MIX

Bangalore TCS RMG:
POC from Bangalore Reginol RMG:
A SivaKumar(a.sivakumar@tcs.com)

Unallocated SWON #(1008420)

As a fresher, I am very enthusiastic when it comes to learning new technologies and working on them. I may not have full-time experience as a developer but I am eager to learn and grow with this company. In college, I was a part of a startup company where I worked as a web developer for 10 months. In this role, I learnt and contributed in many technologies like Node JS, Express, HTML, CSS, Javascript, MongoDB, AWS - EC2, S3, SNS, Nginx. I was also involved in the hiring process where I Shortlisted around 100 people and interviewed more than dozen and hired 3 interns

Hello,
My name is Vipul Kumar. Currently, I am in Bangalore, India.
I prefer Java over other programming languages because it's secure and supports the OOPS concept.
In college, I was a part of a startup company where I worked as a web developer for 10 months. In this role, I learnt and contributed in many technologies like Node JS, Express, HTML, CSS, Javascript, MongoDB, AWS - EC2, S3, SNS, Nginx. I was also involved in the hiring process where I Shortlisted around 100 people and interviewed more than dozen and hired 3 interns.
Currently, I am learning DevOps tools like Docker, Kubernetes, Ansible, CI/CD, etc.

Hi, I am Vipul. Currently, I am a System Engineer working for TCS digital Bangalore, India. Here I work for Apple on an R&D project. I want to switch due to some personal issues. I have working experience of more than a year. In my present and previous companies, I learned and contributed to many technologies like Node JS, Express, HTML, CSS, React, Javascript, MongoDB, AWS - EC2, S3, SNS, Nginx, Docker, Kubernetes. I was also involved in the hiring process where I Shortlisted around 100 people and interviewed more than dozen and hired 3 interns. I am open to new technologies and challenges.
I know I don't have much experience but I am sure that I can handle any challenge.

Thanks & Regards,
Vipul kumar
vipulkumar4321@gmail.com
+918765263553

Backstage

What is Backstage?

- Backstage is a single-page application composed of a set of plugins.
- Developed by the Spotify engineering team, It is an open-source platform used for building developer portals and managing software delivery.
- Backstage gives developers a uniform overview of all their resources, regardless of how and where they are running.
- It also allows the creation of new resources with a few clicks of a button.
- The idea behind Backstage is that it helps reduce the cognitive load on a new developer by pulling together commonly required resources into one browser-based user interface.

Why we need Backstage? Problem.

- As companies grow, their infrastructure systems get messier. So many technologies come into play like Docker Kubernetes, GCP/AWS/Azure and much, much more.
- So, it is hard to mange and maintaine all these infrastructures.

How we can fix the issue? Solution.

- Backstage unifies all our infrastructure tooling, services, and documentation with a single, consistent UI.
- Imagine if all our tools —like Docker Kubernetes, GCP/AWS/Azure and ,whatever else is hiding in our stack — all had the same, easy-to-use interface. That's Backstage.One frontend for all our infrastructure.

History:

The initial alpha version was released on 16 March 2020.
Latest Version: 0.7.16

Tech-stack:

- Node.js: Backstage runs on Node.js or we can say Back-end of Backstage is Node.js
- TypeScript: Backstage is mostly written in TypeScript, though we can code in pure JavaScript.
- React: The frontend code is written using React. React components play a fundamental role in Backstage's plugin architecture. Plugins are essentially individually packaged React components
- Yarn : Yarn is a package manager like NPM.
- Lerna: Lerna is a tool that optimizes multi-package repositories with git and npm.

Getting started with Backstage:

- There are two different ways to get started with Backstage:
    1. Recommended: Create a standalone app
    2. Contributors: Clone the Backstage repository from github
- Creating a standalone app makes it simpler to customize the application for our needs and stay up to date with the project.

Requirement for standalone app:

- Node.js(atleast version 14)
- NPM & YARN

Creation of Backstage app:

- install node, npm, yarn
- npx @backstage/create-app

  ? Enter a name for the app [required] test-app

  ? Select database for the backend [required] SQLite or PLSQL
- yarn dev   //Our App will be running on localhost:3000/

What is Plugin:

- A plugin is a piece of software that acts as an add-on to a web browser and gives the browser additional functionality.

What is Backstage Plugin:

- A Backstage Plugin adds functionality to Backstage.

Creation of Backstage plugin:

- yarn create-plugin

  ? Enter an ID for the plugin [required] test-plugin

Who else uses backstage:

- Spotify, Netflix, fiverr, Expedia Group, American Express and many more have chosen Backstage for their internal developer portal.
- https://github.com/backstage/backstage/blob/master/ADOPTERS.md

How to onboard our own project into Backstage?

- Backstage requires a YAML file from source control System like github, bitbucket, etc..
- And the project must contain a YAML file if we want to onboard that into backstage.
- And also That YAML file must be in a specific format. Given in backstage.

APIs have a visibility: they are either public (making them available for any other component to consume), restricted (only available to a whitelisted set of consumers), or private (only available within their system). As public APIs are going to be the primary way interaction between components, Backstage supports documenting, indexing and searching all APIs so we can browse them as developers.

Backstage natively supports importing catalog data through the use of entity descriptor YAML files.

The catalog has a frontend plugin part, that communicates via a service API to the backend plugin part. The backend has a processing loop that repeatedly ingests data from the sources you specify, to store them in its database.

Service overwiew page.

this page list all the services that we own.

https://appleinc.webex.com/appleinc/j.php?MTID=m20e5cd9da9b177e7993e65c39cb9b84e

A developer portal is where we publish our API's documentation. And is a dashboard for our API products.

TCS developer portal.
https://uxapps.ultimatix.net/#/developer/mFrameios

A developer portal is where we publish our API's documentation. And is a dashboard for our API products.
So plugins help us to achive this goal.

TCS developer portal.
https://uxapps.ultimatix.net/#/developer/mFrameios

Backstage is designed to be a developer portal for all our infrastructure tooling, services, and documentation.

the Software Catalog enables two main use-cases:

1. Helping teams manage and maintain the software they own. Teams get a uniform view of all their software; services, libraries, websites, ML models — you name it, Backstage knows all about it.
2. Makes all the software in your company, and who owns it, discoverable. No more orphan software hiding in the dark corners of your software ecosystem.

It is important to note that any kind of software can be registered in Backstage. Even if the software is not maintained by your company (SaaS offering, for example) it is still useful to create components for tracking ownership.

API integration is in under development.