CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

Devoir 1

Exercice 1 : Le sort du dernier recours

Crawler

Le crawler fonctionne avec d20pfsrd.com. Il s'agit d'une implémentation de Scrapy, un module python dédié aux crawlers. L'araignée (le module crawler) parcours le contenu d'article de la page et ne sort que les liens correspondant à un DOM précis qui pointe vers la première colonne du tableau de sorts.

La bestiole renvoie ensuite le contenu html du « article-content » de chaque page dont le lien à été attrapé et nous le digérons à grand renfort d'expressions régulières. Chaque champ trouvé est envoyé sous la forme d'un objet à variables de type inconnu, ce qui permet de renvoyer des arrays en cas de besoin. Nous récupérons donc :

- Le nom du sort
- Une array key-value représentant les niveaux du sort selon les classes
- Une array contenant les composantes du sort (« V », « M », « S » et/ou « DF »)
- Un champ de texte avec la résistance magique (généralement « yes » ou « no » mais aussi les variantes)

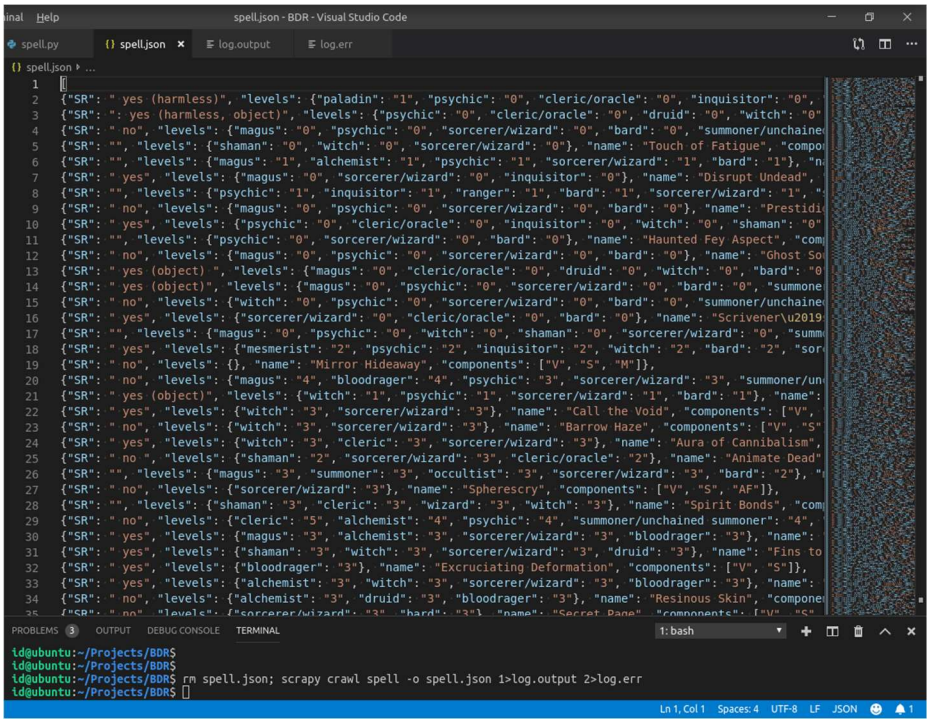

Figure 1 : json de sortie

En raison de l'absence de ces paramètres sur 33 sorts, ceux-ci seront exclus du set par les exceptions du code. 6 autres exceptions seront à exclure des résultats suivants car leurs composants résistent aux sucs digestifs de l'araignée.

Ward Shield ; Armor Lock ; Carry Companion ; Emblazon Crest ; Clarion Call ; Keep Watch.

CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

MongoDB

Après avoir DL MongoDB Compass, nous avons insérés en Json la sortie du crawler. Pour ce faire un petit formatage s'impose. Enlever les virgules à la fin de chaque ligne puis retirer les crochets de début et de fin de fichier. Et hop ! Nous voilà avec une collection rempli de nos sorts sur notre serveur MongoDB .

Ensuite nous avons utilisés le langage de query de MongoDB Compass qui se rapproche des langages query de Splunk ou d'autres agrégateurs. Avec la requête suivante (ici formatté en python) :

```
{
    '$and': [
        {
            '$or': [
                {
                    'levels.sorcerer/wizard': '0'
                }, {
                    'levels.sorcerer/wizard': '1'
                }, {
                    'levels.sorcerer/wizard': '2'
                }, {
                    'levels.sorcerer/wizard': '3'
                }, {
                    'levels.sorcerer/wizard': '4'
                }, {
                    'levels.sorcerer/ wizard': '0'
                }, {
                    'levels.sorcerer/ wizard': '1'
                }, {
                    'levels.sorcerer/ wizard': '2'
                }, {
                    'levels.sorcerer/ wizard': '3'
                }, {
                    'levels.sorcerer/ wizard': '4'
                }
            ]
        }, {
            '$and': [
                {
                    'components': {
                        '$ne': 'M'
                    }
                }, {
                    'components': {
                        '$ne': 'S'
                    }
                }
            ]
        }
    ]
}
```

Nous avons donc pu avoir un résultat égal à 29 sorts différents moins certains qui n'ont pas été bien crawlés auparavant.

CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

bdr

	name String
1	"Desperate Weapon"
2	"Ventriloquism"
3	"Fool's Teleport"
4	"Geas/Quest"
5	"Dimension Door"
6	"Anywhere But Here"
7	"Ward Shield"
8	"Armor Lock"
9	"Storm Step"
10	"Raven's Flight"
11	"Carry Companion"
12	"Buoyancy"
13	"Blindness-Deafness"
14	"Silent Table"
15	"Blur"
16	"Anti-Summoning Shield"
17	"Sundering Shards"
18	"Liberating Command"
19	"Feather Fall"
20	"Emblazon Crest"

CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

SparkDataFrame SQL

Après import des données depuis le json dans un dataframe, il nous a suffit d'exécuter la requête suivante pour obtenir les mêmes résultats que précédemment. (Le calcul prenant tout de même plus de temps.)

```
SELECT `name`, `components` FROM test WHERE
((test.levels.`sorcerer/wizard` = '0' OR
test.levels.`sorcerer/wizard` = '3'
OR test.levels.`sorcerer/wizard` = '1'
OR test.levels.`sorcerer/wizard` = '2'
OR test.levels.`sorcerer/wizard` = '4')
OR
(test.levels.`sorcerer / wizard` = '0'
OR test.levels.`sorcerer / wizard` = '1'
OR test.levels.`sorcerer / wizard` = '2'
OR test.levels.`sorcerer / wizard` = '3'
OR test.levels.`sorcerer / wizard` = '4')
OR
(test.levels.`sorcerer/  wizard` = '0'
OR test.levels.`sorcerer/  wizard` = '1'
OR test.levels.`sorcerer/  wizard` = '2'
OR test.levels.`sorcerer/  wizard` = '3'
OR test.levels.`sorcerer/  wizard` = '4')
OR
(test.levels.`sorcerer/ wizard` = '0'
OR test.levels.`sorcerer/ wizard` = '1'
OR test.levels.`sorcerer/ wizard` = '2'
OR test.levels.`sorcerer/ wizard` = '3'
OR test.levels.`sorcerer/ wizard` = '4'))
AND        NOT        (array_contains(test.components,        "S")        OR
array_contains(test.components, "M"))
```

Il nous faut utiliser cette longue suite de variations pour prendre toutes les possibilités de mise en forme qui existent sur le site d20pfsrd qui n'est pas parfaitement normalisé. On obtient donc les 29 sorts après réduction, comme dans les screenshots ci-dessous. On a toujours le problème des sorts qui ont mal été crawlés.

CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

```
root
 |-- SR: string (nullable = true)
 |-- _corrupt_record: string (nullable = true)
 |-- components: array (nullable = true)
 |    |-- element: string (containsNull = true)
 |-- levels: struct (nullable = true)
 |    |-- adept: string (nullable = true)
 |    |-- alchemist: string (nullable = true)
 |    |-- antipaladin: string (nullable = true)
 |    |-- arcanist: string (nullable = true)
 |    |-- artifice: string (nullable = true)
 |    |-- bard: string (nullable = true)
 |    |-- bloodrager: string (nullable = true)
 |    |-- cleric: string (nullable = true)
 |    |-- cleric/oracle: string (nullable = true)
 |    |-- crimson assassin: string (nullable = true)
 |    |-- darkness: string (nullable = true)
 |    |-- druid: string (nullable = true)
 |    |-- inquisitor: string (nullable = true)
 |    |-- investigator: string (nullable = true)
 |    |-- magus: string (nullable = true)
 |    |-- medium: string (nullable = true)
 |    |-- mesmerist: string (nullable = true)
 |    |-- no two of which can be more than: string (nullable = true)
 |    |-- occultist: string (nullable = true)
 |    |-- oracle: string (nullable = true)
 |    |-- paladin: string (nullable = true)
 |    |-- psychic: string (nullable = true)
 |    |-- ranger: string (nullable = true)
 |    |-- shaman: string (nullable = true)
 |    |-- skald: string (nullable = true)
 |    |-- sorcerer / wizard: string (nullable = true)
 |    |-- sorcerer/  wizard: string (nullable = true)
 |    |-- sorcerer/ wizard: string (nullable = true)
 |    |-- sorcerer/wizard: string (nullable = true)
 |    |-- spiritualist: string (nullable = true)
 |    |-- summoner: string (nullable = true)
 |    |-- summoner/unchained summoner: string (nullable = true)
 |    |-- unchained summoner: string (nullable = true)
 |    |-- warpriest: string (nullable = true)
 |    |-- witch: string (nullable = true)
 |    |-- wizard: string (nullable = true)
 |-- name: string (nullable = true)
```

```
[Desperate Weapon,WrappedArray(V)]
[Ventriloquism,WrappedArray(V, F)]
[Fool's Teleport,WrappedArray(V)]
[Geas/Quest,WrappedArray(V)]
[Dimension Door,WrappedArray(V)]
[Anywhere But Here,WrappedArray(V)]
[Ward Shield,WrappedArray()]
[Armor Lock,WrappedArray()]
[Storm Step,WrappedArray()]
[Raven's Flight,WrappedArray(V)]
[Carry Companion,WrappedArray()]
[Buoyancy,WrappedArray(V)]
[Blindness-Deafness,WrappedArray(V)]
[Silent Table,WrappedArray(V)]
[Blur,WrappedArray(V)]
[Anti-Summoning Shield,WrappedArray(V)]
[Sundering Shards,WrappedArray(V)]
[Liberating Command,WrappedArray(V)]
[Feather Fall,WrappedArray(V)]
[Emblazon Crest,WrappedArray()]
[Touch of Blindness,WrappedArray(V)]
[Clarion Call,WrappedArray()]
[Flare Burst,WrappedArray(V)]
[Keep Watch,WrappedArray()]
[True Strike,WrappedArray(V, F)]
[Mindlink,WrappedArray(V)]
[Wave Shield,WrappedArray(V)]
[Hold Portal,WrappedArray(V)]
[Flare,WrappedArray(V)]

Process finished with exit code 0
```

Conclusion :

Parmi les sorts disponibles à l'utilisation pour notre magicien, trois sortent clairement du lot dans la situation présente. Dimension Door permet de se téléporter sur une courte distance, et ainsi se défaire de ses liens : c'est notre meilleur choix. Raven's flight arrive juste derrière, car il permet de se transformer en corbeau et de voler pendant une courte période de temps, ce qui règle également tous les problèmes. Deux autres choix possibles sont Liberating command, qui permet de donner un check d'escape artist de façon immédiate avec un bonus conséquent mais qui n'est donc pas une méthode certaine, et Storm Step qui transforme l'utilisateur en boule de foudre et le déplace dans une direction. Storm Step nécessite en revanche de faire suffisamment de dégâts à la corde pour la briser, et on ne connait pas son matériau donc ce n'est pas non plus une méthode certaine (la corde pourrait être en adamantium, on ne sait pas… C'est fourbe, un bandit…). Il ne reste plus qu'à espérer que Pito ait préparé un de ces sorts en se levant ce matin.

CHASTAIN Camille
MICHENOT Matthieu MICM30019605
POIGNANT JJ POIJ21119409

Exercice 2 : pagerank

Dans cet exercice, on va simplement utiliser apache spark et ses fonctions pour effectuer le map reduce selon le schéma fourni. L'instruction parallelize nous permet de créer très vite le schéma de notre pagerank, puis il nous suffit d'appliquer la formule mathématique du pagerank.

```
valeurs des contributions pour l'itération 1
(A,1.0)
(B,0.5)
(C,2.5)

valeurs des poids pour l'itération 1
(D,0.15)
(A,1.0)
(B,0.575)
(C,2.275)
--
valeurs des contributions pour l'itération 2
(C,1.225)
(A,2.275)
(B,0.5)

valeurs des poids pour l'itération 2
(D,0.15)
(B,0.575)
(A,2.0837499999999998)
(C,1.19125)
--
```

```
--
valeurs des contributions pour l'itération 19
(A,1.5764851801254862)
(B,0.7451397936871248)
(C,1.6783750261873878)

valeurs des poids pour l'itération 19
(D,0.15)
(B,0.7833688246340561)
(A,1.490012403106663)
(C,1.5766187722592795)
--
valeurs des contributions pour l'itération 20
(B,0.7450062015533315)
(A,1.5766187722592795)
(C,1.6783750261873878)

valeurs des poids pour l'itération 20
(D,0.15)
(B,0.7832552713203318)
(A,1.4901259564203875)
(C,1.5766187722592795)
--
D has rank: 0.15
A has rank: 1.4901259564203875
B has rank: 0.7832552713203318
C has rank: 1.5766187722592795
```