

Aggiungere icon font personalizzati

Cosa sono le icon font? A cosa servono?

Nello sviluppo di applicazioni ci troviamo spesso a dover inserire icone all'interno delle nostre interfacce utente e solitamente utilizziamo immagini bitmap (bmp, jpeg, gif, png, etc.) recuperate in rete o create ad hoc. Questo approccio però porta con se una serie di inconvenienti come ad esempio problemi di scalabilità con risoluzioni differenti (ad esempio su interfacce responsive), aumento del consumo di banda e una certa difficoltà in caso decidessimo, ad esempio, di cambiare l'aspetto dell'immagine in base ad una selezione.

Per risolvere questi problemi possiamo utilizzare le icon immagine e, come tali, godono di tutti vantaggi dei font tra i quali:

- Sono vettoriali e quindi scalano senza sgranare
- Possono essere cambiati di colore tramite css
- Possono essere ombreggiati tramite css
- Occupano meno spazio (e quindi utilizzano meno banda) rispetto ad un'immagine

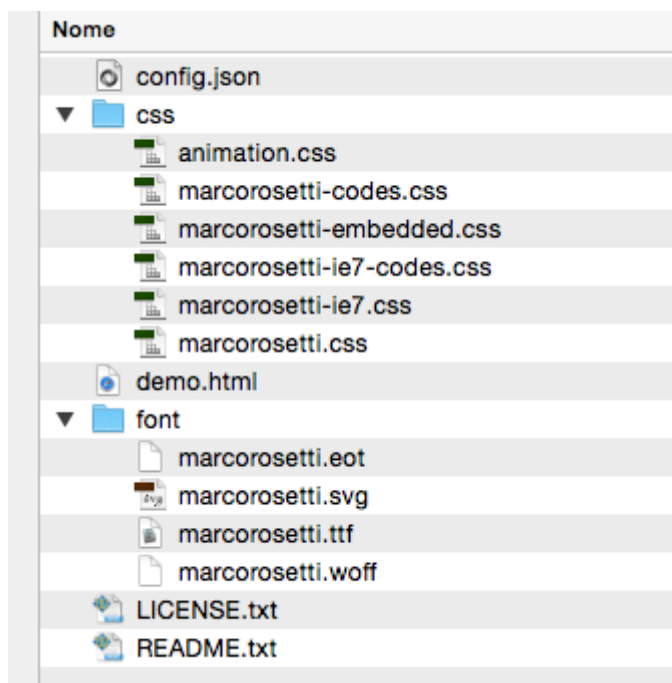
A partire dalla versione 6.2 Liferay include al suo interno un pacchetto di icon font (la lista è disponibile qui: <http://liferay.github.io/alloy-bootstrap/base-css.html#icons>) e possono essere utilizzati out-of-the-box.

Esistono in rete molti siti che permettono di creare, anche in modo gratuito, dei set di icon font scegliendo da elenchi interminabili di possibilità quelle che ci interessano. Citiamo ad

esempio <http://www.fontello.com/>, <http://www.flaticon.com/> e <https://icomoon.io/>. In questo articolo vedremo come utilizzare le icon font di un pacchetto custom all'interno della nostra portlet.

Creare un pacchetto personalizzato di icon font

Prendiamo come esempio il sito Fontello (<http://www.fontello.com/>). Possiamo selezionare tutte le icone che vogliamo inserire all'interno del nostro set e scaricarle in un unico archivio zip.



Vediamo nel dettaglio cosa contiene l'archivio creato:

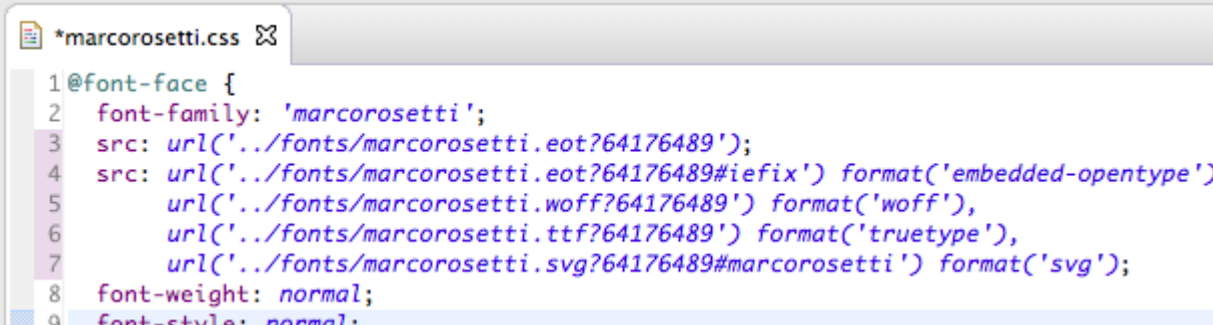
- Cartella CSS – i fogli di stile che tra poco copieremo all'interno del nostro plugin
- Cartella Font – i file contenenti il font vero e proprio, anche questi dovranno essere copiati nel nostro plugin
- config.json – un file in formato JSON che contiene le indicazioni su tutte le icone che fanno parte del nostro pacchetto. Caricando questo file sul sito avremo tutte

le nostre icone già selezionate e impostate: in questo modo è più semplice modificare il nostro set senza dover tutte le volte ripartire da capo

- demo.html – file di demo dell'intero set di icone da aprire in qualsiasi browser
- LICENSE.txt – file di licenza
- README.txt – istruzioni di installazione. Da leggere sempre!! ☐

