

Prerequisiti tecnici:

Un ide per lo sviluppo web, vs code, phpstorm, sublime text

Node.js 18.17.0 o superiore (meglio 20 lts)

Package manager (npm o PnpM o yarn, in base a quello che e' installato o a disposizione)

Messaggio su slack per ricevere invito al repository

In questo workshop andremo a creare un app in Next Js con l'obiettivo finale di avere:

- Un'app in Next js v14 "plain" da riga di comando utilizzando il tool CLI di next.js
- Alcune Nozioni di base della struttura dei file in un'app Next con directory "pages"
- Una home page dove visualizzare la lista dei personaggi della serie Rick e Morty
- Una pagina di dettaglio del personaggio, che include nome, immagine, stato
- Per la lista dei personaggi, saranno implementate le seguenti features: filtraggio e paginazione stile "load more"

Andremo a realizzare un app generata staticamente che utilizza Next.js per il frontend e un api online che ne gestisce i contenuti.

Con che stack tecnologico verrà costruita l'app:

Next.js: Il framework react per il web

React: Una libreria JavaScript per costruire interfacce utente

Tailwind CSS: Un framework CSS "utility-first"

Vercel: deploy, gestione certificati e pipeline di sviluppo

Conoscenze da approfondire per la piena comprensione del workshop:

- Comprensione di base di HTML, CSS e JavaScript
- Familiarità con React, Next Js è costruito e si basa "on top" su React
- Conoscenza generica del framework Tailwind Css
- Conoscenza generica di una API graphql (<https://graphql.org> api) e come funziona una richiesta e una risposta HTTP per comunicare dati

Comprensione del *DOM virtuale*: React utilizza un DOM virtuale, che è diverso dal DOM effettivo di una pagina web.

Conoscenza di base della *sintassi HTML* per creare la struttura e il contenuto di una pagina web.

Conoscenza di base della *sintassi CSS*, utilizzata per definire l'aspetto e la presentazione di una pagina web.

Per quanto riguarda React:

Componenti: i componenti sono gli elementi costitutivi di un'applicazione React. Sono parti di codice riutilizzabile che possono essere composti per creare interfacce utente complesse.

Props: Le props (o "properties"), vengono utilizzate per passare i dati da un componente all'altro. Sono come argomenti per una funzione e possono essere utilizzati per personalizzare il comportamento e l'aspetto di un componente.

State: gli stati vengono utilizzati per archiviare dati che possono cambiare nel tempo. Viene utilizzato per gestire il comportamento dinamico di un componente.

Step e sviluppi successivi?

Questi sono solo alcuni possibili approfondimenti e focus per continuare e portare avanti lo sviluppo dell'app costruita:

- Ricerca di un personaggio per nome
- Approfondimento e focus del deploy dell'app (netlify o vercel) per il goLive
- Implementazione di un tool per la gestione asincrona degli stati (react-query, swr)
- Generazione di pagine server side, andando a introdurre stati per il caricamento e placeholder skeleton, per andare a migliorare l'esperienza utente.
- Generazione di pagine usando l'Incremental Static Regeneration (ISR)
- Integrazione di una libreria per gestire e arricchire l'interfaccia con animazioni e transizioni di pagina (framer motion, motion one, gsap)