

Diario di lavoro

Luogo	Canobbio
Data	09.03.2020

Lavori svolti

08h20 – 08h50

Mi sono occupato di risolvere un warning di PHP il quale segnalava che alcuni headers venivano modificati dopo che il contenuto della pagina era già stato inviato all'utente. Il problema affliggeva il percorso riguardante la creazione dei PDF. La libreria utilizzata per creazione dei PDF si occupava già di mandare gli header adeguati, che però venivano rimandati anche dalla libreria php-rest la quale si occupa di gestire le risposte. Vedi sezione "Problemi riscontrati e soluzioni adottate" per la soluzione del problema.

08h50 – 09h10

Ho aggiunto il nome del file dei PDF il quale viene generato automaticamente con l'ultima data di aggiornamento del congedo. Inoltre ho aggiunto anche un bottone il quale permette di scaricare il PDF dei congedi direttamente dal browser.

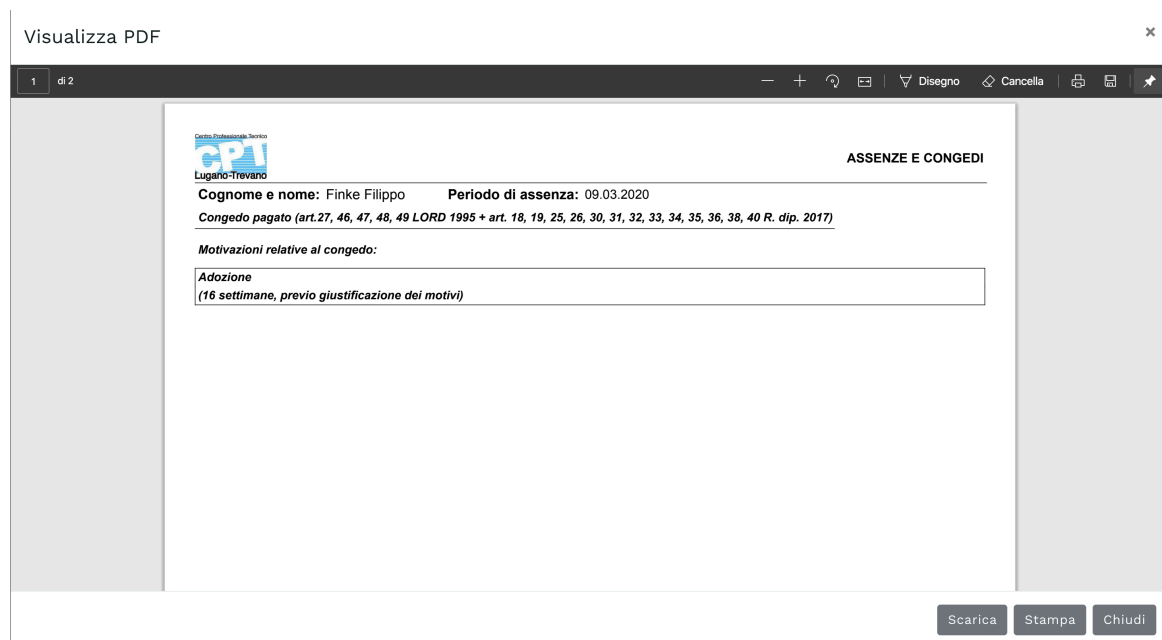


Figura 1 Schermata visualizzazione PDF.

09h10 – 09h40

Ho risolto un problema nell'aggiornamento dei congedi il quale non permetteva il salvataggio delle modifiche della tipologia di supplenza, il problema era dovuto alla riassegnazione di una variabile. Vedi sezione "Problemi riscontrati e soluzioni adottate" per la soluzione del problema.

09h40 – 09h50

Ho reso dinamica la selezione del font nella stampa del nome e cognome dei supplenti all'interno della generazione del PDF, questo in modo che il testo sia sempre all'interno delle celle. Per fare questo ho utilizzato un ciclo il quale controlla la lunghezza della stringa da stampare decrementando la grandezza del font fino a quando resta all'interno della cella:

```
$fontSize = 7;
while ($this->GetStringWidth($substitute) >= $hoursWidth) {
    $this->SetFont('Arial', '', $fontSize);
    $fontSize--;
}
```

In questo modo il font è dinamico e il testo verrà sempre stampato all'interno della cella.

		Ora	08:20 09:05	09:05 09:50
LUNEDÌ	09.03.2020	Classe	I4AC	
		Aula	A-417	
		Supplente	SI Valsangiacomo	

Figura 2 Risultato font dinamico.

10h05 – 11h35

Mi sono occupato di commentare tutto il codice sorgente del progetto scritto fino ad ora.

13h15 – 14h45

Ho caricato l'ultima versione dell'applicativo su infomaniak, successivamente dopo svariati test di performance mi sono accorto che in alcuni casi il caricamento del sito è molto lento, questo dovuto anche all'hosting. Ho dunque scaricato tutte le dipendenze localmente e creato una versione compressa di tutte le risorse in modo da far scaricare meno dati agli utenti. Successivamente non essendo contento del tempo di caricamento di circa 6 secondi:

login	200	document	Other	1.8 KB	71 ms
work-sans.css	200	stylesheet	login	729 B	40 ms
style.css	200	stylesheet	login	67.7 KB	69 ms
custom.css	200	stylesheet	login	920 B	69 ms
script.js	200	script	login	1.0 MB	430 ms
jquery.highlight.js	200	script	login	4.2 KB	168 ms
functions.js	200	script	login	2.7 KB	106 ms
QGYsz_wNahGAdqQ43Rh_fkDptfpA4Q.woff2	200	font	login	15.1 KB	112 ms
QGYpz_wNahGAdqQ43Rh3j4P8mNhNy_r-Kw...	200	font	login	16.6 KB	60 ms
fontawesome-webfont.woff2	200	font	login	75.7 KB	100 ms
jquery.mousewheel.min.js	307		script.js:2	0 B	6 ms
jquery.mousewheel.min.js	200	script	jquery.mousewheel.mi...	1.2 KB	21 ms
favicon.ico	403	text/html	Other	1.3 KB	39 ms
35 requests 1.2 MB transferred 3.3 MB resources Finish: 5.48 s DOMContentLoaded: 5.38 s Load: 5.43 s					

