

# Progetto “Collectors Site Services”

Versione 2.0

## Premessa

---

Il presente progetto estende un progetto del corso di Web Engineering. Per prima cosa, quindi, potrebbe essere utile leggere questa specifica di base. Tuttavia, non si dovrà realizzare la *web application* con tutte le funzionalità descritte nella specifica base, ma solo la relativa base di dati, a cui poi affiancare il servizio web RESTful che esporrà questi dati e alcune delle relative funzionalità di accesso e modifica. Si dovrà infine implementare, utilizzando HTML5 e JQuery, un semplicissimo client per questo servizio, utile anche a testarne la correttezza.

Nota: il servizio richiesto realizza, in pratica, un disaccoppiamento della gestione dati (e di alcune funzionalità correlate), che risiede necessariamente sul server, dalla logica di business e dal rendering dell'interfaccia, che potrebbero essere totalmente gestite dal client con Javascript, come spesso accade nelle SPA (*single page application*, applicazioni a pagina singola).

## Il Servizio

---

A partire dalla base di dati del progetto *Collectors Site* (che non dovete necessariamente realizzare ex novo se non siete anche studenti del corso di Web Engineering: siete liberi di usare il database realizzato da qualche vostro collega), si costruisca il servizio RESTful specificato di seguito. La logica del servizio potrà, opzionalmente, utilizzare una libreria ORM per accedere ai dati.

### Operazioni:

1. Login/logout con username e password
2. Elenco collezioni (personali) dell'utente (qui e successivamente con “utente” intendiamo quello autenticato)
3. Elenco collezioni condivise con un utente
4. Elenco dei dischi in una collezione (accessibile dall'utente)
5. Dettagli singolo disco contenuto in una collezione
6. Inserimento di un nuovo disco in una delle collezioni dell'utente
7. Ricerca di un disco (in base a vari criteri) tra le collezioni personali, tra le collezioni condivise o tra quelle pubbliche (tre *endpoint* diversi)
8. Elenco di tutti gli autori presenti nel sistema
9. Elenco di tutti i dischi di un certo autore presenti nelle collezioni pubbliche
10. Aggiornamento di un disco in una collezione personale dell'utente
11. Estrazione di alcune statistiche (si veda la specifica del progetto Collectors Site per alcuni suggerimenti) dalla collezione di un utente

Prestate attenzione alla *struttura delle URL*, che deve seguire per quanto possibile il paradigma collection-item, evidenziando le relazioni tra gli oggetti tramite nidificazione.

Se alcune operazioni necessitano di accesso con autenticazione, potrete usare uno qualsiasi degli approcci visti a lezione per trasmettere il *token di sessione/autenticazione*, precedentemente ottenuto tramite la procedura (API) di login: ad esempio inserire un prefisso *auth/{SID}* sulla URL o passare tale token tramite *cookie* o con l'header *Authentication*.

## Il Client

---

Si crei un client HTML5/Javascript con il supporto di JQuery per il servizio appena descritto, sotto forma di una semplice *single page application* che utilizza AJAX per interagire col servizio mostrandone i dati opportunamente sulla pagina.

In particolare, il client dovrà permettere di navigare tra le collezioni personali di un utente: dalla lista delle collezioni alla lista dei dischi nella collezione fino ad arrivare ai dettagli di un singolo disco.

Questa è una specifica di base. Potete aggiungere altre funzionalità a vostra scelta che sfruttino le altre API del servizio, se volete, aumentando così il valore del progetto.

Nota: normalmente, una SPA di una certa complessità andrebbe realizzata sfruttando framework di più alto livello come Angular o Vue. Tuttavia, per applicazioni semplici, l'uso di JQuery a basso livello risulta essere più pratico, in quanto l'uso di framework come AngularJS aggiunge un notevole overhead al progetto.

## Tecnologie da utilizzare

---

- Nel realizzare le pagine web, se necessarie, si utilizzi HTML5.
- Per la programmazione lato *client*, oltre a JQuery, si possono liberamente includere nel progetto altre librerie e plugin, a patto che nella relazione siano citate e descritte.
- Per la programmazione lato *server*, il servizio potrà essere realizzato in *JAVA* o *PHP*. Sviluppando in Java, si utilizzi preferibilmente un'implementazione di *JAX-RS* (come Jersey), che permette di realizzare servizi con una struttura più modulare. L'uso di framework come *Spring* per creare il servizio è ammesso, ma solo se il progetto dovrà essere integrato o esteso, ad esempio con quello di un altro corso. In PHP, invece, è possibile usare un framework come *Slim* o un altro a scelta.

## Svolgimento e Documentazione del Progetto

---

Le specifiche fornite potrebbero non risultare esaustive o completamente definite. Ogni funzionalità aggiunta o raffinata, anche tramite l'interazione con il committente o con gli utenti finali del sito, sarà adeguatamente valutata. Tutte le scelte progettuali vanno comunque discusse e motivate.

Il progetto, svolto secondo le linee guida date dalle specifiche, dovrà essere consegnato nella forma di una o più web application completamente funzionanti, i cui contenuti e le cui caratteristiche saranno valutati in sede d'esame. Le parti della specifica marcate come *opzionali*, se omesse, non renderanno il progetto insufficiente ma non gli permetteranno comunque di raggiungere il massimo dei voti. Nel caso si decida di realizzarle, non sarà necessario che siano perfette o complete, ma che dimostrino chiaramente il vostro impegno nell'affrontare una tematica avanzata.

La documentazione (**in formato elettronico**) che accompagna il progetto **deve** contenere almeno le seguenti informazioni:

1. Indicazione delle **dipendenze** software (di quali librerie avete bisogno dal lato server e client?).
2. Indicazione delle **funzionalità realizzate** e di quelle eventualmente non realizzate. Descrizione dettagliata delle eventuali funzionalità extra o opzionali inserite nel progetto.
3. **Specifica dettagliata dell'API** redatta in formato *OpenAPI* (ad esempio tramite il tool di *Swagger*, <https://swagger.io/tools>): per ogni operazione realizzata dovrete specificare

- la URL,
- gli eventuali parametri, dove sono inseriti (nel path o nella query string) e i loro possibili valori,
- il verbo HTTP usato,
- il formato del payload di richiesta (se necessario),
- i codici di stato http ritornati e, per ogni stato, il formato dell'output (se presente).

Per quel che riguarda il formato dei dati scambiati (payload e output) fornite sempre un esempio completo della struttura JSON che intendete usare.

Nel caso di gruppi di lavoro composti da più componenti, *il contributo effettivo offerto da ciascun componente* alla realizzazione finale **deve** essere descritto nella documentazione. In sede di esame, i responsabili potranno essere chiamati a riferire sugli aspetti loro delegati.

## Valutazione del Progetto

---

Nel valutare il progetto consegnato saranno prese in considerazione le seguenti caratteristiche (in ordine di importanza):

1. Rispetto delle specifiche.
2. Correttezza tecnica.
3. Uso appropriato delle tecnologie di base descritte nel corso.

A questa valutazione si aggiungerà quella generale derivata dalla discussione del progetto in sede d'esame.

## Ulteriori Informazioni

---

Questa specifica è disponibile in formato PDF sulla pagina web del corso di Sviluppo Web avanzato. Ulteriori informazioni e chiarimenti sulle specifiche possono essere richiesti direttamente via email all'indirizzo [giuseppe.dellapenna@univaq.it](mailto:giuseppe.dellapenna@univaq.it).

Si ricorda che i progetti vanno svolti in *piccoli* gruppi. Eccezioni a questa regola andranno concordate direttamente col docente.