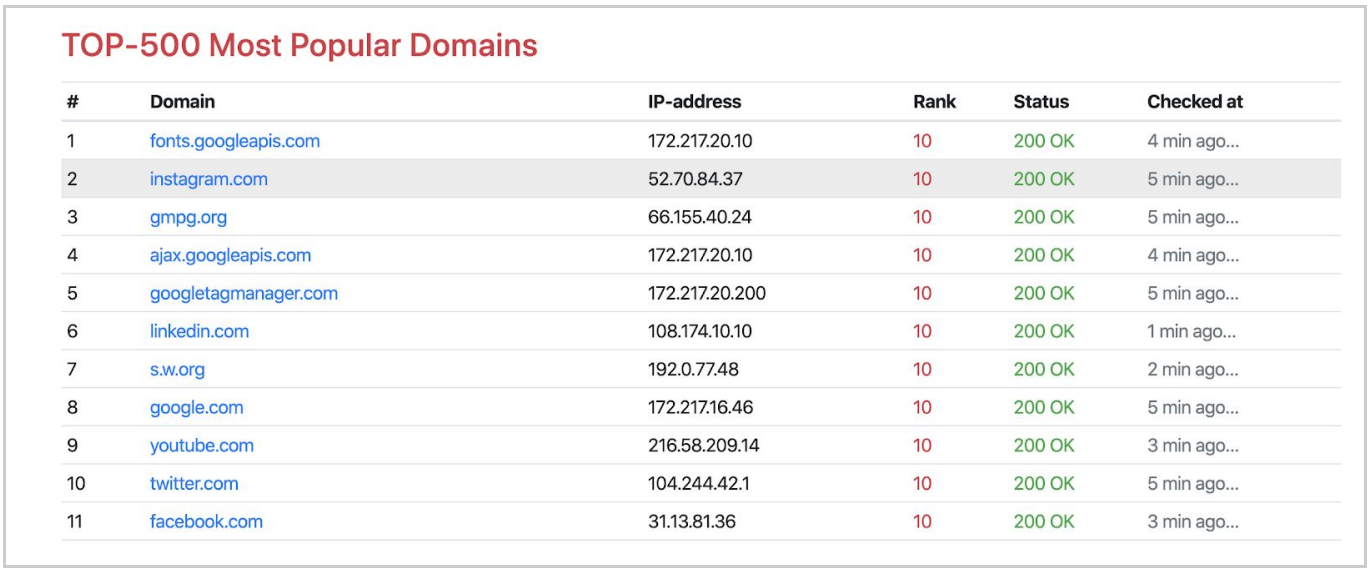
Відображення інформації

Після того як дані із **усіх файлів** прочитано, необхідно відобразити ТОП-**500** доменів відсортованих в порядку спадання по полю **rank**, та вивести їх в табличному вигляді на `html`-сторінці.

Перед виводом даних в таблицю необхідно унікалізувати масив із списком доменів (забрати повторення/дублі цих доменів).

Також, крім полів **domain** та **rank** у таблиці необхідно виводити **ip**-адресу цього домену. Користуйтеся вбудованими функціями **PHP** для того щоб конвертувати домен в ip-адресу і навпаки.

*Значення атрибутів **Src** та **Last checked** на цьому етапі залишаємо пустими. Їх реалізація буде винесена в окреме завдання.*



TOP-500 Most Popular Domains

#	Domain	IP-address	Rank	Status	Checked at
1	fonts.googleapis.com	172.217.20.10	10	200 OK	4 min ago...
2	instagram.com	52.70.84.37	10	200 OK	5 min ago...
3	gmpg.org	66.155.40.24	10	200 OK	5 min ago...
4	ajax.googleapis.com	172.217.20.10	10	200 OK	4 min ago...
5	googletagmanager.com	172.217.20.200	10	200 OK	5 min ago...
6	linkedin.com	108.174.10.10	10	200 OK	1 min ago...
7	s.w.org	192.0.77.48	10	200 OK	2 min ago...
8	google.com	172.217.16.46	10	200 OK	5 min ago...
9	youtube.com	216.58.209.14	10	200 OK	3 min ago...
10	twitter.com	104.244.42.1	10	200 OK	5 min ago...
11	facebook.com	31.13.81.36	10	200 OK	3 min ago...

Задача 2. cURL. Перевірка доступності ТОП-500 доменів.

Необхідно вдосконалити код програми із попереднього завдання додавши **namespaces** (неймспейси) до класів, а також реалізувати імпорт цих **namespaces** в коді-клієнті (код клієнт - це код який використовує класи).

Якщо на цьому етапі реалізації проекту є дублювання коду, винесіть його в окрему структуру, наприклад батьківський клас чи трейт. (Стаття "Що таке трейти").

Під час написання програмного коду (і розробки ПО) рекомендується слідувати наступним принципам: **DRY** "**Don't repeat yourself**" та **KISS**.

Також необхідно перевірити "доступність" кожного домену із **ТОП-500**, по протоколу **HTTP** і відобразити результати в таблиці, в полі **status**. Час перевірки у форматі `x min ago…` відображається в останньому полі .

TOP-500 Most Popular Domains

#	Domain	IP-address	Rank	Status	Checked at
1	fonts.googleapis.com	172.217.20.10	10	200 OK	4 min ago...
2	instagram.com	52.70.84.37	10	200 OK	5 min ago...

Для реалізації перевірки доступності домена реалізуйте окремий клас `HttpClient`, який використовує бібліотеку curl, а точніше її "**php-обгортку**" (wrapper). Приклад використання **`curl`** можна знайти в офіційній документації.

Під час реалізації цього завдання слід пам'ятати, що робота ведеться із протоколом **HTTP**, який працює в режимі запит (**HTTP Request**) та відповідь (**HTTP Response**), і по мережі передаються не лише самі дані (data, content) а й системна інформація **HTTP**-протоколу у вигляді **HTTP**-заголовків. (Див скріншот `Chrome Developer Tools`, нижче). В цьому завданні необхідно з допомогою бібліотеки curl здійснити **HTTP Request** на кожен із ТОП-500 доменів і перевірити значення **`Status Code`** (див скріншот нижче), який повертається у заголовках відповіді **HTTP**

Response, якщо він рівний `200 OK` - сайт доступний. Повний список **`Status`**-кодів та їх значення можна знайти за посиланнями російською та англійською. Слід пам'ятати про редіректи (`Status Code 301`) і що їх може бути декілька, перш ніж сервер "віддасть" відповідь `200 OK`. Для обходу цієї ситуації користуйтеся вбудованими опціями бібліотеки curl, повний перелік яких можна знайти на сторінці офіційної документації.