

CC PHP : Session 1 - 1h30 - Sur machine

Téléchargez et décompressez l'archive déposée sur Moodle pour cet examen. Le dossier résultant contient différents fichiers à réutiliser ou à compléter. A l'issue du temps imparti, archivez votre dossier et déposez l'archive sur Moodle.

Les exercices sont indépendants et la plupart des questions peuvent être traitées dans l'ordre de votre choix. Pour visualiser ce qui est attendu dans les exercices 2 et 3, consultez ce [démonstrateur](#).

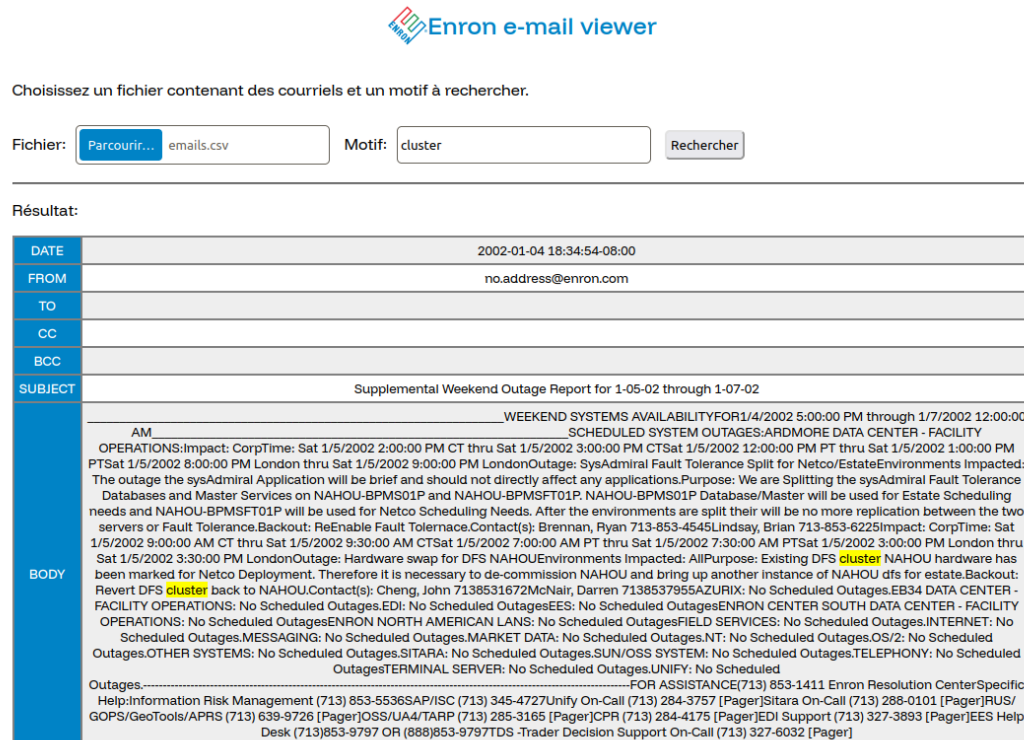
Exercice 1. [Manipulation de tableaux]

Le fichier **logs.txt** contient un historique de logs système, où chaque ligne représente un événement avec un timestamp, un niveau de criticité et un message. Le fichier **logs.php** lit ce fichier et le convertit en `$logs_bruts` stocké dans **solutions/00_logs_bruts.php** : tableau indexé de tableaux indexés (indice 0 pour timestamp, indice 1 pour criticité et indice 2 pour message). L'objectif est d'implémenter différents traitements sur ce tableau : comptage, filtrage, tri, et calculs. Les résultats attendus pour les questions 1 à 5 sont donnés dans le dossier **solutions** et sont importés pour que vous puissiez les comparer aux vôtres.