Figura 3 Analisi richieste, 6 secondi per caricare.

Ho aggiunto alla risposta di qualsiasi risorsa degli headers per permettere al browser di eseguire una copia locale dei file per un determinato tempo, in questo caso per un'ora. Il codice è il seguente:

```
header_remove("Pragma");
return $response
->withHeader("Cache-Control", "public, max-age=3600")
->withHeader("Expires", gmdate('D, d M Y H:i:s', time()+3600).'GMT')
->withHeader("Content-Type", $type)
->withBody(file_get_contents($file))
->withStatus(200);
```

Il codice in questione si occupa di aggiungere l'header Cache-Control per il protocollo HTTP versione 1.1 dicendo di eseguire una cache per un ora dei file. Viene aggiunto l'header Expires per supportare anche HTTP 1.0 e viene rimosso l'header Pragma per permettere la cache.

Name	Status	Type	Initiator	Size	Time
login	200	document	Other	1.8 KB	33 ms
work-sans.css	200	stylesheet	login	(memory cache)	0 ms
style.css	200	stylesheet	login	(memory cache)	0 ms
custom.css	200	stylesheet	login	(memory cache)	0 ms
script.js	200	script	login	(memory cache)	0 ms
jquery.highlight.js	200	script	login	(memory cache)	0 ms
functions.js	200	script	login	(memory cache)	0 ms
QGYsz_wNahGAdqQ43Rh_fkDptfA4Q.woff2	200	font	script.js:28	(memory cache)	0 ms
QGYpz_wNahGAdqQ43Rh3j4P8mNhNy_r-Kw.woff2	200	font	script.js:28	(memory cache)	0 ms
QGYpz_wNahGAdqQ43Rh3x4X8mNhNy_r-Kw.woff2	Finished	font	script.js:28	(memory cache)	0 ms
fontawesome-webfont.woff2	200	font	script.js:28	(memory cache)	0 ms
fontawesome-webfont.woff	Finished	font	script.js:28	(memory cache)	0 ms
fontawesome-webfont.ttf	Finished	font	script.js:28	(memory cache)	0 ms
foundation-icons.woff	Finished	font	script.js:28	(memory cache)	0 ms
foundation-icons.ttf	Finished	font	script.js:28	(memory cache)	0 ms
ionicons.ttf	Finished	font	script.js:28	(memory cache)	0 ms
ionicons.woff	Finished	font	script.js:28	(memory cache)	0 ms
themify.woff	Finished	font	script.js:28	(memory cache)	0 ms
themify.ttf	Finished	font	script.js:28	(memory cache)	0 ms
jquery.mousewheel.min.js	307		script.js:2	0 B	8 ms
jquery.mousewheel.min.js	200	script	jquery.mousewheel.min.js	(disk cache)	1 ms
favicon.ico	403	text/html	Other	1.3 KB	33 ms

22 requests 3.1 KB transferred 1.6 MB resources Finish: 255 ms DOMContentLoaded: 175 ms Load: 215 ms

Figura 4 Analisi richieste, 255 ms per caricare.

La cache ha abbassato i tempi di caricamento da 5,48 secondi a 255 millisecondi sulla stessa pagina ovvero il 2149% più veloce.

Problemi riscontrati e soluzioni adottate

Problema headers

La libreria di creazione dei PDF si occupa di inviare gli headers adeguati e di stampare il contenuto se il tipo di output è "I", ho quindi aggiunto una condizione il quale permette di distinguere quale tipo di output si sta utilizzando e in base a quello aggiungere ulteriori headers oppure no. Ho dunque utilizzato il seguente codice per eseguire la distinzione:

```
public function getContent($type)
{
```

```

$output = $this->Output($type);
if ($type == "I") {
    return true;
} else {
    return $output;
}
}

```

Il quale poi viene gestito dal controller delle richieste di congedo. Il controllo è il seguente:

```

if (!ModelsRequests::generatePdfForId($id)) {
    return $response->withStatus(404)->withText("Il congedo non è stato trovato!");
}

```

In questo modo non vengono inviati headers dopo che del contenuto è già stato spedito all'utente e quindi il problema è stato risolto.

Problema salvataggio modifiche congedi

Ho riscontrato un problema nel salvataggio delle modifiche dei congedi il quale era causato dalla riassegnazione di una variabile con dei valori errati. Per risolverlo ho semplicemente istanziato un'altra variabile la quale si occupa di tenere lo stato originale della variabile che andrà modificata.

```

var type = "";
if(substitute.type != "") {
    type = " (" + substitute.type + ")";
}

```

Punto della situazione rispetto alla pianificazione

Al momento ho un leggero vantaggio rispetto alla pianifica.

Programma di massima per la prossima giornata di lavoro

Aggiornare la documentazione.