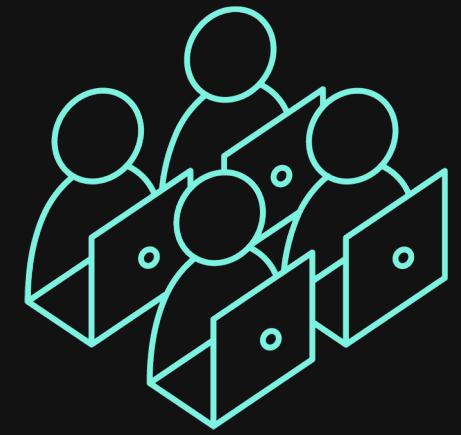
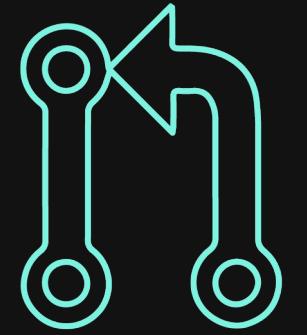


The Backstage community



2.2K+
adopters



1.2K+
contributors



24K
stars



11K+
members



4.5K+
project forks

Is Backstage for me?

Jorge Lainfiesta • KCD Denmark 2023

Jorge Lainfiesta

- Technical Product Marketing Manager @ Spotify
- Author of the Linux Foundation Introduction to Backstage course
- Digital communication @ AAU CPH





Photo by Godwin Omeri Okoro on [Unsplash](#)

Tadarida brasiliensis



Nycticeinops schleffenii



- Lives in large communities: up to 2 million!
- Much larger call recognition records to find its offspring.

- Lives alone with its offspring

Social complexity hypothesis

More complex social systems
require more complex
communication systems.

**Engineering organizations are
complex technical societies.**

Is Backstage for me?

**Which 🚀 do you need to
express in your platform?**



Do we have a service/API/library for X?

How do I use it or extend it?

- Pings channels and people in Slack
- – Helena logged all the services we had in 2021 maybe? Check it out it's on a table in Jira – We don't use Jira??
- Visits 3 outdated Confluence pages
- Last commit: 2 years ago

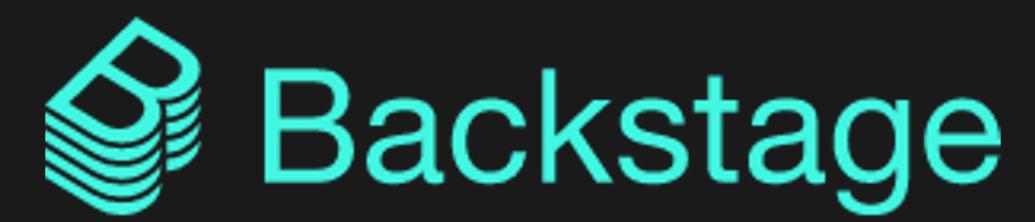


Something is failing, who can help me with Y?

If Y fails, what else is going to fail?

- Email chain with growing number of CC'ed people
- “Ah yes I think Alessio did that before he left”
- Zoom meeting with most people on mute and no camera
- README.md? I barely know her

**The team is not “bad” at communicating:
its dynamics are too complex for the
available communication systems.**



Catalog

Enables discoverability by tracking ownership and dependencies

The screenshot shows the Backstage Catalog interface. On the left, there is a sidebar with various links: Home, APIs, Docs, Explore, Tech Radar, Cost Insights, and GraphQL. Below these are sections for PERSONAL (Owned: 3, Starred: 0), BACKSTAGE (All: 15), OWNER, LIFECYCLE, TAGS, and PROCESSING STATUS. The main area is titled "Backstage Catalog" and displays a table titled "All components (15)". The table has columns for NAME, SYSTEM, OWNER, TYPE, LIFECYCLE, DESCRIPTION, and TA. The components listed include artist-lookup, backstage, backstage-demo, petstore, playback-order, playback-sdk, podcast-api, queue-proxy, searcher, shuffle-api, wayback-archive, wayback-archive-ingestion, and wayback-archive-storage.

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TA
artist-lookup	artist-engagement-portal	team-a	service	experimental	Artist Lookup	jd
backstage		cncf	library	experimental	Backstage is an...	
backstage-demo		backstage/mainainers	website	experimental	An example...	
petstore		Team C	service	experimental	[The...	
playback-order	audio-playback	Guest User	service	production	Playback Order	jp
playback-sdk	audio-playback	Team C	library	experimental	Audio and video...	
podcast-api	podcast	Team B	service	experimental	Podcast API	ja
queue-proxy	podcast	Team B	website	production	Queue Proxy	g w
searcher		Guest User	service	production	Searcher	g
shuffle-api	audio-playback	Guest User	service	production	Shuffle API	g
wayback-archive		team-a	service	production	Archive of the...	
wayback-archive-ingestion		Team D	service	production	Ingestion...	
wayback-archive-storage		team-a	service	production	Storage...	

Catalog entity

Can express services/groups/etc and showcase its multiple dimensions

The screenshot shows the Backstage service catalog interface. At the top, it displays the component details for 'artist-lookup'. The 'About' section includes links to 'VIEW SOURCE' and 'VIEW TECHDOCS'. Below this, the component's metadata is listed: OWNER 'team-a', SYSTEM 'artist-engagement-portal', TYPE 'service', LIFECYCLE 'experimental', and TAGS 'java', 'data'. To the right, a 'Relations' panel shows a graph of dependencies. Nodes include 'component:www-artist', 'group:team-a', and 'system:artist-engagement-portal', connected by edges labeled 'consumesApi / apiConsumedBy', 'ownerOf / ownedBy', and 'dependsOn / depend'. At the bottom, sections for 'Links' and 'Has subcomponents' are shown, both currently empty.

