Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Scuola d’arti mestieri Trevano |
| Data | 14.10.2020 |

|  |
| --- |
| Lavori svolti |
| Oggi ho continuato la pagina visualizza\_backup. In effetti il metodo viewBackup del controller visualizza\_backup non si chiama più così ma lo rinominato in getDbLinksFileBackup.  Immagine che contiene testo  Descrizione generata automaticamente  In questo metodo ho risolto i problemi che mi sono compparsi ieri. Come ad esempio l’indice della delll’array $row. L’array $row è un array associativo e quindi al suo interno posso solo scrivere un indice testuale (stringa). In questo caso voglio recuperare il nome del database e nel database db\_link questo dato si trova nella colonna dbname. Ho anche incrementato la variabile $i, in modo che ad ogni ciclo cambio riga all’interno dell’array $all\_file. Ogni riga di quest’ultimo corrisponde ad un collegamento. Ho aggiunto anche lo stato del backup nella quarta colonna.  Ho aggiornato anche il metodo viewBackupFile del models backup\_model.php. Quest’ultimo lo rinominato in getDbLinkBackupFile.  Immagine che contiene testo  Descrizione generata automaticamente  Immagine che contiene testo  Descrizione generata automaticamente  Il metodo riceve come parametro $dir\_dblink che corrisponde al nome della directory del collegamento.  La variabile $files è un array nel quale salvo l’ultimo file SQL del backup, l’ultimo file log e lo stato del backup. Nella variabile $path e $path\_dir\_log salvo il percorso della directory del collegamento e la directory dei file log del collegamento. In caso che la directory non esiste vuole dire che il backup del collegamento (almeno il primo backup, perché la cartella del collegamento viene creata al primo backup) non è stato eseguito e quindi salvo nell’array $file nella prima e la seconda colonna due stringhe vuote per il file SQL e il file log. Per lo stato del backup salvo all’interno dell’array $file false, perché non è stato eseguito un backup. Se risco a leggere il contenuto della directory del collegamento + directory log del collegamento. Posso salvare nella prima colonna dell’array $file l’ultimo file SQL della directory. Nella seconda colonna salvo l’ultimo file log e per finire nell’ultima colonna salvo lo stato del backup. In effetti per sapere se lo stato del backup ho implementato un metodo che si chiama statusBackup all’interno del model backup\_model.  Immagine che contiene testo  Descrizione generata automaticamente  Quest’ultimo riceve anche come parametro il nome del file log del corrispettivo collegamento. Con il metodo fopen posso aprire un file nella modalità che voglio. In effetti questo metodo riceve due parametri. Il primo corrisponde al nome del file e il secondo corrisponde la modalità. Come modalità ho messo r che corrisponde lettura (quindi apro il file solo in lettura). Poi ho creato un ciclo che permette di leggere solo alcune righe di un file. In effetti leggo solamente le prime due righe del file log, in modo che alla seconda riga del file verifico che l’utente è risucito ad collegarsi al datbase. Se riesce ad collegarsi vuole dire che il backup è stato eseguito in modo corretto. In effetti per verificare che una stringa contiene una determinata stringa di caratteri, utilizzo il metodo strpos.  Ho aggiornato anche la views visualizza\_backup aggiustando anche i problemi di ieri.  Immagine che contiene testo  Descrizione generata automaticamente    Questa tabella mi mostra per ogni collegamento i file di backup (SQL), viene anche mostrato il file log del corrispettivo backup e per finire anche lo stato del backup.  Per cambiare il colore del segnale dello stato controllo se nella variabile $file se all’indice 3 (nel seguente indice viene salvato lo stato del backup). Se il backup è andato a buon fine salvo nella variabile $colorStatus il nome della classe bootstrap che permette impostare il colore verde di un pulsante (btn-success), altrimenti imposta il colore rosso (btn-danger). Ho creato all’interno del controller visualizza\_backup due metodi che si chiamano getContentFile e getContentFileLog. Qust’ultimi permettono di visualizzare il contenuto del file SQL e log.    Il metodo file\_get\_contents permette di leggere il contenuto di un file e con il metodo nl2br posso stampare il contenuto tenendo conto anche il return carriage.  Ho anche creato un pulsante report che permette di creare un report. Quando clicco sul pulsante mi richiama il metodo createReport. All’inizio creavo questo pulsante per ogni collegamento, ma poi dalla discussione con Valsangiacomo ho optato per un solo pulsante che contiene tutte le informazioni dei vari collegamenti. Per creare un file pdf utilizzo la libreria fpdf. FPDF è una classe PHP che permette di generare files PDF direttamente da PHP, senza utilizzare la liberia PDFlib. La F di FPDF sta per "Free": puoi usarla per qualsiasi utilizzo e modificarla per soddisfare le tue esigenze. FPDF ha anche altri vantaggi: funzioni di alto livello. Ecco una lista delle sue caratteristiche principali:   * Scelta dell'unità di misura, del formato pagina e dei margini * Utilizzo dell'intestazione e del piè di pagina * Cambio pagina automatico * Ritorno a capo automatico e giustificazione del testo * Supporto immagini (JPEG, PNG e GIF) * Colori * Links * TrueType, Type1 e supporto codifica * Compressione pagine   Questa libreria può essere scaricata dal sito <http://www.fpdf.org> .  Immagine che contiene testo  Descrizione generata automaticamente  Dopo aver incluso la libreria, creiamo un oggetto FPDF. Il costruttore viene usato, qui sopra, con i valori di default: le pagine sono formato A4, verticale, e con unità di misura in millimetri.  Con il metodo AddPage permette di aggiungere una nuova pagina al documento. Con il metodo SetFont posso implementare il font utilizzato per stampare stringhe di caratteri. Per fenire il metodo output passa il documento a una destinazione: stringa, file locale o browser. Con l’opzione D manda al browser e forza il download del file con il nome dato con name. |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| Oggi non ho riscontrato dei problemi. Però mi sono reso conto che quando cancello un collegamento, devo anche cancellare tutta la directory che contiene tutti i file di backup. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Rispetto alla pianificazione sono ad un buon punto, perché oggi ho implemenatato un sistema che permetta di visualizzare il contenuto dei file. Ho anche implementato un segnale che mostra lo stato del backup. Per fenire ho implementato un pulsante che permette di creare un report in formato pdf. |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Per la prossima giornata devo terminare il metodo che permette di crere un pdf. Cioè al interno del file devo salvare il contenuto dei vari collegamenti. Finito questa attività posso implementare con PHPMailer un sistema che invia le email quando c’è stato un problema con il backup. |