1. Créez le tableau associatif `$cpt_logs` contenant le nombre de logs pour chaque niveau de criticité (DEBUG, ERROR, FATAL, INFO, WARNING), dans l'ordre alphabétique. Les clés du tableau doivent être les niveaux de criticité, et les valeurs doivent être le nombre de logs correspondants.
2. Triez le tableau `$logs_bruts` et stockez le résultat dans la variable `$logs` en respectant les règles suivantes :
 - par ordre croissant de timestamp (au format `yyyy-mm-dd HH:MM`),
 - en cas d'égalité, par ordre alphabétique du niveau de criticité.
3. À partir du tableau trié `$logs`, stockez dans la variable `$premier_debug` le premier log de niveau DEBUG.
4. Calculez la longueur moyenne des messages de type ERROR, arrondie à 2 chiffres après la virgule, et stockez le résultat dans la variable `$avg_len_msg_error`.
5. Créez la variable `$cpt_dates` contenant le nombre de dates uniques (au format `yyyy-mm-dd`) ayant au moins un log dans le fichier **logs.txt**.
6. [Question bonus] Reproduisez la variable `$logs_bruts` dans la variable `$mes_logs_bruts` à partir du fichier **logs.txt**.

Exercice 2.

On se propose de développer un site pour la recherche de motifs dans des courriels stockés dans un fichier CSV. Le site demande à charger un fichier et saisir un motif, puis affiche tous les courriels dont le corps (colonne BODY dans le fichier CSV) contient le motif, sans différencier entre majuscules et minuscules. Chaque occurrence du motif doit être surlignée en jaune (voir Figure 1). Toutes les fonctions demandées sont à implémenter dans le fichier **fonctions.php**. Le code principal est à développer à l'endroit indiqué dans le fichier **enron.php**. Le fichier **emails.csv** contient une liste de courriels pour tester votre code.



Chisissez un fichier contenant des courriels et un motif à rechercher.

Fichier: emails.csv Motif:

Résultat:

DATE	FROM	TO	CC	BCC	SUBJECT
2002-01-04 18:34:54-08:00	no.address@enron.com				Supplemental Weekend Outage Report for 1-05-02 through 1-07-02
					<p>WEEKEND SYSTEMS AVAILABILITY FOR 1/4/2002 5:00:00 PM through 1/7/2002 12:00:00 AM</p> <p>SCHEDULED SYSTEM OUTAGES: ARDMORE DATA CENTER - FACILITY OPERATIONS: Impact: CorpTime: Sat 1/5/2002 2:00:00 PM CT thru Sat 1/5/2002 3:00:00 PM CTSat 1/5/2002 12:00:00 PM PT thru Sat 1/5/2002 1:00:00 PM PTSat 1/5/2002 8:00:00 PM London thru Sat 1/5/2002 9:00:00 PM LondonOutage: SysAdmiral Fault Tolerance Split for Netco/EstateEnvironments Impacted: The outage the sysAdmiral Application will be brief and should not directly affect any applications. Purpose: We are Splitting the sysAdmiral Fault Tolerance Databases and Master Services on NAHOU-BPMS01P and NAHOU-BPMSFT01P. NAHOU-BPMS01P Database/Master will be used for Estate Scheduling needs and NAHOU-BPMSFT01P will be used for Netco Scheduling Needs. After the environments are split their will be no more replication between the two servers or Fault Tolerance. Backout: ReEnable Fault Tolerance. Contact(s): Brennan, Ryan 713-853-4545 Lindsay, Brian 713-853-6225 Impact: CorpTime: Sat 1/5/2002 9:00:00 AM CT thru Sat 1/5/2002 9:30:00 AM CTSat 1/5/2002 7:00:00 AM PT thru Sat 1/5/2002 7:30:00 AM PTSat 1/5/2002 3:00:00 PM London thru Sat 1/5/2002 3:30:00 PM LondonOutage: Hardware swap for DFS NAHOU Environments Impacted: All Purpose: Existing DFS cluster NAHOU hardware has been marked for Netco Deployment. Therefore it is necessary to de-commission NAHOU and bring up another instance of NAHOU dfs for estate. Backout: Revert DFS cluster back to NAHOU. Contact(s): Cheng, John 7138531672 McNair, Darren 7138537955 AZURIX: No Scheduled Outages. EB34 DATA CENTER - FACILITY OPERATIONS: No Scheduled Outages. EDI: No Scheduled Outages. EES: No Scheduled Outages. ENRON CENTER SOUTH DATA CENTER - FACILITY OPERATIONS: No Scheduled Outages. ENRON NORTH AMERICAN LANS: No Scheduled Outages. FIELD SERVICES: No Scheduled Outages. INTERNET: No Scheduled Outages. MESSAGING: No Scheduled Outages. MARKET DATA: No Scheduled Outages. NT: No Scheduled Outages. OS/2: No Scheduled Outages. OTHER SYSTEMS: No Scheduled Outages. SITARA: No Scheduled Outages. SUN/OSS SYSTEM: No Scheduled Outages. TELEPHONY: No Scheduled Outages. TERMINAL SERVER: No Scheduled Outages. UNIFY: No Scheduled Outages. FOR ASSISTANCE (713) 853-1411 Enron Resolution Center Specific Help: Information Risk Management (713) 853-5536 SAP/ISC (713) 345-4727 Unify On-Call (713) 284-3757 [Pager] Sitara On-Call (713) 288-0101 [Pager] RUS/ GOPS/GeoTools/APRS (713) 639-9726 [Pager] OSS/UA4/TARP (713) 285-3165 [Pager] CPR (713) 284-4175 [Pager] EDI Support (713) 327-3893 [Pager] EES Help Desk (713) 853-9797 OR (888) 853-9797 TDS - Trader Decision Support On-Call (713) 327-6032 [Pager]</p>

