



www.avalia.io

An introduction to Backstage

An open source platform for building developer portals

CLOUD NATIVE BERN MEETUP
NOVEMBER 17th 2022



CONFIDENTIAL



Data-driven approach to drive continuous improvement

A new approach to assess the impact of software products, processes, and people in the business

DATA



ANALYSIS



ROADMAP



The Avalia Platform collects data from software development systems such as code version control (Git) and project management tools (Jira).

Our experts, through industry frameworks and computational methods such as machine learning, analyse the data and dive deep into the key issues.

Our team builds a bespoke roadmap, planning actions and acceptance criteria to drive systems from their “as-is” to a desired “to-be” state.

Empower development teams



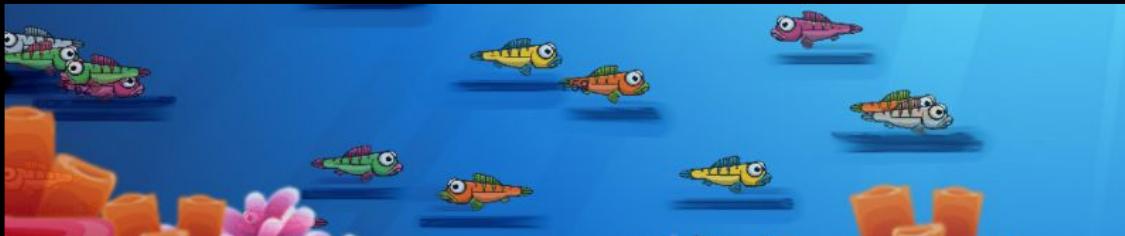
Workshop

Situation vs goals.



Training & coaching

Enable teams to act
TDD, BDD, metrics.



Gamification

Trigger start of transformation.

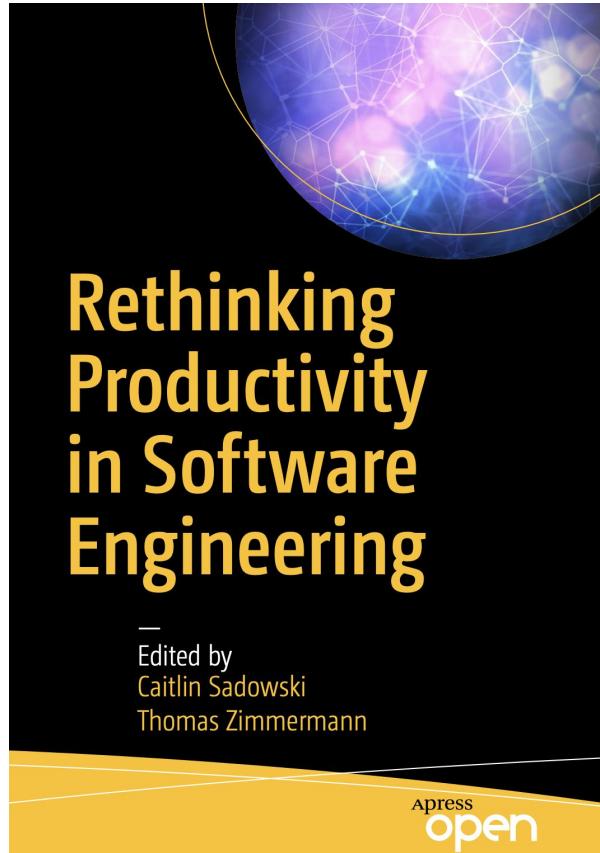
Celebrate success!



https://www.youtube.com/watch?v=vVyHghC_4IM

CONFIDENTIAL

Productivity in software engineering



Part I: Measuring Productivity: No Silver Bullet

- There is **no single productivity metric** for software engineers.
- Instead, focus on **a set of custom metrics** targeted to a **specific question**.
- Improving productivity requires explaining the factors that affect it, but that requires **qualitative insights** into team behavior.
- Managers are best positioned to get these qualitative insights by **interacting** with their team.

Part II: Introduction to Productivity

Part III: The Context of Productivity

Part IV: Measuring Productivity in Practice

Part V: Best Practices for Productivity

- There are several different types of preventable "**wastes**" that occur during software development and represent lost productivity.
- **While it may be hard to define and measure productivity, identifying/reducing waste is an effective way to become more productive.**

Developer eXperience: the key to boost productivity

JAMES GOVERNOR'S MONKCHIPS

What is Developer Experience? a roundup of links and goodness

By James Governor | @monkchips | February 21, 2022



Developer Experience is about creating an environment in which a developer can do their best work.

DX is the context in which developer productivity can be unleashed, in which individual needs can be successfully balanced with those of the engineering team.

