# Riferimenti al codice trovato al di fuori del materiale delle lezioni e adattato

I seguenti link sono stati consultati durante lo svolgimento del progetto e rivisitati in data 11 aprile 2021.

## Riferimenti per la web-application ed il client REST implementati in Java

* **Download di un file in *JAX-RS*:** <https://stackoverflow.com/a/12251265>  
  (utilizzando il *MediaType “application/octet-stream”*)
* **Upload di un file in *JAX-RS*:** <https://stackoverflow.com/a/25889454>   
  (utilizzando il *MediaType “multipart/form-data”*)
* **Utilizzo di Objectify fuori dal contesto di una *request*:** <https://stackoverflow.com/a/34484715>
* **Conversione da *Input Stream* a *byte[]* in Java.** <https://stackoverflow.com/a/1264737>
* ***Mediatype “multipart/form-data”* dal client REST (in Java)** <https://stackoverflow.com/q/24637038>

## Riferimenti per l’applicazione Vue ed il codice JavaScript

* **Configurazione di *axios* per invio di Cookie e credenziali** <https://stackoverflow.com/a/55643460>
* **Download di un file (*Blob*)** <https://stackoverflow.com/q/33247716>
* **Upload di un file (content-type: *multipart/form-data*)** <https://stackoverflow.com/a/43014086>
* **Ordinamento di un oggetto *Map*** <https://stackoverflow.com/a/50427905>
* **Utilizzo della funzione *Promise.all*** <https://stackoverflow.com/a/31414472>

## Riferimenti alla documentazione consultata (relativa ad argomenti non trattati a lezione) o alle porzioni di codice utilizzate per implementare funzionalità non richieste dai requisiti di progetto

I seguenti link fanno riferimento alla documentazione consultata durante lo sviluppo del progetto, relativa ad argomenti non direttamente trattati a lezione. Nella lista sono presenti anche i link visitati per implementare delle funzionalità aggiuntive non esplicitamente richieste dai requisiti (ad esempio la protezione *CSRF*), aggiunte per completezza al progetto.

* Classe *Cookie* in *JAX-RS*: <https://stackoverflow.com/a/28004533>
* *Annotation* *@Context* in *JAX-RS*: <https://stackoverflow.com/a/14255549>  
  (per ottenere l’istanza di *HttpServletRequest*)
* Interfaccia *ExceptionMapper* in *JAX-RS*: <https://stackoverflow.com/a/45758691>  
  (per ottenere maggiori informazioni sulle eccezioni generate da *Jersey*)
* Utilizzo dell’artefatto *jersey-media-multipart* <https://stackoverflow.com/a/25312655>,

<https://stackoverflow.com/a/29227044>

* Esecuzione di task periodici in *App Engine*: <https://stackoverflow.com/a/15979300>
* *Wrapper* per *HttpServletRequest*: <https://stackoverflow.com/a/25640232>  
  (per leggere il valore il *body* di una *HTTP Request*, evitando i problemi di “*stream* chiusi” dovuti all’utilizzo del metodo *ServletRequest.getReader()* )
* Classe *SystemProperty* <https://cloud.google.com/appengine/docs/standard/java/tools/using-local-server#detecting_the_application_runtime_environment>  
  (per distinguere l’ambiente di sviluppo da quello di produzione in *Google App Engine*)
* Utilizzo di un *plugin Maven* per eseguire *npm* e *Node.js* <https://stackoverflow.com/a/62535901>
* Utilizzo di un *plugin Maven* per copiare una cartella <https://stackoverflow.com/q/28584442>
* Generazione di *Javadoc* da *Maven* <https://maven.apache.org/plugins/maven-javadoc-plugin/javadoc-mojo.html#show>
* Web Server GAE configurato per HTTPS <https://cloud.google.com/appengine/docs/standard/java/config/webxml#secure-urls>
* Utilizzo del file *index.yaml* per il *GAE Datastore* <https://cloud.google.com/appengine/docs/standard/java/configuring-datastore-indexes-with-index-yaml>
* Istanze di Google App Engine <https://cloud.google.com/appengine/docs/standard/java/an-overview-of-app-engine#instances>
* Utilizzo di *Objectify* in test *JUnit* <https://stackoverflow.com/a/36247734>
* Utilizzo di *sweetalert* per *alert* personalizzati <https://sweetalert.js.org/guides/>
* Utilizzo di *Map* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map>
* Utilizzo della funzione *flatMap* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/flatMap>
* Decodifica di una stringa in formato *Base64* <https://developer.mozilla.org/en-US/docs/Web/API/WindowOrWorkerGlobalScope/atob>
* Utilizzo della funzione *Object.freeze()* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/freeze>
* Conversione di una stringa dal formato *Camel-Case* <https://stackoverflow.com/a/7225450>
* Impostare le dimensioni degli script in cui suddividere l’output del progetto *Vue*

<https://stackoverflow.com/a/52634444>

* Modificare le configurazioni di ottimizzazione e prestazioni dei *chunk* prodotti da *Webpack*

<https://webpack.js.org/configuration/optimization/>

<https://webpack.js.org/plugins/split-chunks-plugin/>

https://webpack.js.org/configuration/performance/

* Disabilitare l’ereditarietà degli attributi nei componenti *Vue* <https://v3.vuejs.org/guide/component-attrs.html#disabling-attribute-inheritance>
* *Re-mount* di un componente *Vue* <https://stackoverflow.com/a/47466574>
* Utilizzare *Firebase* in *Vue* <https://medium.com/dailyjs/authenticating-a-vue-js-application-with-firebase-ui-8870a3a5cff8>

<https://firebase.google.com/docs/web/setup#node.js-apps>

* Impostare lo stato di persistenza di *Firebase* <https://firebase.google.com/docs/auth/web/auth-state-persistence#supported_types_of_auth_state_persistence>
* In *Firebase*, impostare login tramite popup che permetta di scegliere con quale account *Google* autenticarsi

<https://stackoverflow.com/a/59744590>