FIGURE 1 – Affichage de courriels contenant le mot “cluster” (notez le surlignage en jaune)

1. [Traitement de fichiers CSV] Implémentez la fonction `fichier_a_tableau()` qui prend comme paramètre le nom d'un fichier CSV de même format que **emails.csv**, convertit chaque ligne, sauf la ligne d'en-têtes, en tableau associatif et renvoie ces tableaux dans un tableau indexé. Les clés utilisées dans les tableaux associatifs correspondent aux noms des colonnes du fichier CSV. Voici un exemple de résultat d'appel à la fonction :

```
[
  [
    [date] => 2001-08-01 00:43:14-07:00
    [from] => ryan.slinger@enron.com
    [to] => donald.robinson@enron.com
    [cc] => sean.crandall@enron.com
    [bcc] => diana.scholtes@enron.com
    [subject] => Position Manager for 8/1/01
    [body] => It appears that there is a 10mw transmission
              schedule missing on peak from Pac System...thanks-ryan
  ]
  ...
]
```

2. [Chaînes de caractères] Implémentez une fonction `nettoyer_email()` qui prend une chaîne de caractères en paramètre, en extrait la sous-chaîne contenue entre chevrons `<` et `>`, et y supprime tout caractère interdit dans une adresse e-mail. Considérez comme caractères interdits les point-virgule `;` et esperluettes `&` et étoiles `*`. Par exemple, le résultat de `nettoyer_email("legal <joe*y&kool.shen@enron.com>")` est la chaîne `"joeykool.shen@enron.com"`.

3. [Génération HTML] Implémentez une fonction `generer_html_table_vertical()` qui prend en paramètre un tableau associatif correspondant à un courriel et retourne une chaîne de caractères contenant un tableau HTML à 2 colonnes. Les en-têtes du tableau HTML correspondent aux clés du tableau associatif. Elles forment la première colonne et sont affichées en majuscules tel qu'illustré en Figure 1.

4. [Code principal] Le code principal est à développer dans `enron.php`. Si besoin, utilisez les fonctions des questions précédentes `fichier_a_tableau()` et `generer_html_table_vertical()`.