Aggiungere le icon font al proprio plugin

1. Copiare i file dei font all'interno di docroot. Io consiglio di utilizzare la sottocartella fonts ma non è obbligatorio
2. Copiare i file css all'interno di docroot. Anche in questo caso nessun path è obbligatorio ma consiglio di utilizzare la sottocartella css. In realtà per il nostro esempio ci interessa solo il file senza suffisso, nel nostro caso marcorosetti.css
3. Apriamo il file marcorosetti.css e per prima verificiamo (ed eventualmente correggiamo) i path relativi ai font che troviamo all'interno dell'istruzione `@font-face`



```
*marcorosetti.css ☒
1 @font-face {
2   font-family: 'marcorosetti';
3   src: url('../fonts/marcorosetti.eot?64176489');
4   src: url('../fonts/marcorosetti.eot?64176489#iefix') format('embedded-opentype')
5         url('../fonts/marcorosetti.woff?64176489') format('woff'),
6         url('../fonts/marcorosetti.ttf?64176489') format('truetype'),
7         url('../fonts/marcorosetti.svg?64176489#marcorosetti') format('svg');
8   font-weight: normal;
9   font-style: normal;
```

4. Assicuriamoci che il selettore css e le classi del set non inizino con "icon-" o potremmo ritrovarci con alcune incompatibilità con gli icon font di default di liferay. Possiamo modificare il file css a mano oppure utilizzare le impostazioni avanzate di esportazione di fontello. Il

risultato da ottenere dovrà essere simile a questo:

```
22 [class^="myicon"]:before, [class*=" myicon"]:before {
56
57 .myiconsun:before { content: '\e800'; }
58 .myiconcloud-sun:before { content: '\e801'; }
59 .myiconfog-sun:before { content: '\e802'; }
60 .myiconcloud-flash-alt:before { content: '\e803'; }
61 .myiconsnow:before { content: '\e804'; }
62 .myiconrain:before { content: '\e805'; }
```

5. Sempre per questioni di compatibilità meglio rimuovere le direttive su width e margin-right:

```
23 font-family: "marcorosetti"
24 font-style: normal;
25 font-weight: normal;
26 speak: none;
27
28 display: inline-block;
29 text-decoration: inherit;
30 /*width: 1em;
31 margin-right: .2em;*/
32 text-align: center;
33 /* opacity: .8; */
34
```

6. Infine aggiungiamo al file liferay-portlet.xml i riferimenti ai css <header-portlet-css>/css/marcorosetti.css</header-portlet-css> in modo che il css venga caricato automaticamente su ogni nostra pagina

```
15 <portlet>
16   <portlet-name>Icon Font Test</portlet-name>
17   <icon>/icon.png</icon>
18   <header-portlet-css>/css/main.css</header-portlet-css>
19   <header-portlet-css>/css/marcorosetti.css</header-portlet-css>
20   <footer-portlet-javascript>
21     /js/main.js
22   </footer-portlet-javascript>
23   <css-class-wrapper>icon font test-portlet</css-class-wrapper>
24 </portlet>
25 </portlet->
```

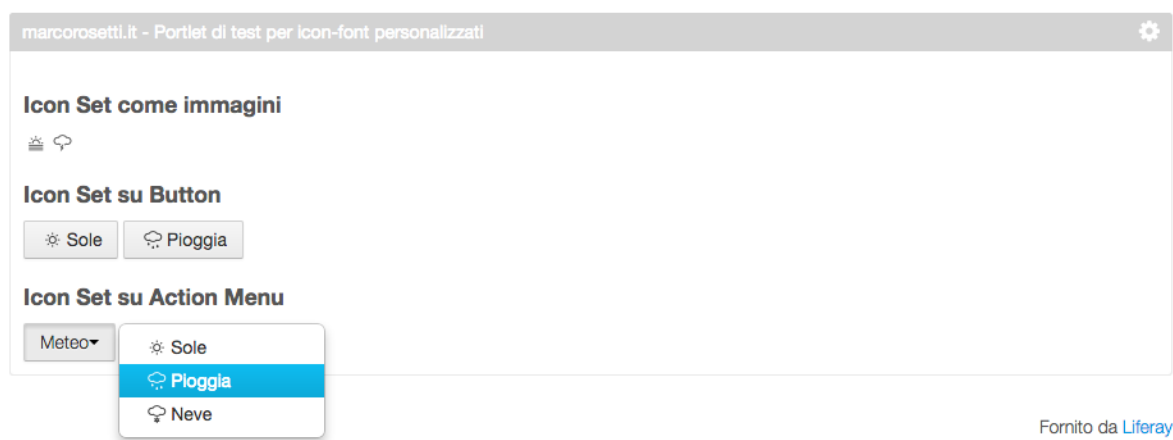
Esempi di utilizzo

A seconda del componente in cui vogliamo inserire l'icona l'utilizzo è leggermente diverso.

Ecco il codice di una piccola jsp di esempio per l'utilizzo come semplice immagine (tag <i>), in un pulsante (tag <auibutton>) e tra le azioni di un icon-menu (tag <liferay-

ui:icon> all'interno di <liferay-ui:icon-menu>)

E questo è il risultato finale



Approfondimenti e riferimenti

- <http://www.html.it/articoli/icon-fonts-1/>
- <http://www.fontello.com/>
- <http://www.flaticon.com/>
- <http://www.vanseodesign.com/web-design/icon-fonts/>
- <https://css-tricks.com/examples/IconFont/>