

Tecnologie Web 23-24

Proposta di Progetto

Idea generale

Si vuole realizzare un **servizio di Media Discovery and Request** per un utilizzo in-house.

L'idea nasce dalla necessità personale e concreta di rendere più efficace e fruibile l'utilizzo di un media server "casalingo" **self-hosted** che offre servizi come Plex e/o Jellyfin per lo **streaming di contenuti multimediali** in rete locale: i fruitori dei suddetti servizi (amici, parenti) vorrebbero poter visualizzare un ampio catalogo di film, da cui selezionare titoli di loro interesse per la visione.

Se non già presente nel catalogo dell'home server, **gli utenti potranno richiedere il titolo di loro interesse** in modo che l'utente gestore del servizio possa adoperarsi per acquistarlo e/o noleggiarlo e renderlo disponibile per tutti, soddisfacendo la richiesta.

La libreria self-hosted si va così componendo di titoli espressamente voluti dai fruitori del servizio, a differenza dei più blasonati servizi streaming dove l'utente è un semplice "osservatore" e non ha potere decisionale sul catalogo proposto.

L'idea generale prende ispirazione da progetti preesistenti e diffusi nel mondo dei server self-hosted, come [Overseerr](#), cercando una reinterpretazione più personale e concreta, dall'**utilizzo immediato e semplice per tutti**, e ovviamente utilizzando il framework Django.

I dati utilizzati nel catalogo sono basati su parti del **DB ufficiale di TMDb**, opportunamente adattati alle necessità del servizio web che si va a creare.

N.B. Non si tratta, quindi, di implementare un servizio di streaming self hosted, quanto più di creare un servizio che permetta agli utenti di scoprire e scegliere titoli in base alle loro preferenze e richiederne la disponibilità su una piattaforma esterna e arbitraria.

Features rilevanti

- **Visualizzazione:** catalogo completo, possibilità di visualizzare dettagli (descrizione, tipologia, anno di produzione, ecc...) e copertina per ogni titolo.
- **Richieste di titoli:** gli utenti registrati possono richiedere film se marcati come non ancora disponibili sull'home server.
- **Scraping di immagini:** le immagini di copertina vengono ottenute direttamente da [TMDB](#) tramite richieste GET corredate da API key appositamente generata. Si evita quindi il salvataggio in locale di una grande quantità di immagini, con risparmio in termini di spazio di archiviazione.
- **Stato richiesta e dashboard:** ogni utente registrato ha accesso a una dashboard personale dove consultare le richieste passate e vedere il loro stato attuale (in attesa, accettato, rifiutato).
- **Watchlist:** gli utenti registrati possono inserire titoli (disponibili e non) in una watchlist, ossia una lista personale di titoli da guardare in futuro.
- **"Recently added":** visualizzazione degli ultimi titoli la cui richiesta è stata approvata e che sono quindi stati inseriti a catalogo.
- **Personalizzazione utente:** ogni utente può caricare un'immagine profilo a suo piacimento, oltre ad inserire altre informazioni personali.
- **Differenziazione utenti:** le features a cui ogni utente ha accesso sono determinate dal suo livello di autenticazione

Tipologie di utenti

- **anonimo** → può visualizzare il catalogo completo, vedere quali titoli sono già marcati come disponibili. Non può vedere le richieste.
- **registrato** → estende i privilegi dell'utente anonimo, possibilità di richiedere titoli in quanto membri e fruitori effettivi del servizio. Pagina personale personalizzabile con informazioni e foto profilo.
- **moderatore** → estende i privilegi dell'utente registrato, ha la possibilità di vagliare le richieste in attesa (accettare / rifiutare) provenienti da qualsiasi utente. Un utente viene inserito come moderatore in quanto addetto al mantenimento dei servizi self-hosted e con accesso ai canali di acquisizione dei titoli (ha, in sostanza, i mezzi e le capacità per soddisfare le richieste degli utilizzatori).
- **admin** → controllo completo, estende i privilegi del moderatore con la possibilità di inserire nuovi titoli a catalogo, creare ed eliminare utenti.