Votre code doit assurer :

- la récupération des paramètres issus du formulaire.
- la lecture du fichier CSV.
- le filtrage des courriels en fonction du motif donné.
- l'ajout de la balise HTML `...` à chaque occurrence du motif dans le corps du courriel.
- l'affichage des tableaux HTML correspondant aux courriels contenant le motif.

Exercice 3. [API HTTP, fichier JSON]

On propose de développer un formulaire **ed.php** permettant d’extraire des statistiques de soutenances de thèse pour les doctorants inscrits à l’école doctorale MathSTIC (voir Figure 2). La recherche s’effectue selon 2 critères : la discipline scientifique (p. ex. “Chimie”) et l’année de soutenance (entre 2017 et 2023). Au premier chargement de la page (en HTTP GET), le fichier **ed.php** pré-remplit le formulaire pour l’année 2020 et la discipline “Mathématiques” puis affiche les statistiques correspondantes (nombre de soutenances, etc.). Il répond ensuite à chaque soumission (en HTTP POST) en réaffichant données soumises et statistiques associées.

ed.php s’appuie sur deux sources de données :

- un fichier local **disciplines.json** au format JSON décrivant les disciplines scientifiques de l’école doctorale qui sont numérotées de 1 à 6,
- une API HTTP implémentée par le fichier **/api/api.php** qui, pour un numéro de discipline et une année de soutenance communiquées par requête GET, fournit les statistiques correspondantes au format JSON. **api.php** extrait ces données du fichier **/api/soutenances.json** qui stocke les statistiques pour différentes paires <n° discipline, année>. Notez que certaines statistiques ont la valeur `null` (voir Figure 3) et que certaines paires ne sont pas modélisées (voir Figure 4).

1. Complétez le fichier **ed.php** en suivant les instructions données en commentaires dans le code source.
2. [Question bonus] Le fichier **api.php** est déjà implémenté mais le code source est obfusqué. Ré-implementez les mêmes fonctionnalités dans le fichier **api-clone.php** en suivant les instructions données en commentaires dans le code source. Testez-le directement dans votre navigateur avec de bonnes URL (incluant chaîne de requête).

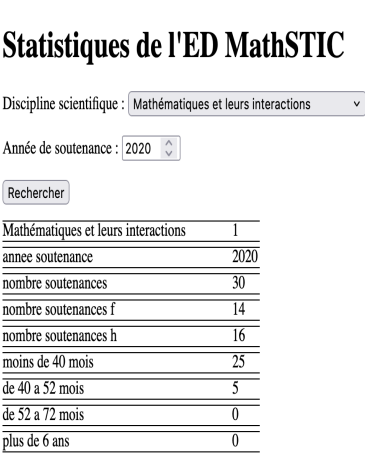


FIGURE 2 – Page d’accueil

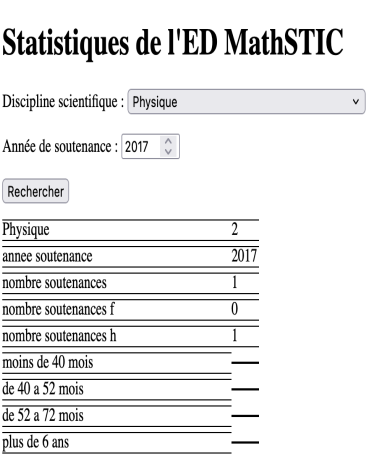


FIGURE 3 – Statistiques inconnues

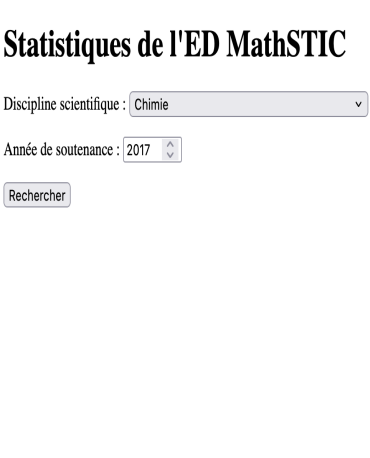


FIGURE 4 – Année/discipline non modélisée