COMPONENT — SERVICE
artist-lookup ☆

OVERVIEW CI/CD API DEPENDENCIES DOCS TODOS

About

VIEW SOURCE VIEW TECHDOCS

DESCRIPTION

Artist Lookup

OWNER team-a

SYSTEM artist-engagement-portal

TYPE service

LIFECYCLE experimental

TAGS java, data

Relations

View graph →

Links

- Examples Users
- Link with Cloud Icon
- Support
- Alerts
- Example Group
- Dashboard
- Website

Has subcomponents

NAME	OWNER	TYPE	LIFECYCLE	DESCRIPTION
No subcomponent is part of this component Learn how to change this.				

Backstage is a framework

Make it work for you

- Every organization is unique: Backstage lets you express exactly that.
- Who benefits from the Catalog and how?
- When building the Catalog: minimize friction and play by your culture
- If it doesn't bring value, ditch it!



TECH NOIR



We'll hire 5 engineers next Q, how do we onboard them?

What's the path to production?

New hire tries but can't figure out what tech to focus on

Spends a week poking a system they thought was relevant - turns out it was deprecated tacitly last year.

– Why are you using AngularJS here? – That's what the Welcome tutorial said.

Manager and teammates too busy with a production incident and an imminent product lunch.

Software Templates

Offers Golden paths and self-service affordances.

The screenshot shows the Backstage application's interface. On the left is a dark sidebar with white text and icons for Backstage, Search, Explore, Manage, Experiment, Data, ML, Docs, Create, More, and Add Shortcuts. At the bottom of the sidebar is a button labeled 'Unpin Sidebar' with a left arrow icon. The main area is titled 'Create...' and features a search bar with a magnifying glass icon and a clear 'x' button. Below the search bar is a section titled 'Backend Service #backend-support'. It contains three template cards:

- Backend Service Blueprint** by **web-infra**: Labeled as a 'Golden Path'. Description: Backend service Blueprint based on [Apollo Standalone](#). Usually the best choice for consumer-facing services and APIs. Includes support for gRPC, HTTP, deployment with Kubernetes, logging, metrics and Envoy sidecar (with CNAME & Edge Proxy) for transcoding. A 'CHOOSE' button with a GitHub icon is at the bottom.
- Apollo Service** by **weaver**: Description: Runnable backend service template based on [Apollo Standalone](#). Usually the best choice for consumer-facing services and APIs. Includes support for gRPC, HTTP, deployment with Kubernetes, logging, metrics and Envoy sidecar for transcoding. A 'CHOOSE' button with a GitHub icon is at the bottom.
- Java Library** by **weaver**: Labeled as 'Experimental'. Description: Java library template following the [Java Library Best Practices](#). Use this for creating a new repository for your library. If you want to add a library to an existing repository, consider using the [java-library-archetype](#) instead. A 'CHOOSE' button with a GitHub icon is at the bottom.

At the very bottom of the main content area, there are three additional cards partially visible:

- mcf** by **Service Registry**: Overview: Service Registry (backstage)
- fs-integrations** by **File System Integrations**: Overview: File System Integrations (backstage)
- python** by **Python Library/CLI**: Overview: Python Library/CLI (backstage)



We need to promote Z new practice

How many people have migrated?

- Carefully creates example repo with new practice – nobody sees it.
- “Sorry we missed your @here, my team is too busy rn we’ll create a ticket for your migration thingy”
- Manually checks repos for changes – gives up after 25
- Sends out email survey with a 10% response rate

SoundCheck

Tracks aggregated information from the Catalog.

The screenshot shows the Backstage application interface. On the left is a sidebar with the following items:

- Backstage logo
- Search bar
- Home
- APIs
- Docs
- Create...
- Tech Radar
- Skill Exchange
- Pulse
- RBAC
- Soundcheck (highlighted)
- Insights
- Settings

The main content area is titled "COMPONENT – SERVICE" and "awesome-component ☆". It has tabs for OVERVIEW, CI/CD, SOUNDCHECK (which is active), API, and DEPENDENCIES. The SOUNDCHECK tab displays the following information:

L1 TEST CERTIFIED

L1 Test Certified
Improve quality and reliability of your software component by measuring the use of testing best practices.
[Learn more](#)

LEVEL 1

✓ Tests run on CI	1 minute ago	
✓ Coverage is known	1 minute ago	
✓ Static analysis performed	now	
✓ SLIs are defined	now	

LEVEL 2

✗ Tests passed on CI	now	
----------------------	-----	--

Tests passed on CI

Check did not pass

Description
This check validates both that you have tests and that the number of tests run is greater than zero. The number of tests run is determined by using the JUnit test count metric.

Be critical

There are no magic solutions

- Standardizing practices and metrics requires ample consensus, a long term vision, and commitment.
- The templates and quality metrics must be based on battle-tested practices that work in *your* organization.
- Templates and metrics must be regularly revisited and updated.

Is Backstage for me?

- Adopting Backstage is not about numbers (number of services, number of developers) but the complexity of your system and what you want to communicate.
- Backstage is not magic: it won't solve your Discoverability or create self-service workflows for you. Any Developer Portal needs you to put work into making it fit your teams' tools and dynamics.

	Discoverability	Ownership	Onboarding	Migrations	Standardization	Security & Compliance	Streamline workflows
Catalog	X	X	X	X		X	X
TechDocs	X		X		X		X
API Docs	X		X				X
Search	X		X				
Scaffolder		X	X	X	X	X	X
Integrations	X	X	X	X	X	X	X
SoundCheck		X		X	X	X	

Jorge Lainfiesta

linkedin.com/in/jrlainfiesta/



 Backstage



Feedback