

■ React Project - Movie/Series Explorer (TVMaze Adapter)

Questo file contiene le istruzioni aggiornate per l'esercizio React e l'implementazione completa di un ****adapter**** per l'API pubblica di TVMaze. L'adapter semplifica le chiamate all'API e restituisce i dati in un formato coerente e facilmente utilizzabile nei componenti React.

■ API Assegnata: TVMaze

- Endpoint base: `https://api.tvmaze.com` - Ricerca per titolo: `/search/shows?q=` - Dettaglio per ID: `/shows/:id`

■ Codice Adapter (tvmazeAdapter.js)

```
// src/api/tvmazeAdapter.js
const API_URL = "https://api.tvmaze.com";

// ■ Ricerca show per titolo (no key richiesta)
export async function searchTitles(query, page = 1) {
  if (!query?.trim()) {
    return { items: [], totalResults: 0, totalPages: 0 };
  }
  const res = await fetch(`${API_URL}/search/shows?q=${encodeURIComponent(query)}`);
  if (!res.ok) throw new Error("Errore nella chiamata di ricerca");
  const data = await res.json(); // [{ score, show }, ...]

  // TVMaze non ha paginazione nativa su /search → gestiamo client-side
  const normalized = data.map(d => normalizeSearchItem(d.show));
  const pageSize = 10;
  const totalResults = normalized.length;
  const totalPages = Math.max(1, Math.ceil(totalResults / pageSize));
  const start = (page - 1) * pageSize;
  const items = normalized.slice(start, start + pageSize);

  return { items, totalResults, totalPages };
}

// ■ Dettaglio show per ID
export async function getTitleById(id) {
  const res = await fetch(`${API_URL}/shows/${id}`);
  if (!res.ok) throw new Error("Errore nel caricamento del dettaglio");
  const raw = await res.json();
  return normalizeDetailItem(raw);
}

// ■ Normalizzazioni (UI-friendly)
function normalizeSearchItem(show) {
  return {
    id: show.id,
    title: show.name,
    year: show.premiered ? show.premiered.slice(0, 4) : "-",
    type: show.type || "series",
    poster: show.image?.medium || show.image?.original || null,
    rating: show.rating?.average ?? null,
  };
}

function normalizeDetailItem(show) {
  return {
    id: show.id,
    title: show.name,
    year: show.premiered ? show.premiered.slice(0, 4) : "-",
    genres: show.genres || [],
  };
}
```

```
summaryHTML: show.summary || "",
poster: show.image?.original || show.image?.medium || null,
rating: show.rating?.average ?? null,
status: show.status,
language: show.language,
runtime: show.runtime,
premiered: show.premiered,
officialSite: show.officialSite || show.url,
};
}
```

■ Ora puoi importare `searchTitles` e `getTitleById` nei tuoi componenti React e concentrarti sulla UI.