DX is about developer feelings – **it is a sociotechnical system which should consider every touchpoint a developer interacts with to plan and produce software**, from learning a platform to the very first line of code all the way through its promotion into production. From documentation to SDKs to version control to monitoring and observability, **everything a developer touches will have an impact on their productivity**.

Culture too, helps to define a good DX of the lack of it.

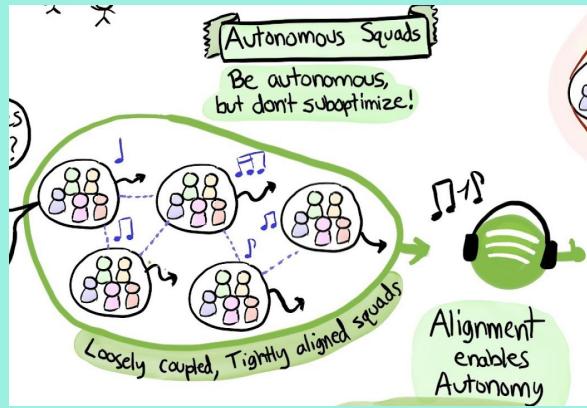
Backstage is an open platform for building developer portals.

Created at



Donated to

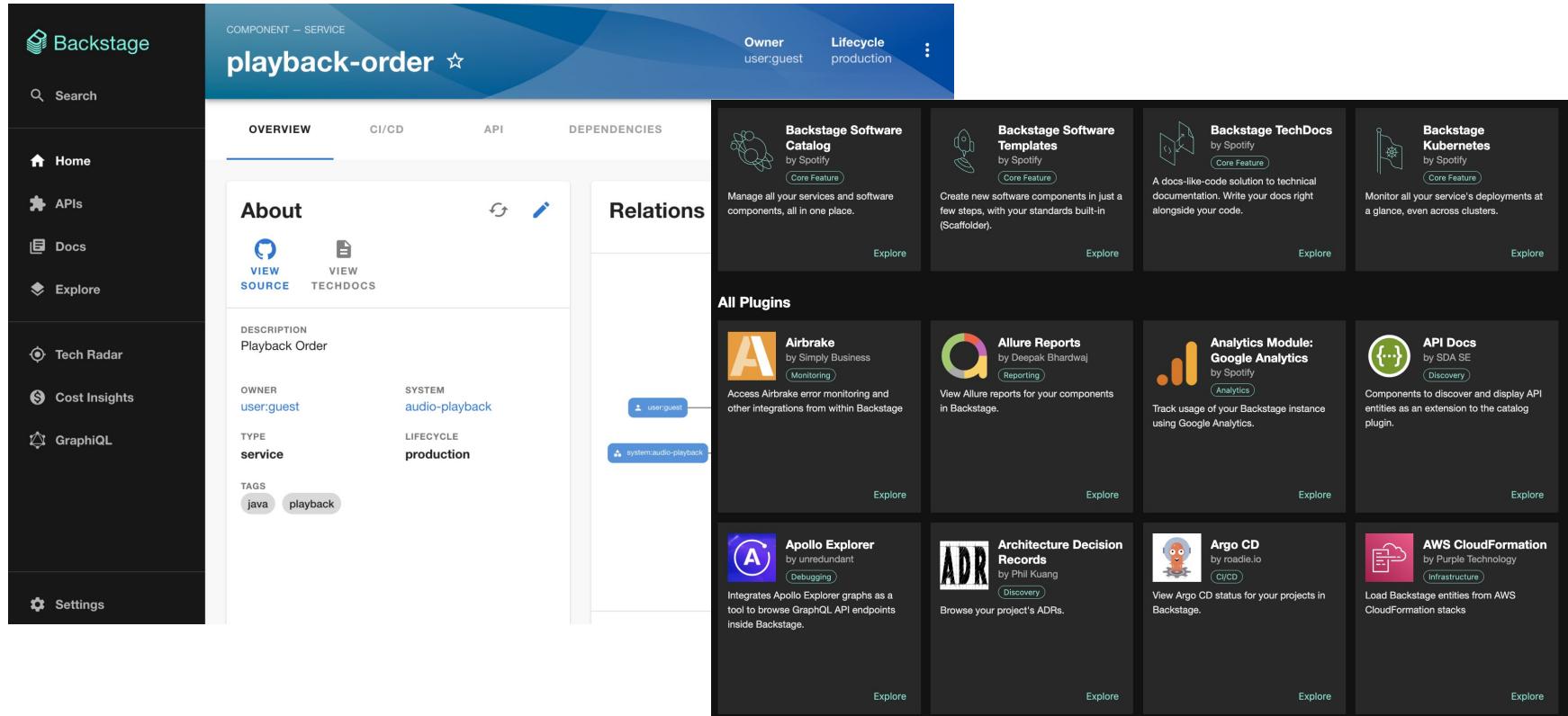




Backstage: the entrypoint into your software ecosystem



Backstage: a platform, with a growing ecosystem of plugins



COMPONENT – SERVICE

playback-order ☆

Owner: user:guest | Lifecycle: production | ...

OVERVIEW CI/CD API DEPENDENCIES

About

Relations

All Plugins

- Backstage Software Catalog** by Spotify (Core Feature) Manage all your services and software components, all in one place. Explore
- Backstage Software Templates** by Spotify (Core Feature) Create new software components in just a few steps, with your standards built-in (Scaffolder). Explore
- Backstage TechDocs** by Spotify (Core Feature) A docs-like-code solution to technical documentation. Write your docs right alongside your code. Explore
- Backstage Kubernetes** by Spotify (Core Feature) Monitor all your service's deployments at a glance, even across clusters. Explore
- Airbrake** by Simply Business (Monitoring) Access Airbrake error monitoring and other integrations from within Backstage. Explore
- Allure Reports** by Deepak Bhardwaj (Reporting) View Allure reports for your components in Backstage. Explore
- Analytics Module: Google Analytics** by Spotify (Analytics) Track usage of your Backstage instance using Google Analytics. Explore
- API Docs** by SDA SE (Discovery) Components to discover and display API entities as an extension to the catalog plugin. Explore
- Apollo Explorer** by unreduntant (Debugging) Integrates Apollo Explorer graphs as a tool to browse GraphQL API endpoints inside Backstage. Explore
- Architecture Decision Records** by Phil Kuang (Discovery) Browse your project's ADRs. Explore
- Argo CD** by roadie.io (CI/CD) View Argo CD status for your projects in Backstage. Explore
- AWS CloudFormation** by Purple Technology (Infrastructure) Load Backstage entities from AWS CloudFormation stacks. Explore

CONFIDENTIAL

Backstage: core features

Search



- Entities
- System model
- Ingestion
- YAML descriptors

- Doc as Code
- Diagrams as Code
- Publication process

- Templating
- Custom actions

- Integration in your ecosystem
- Front-end
- Back-end

Backstage Demo (out-of-the-box experience)



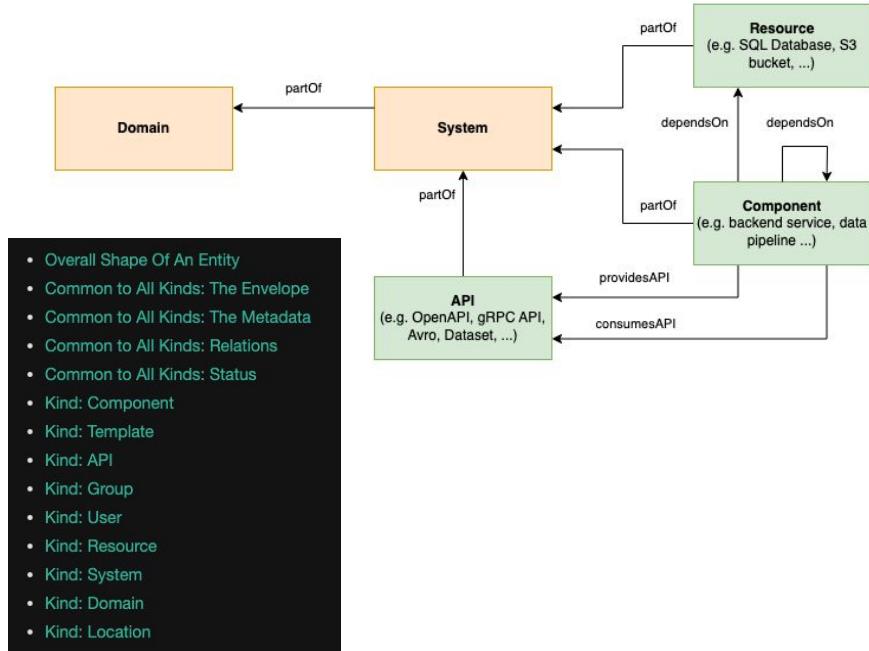
The screenshot shows the Backstage Catalog interface. The left sidebar contains navigation links: Home, APIs, Docs, Explore, Tech Radar, Cost Insights, and GraphQL. The main header is "Backstage Catalog". Below it, a section titled "Components" is shown with a dropdown menu. The main content area displays a table titled "Owned (3)" with three rows of data:

NAME	SYSTEM	OWNER	TYPE	LIFECYCLE	DESCRIPTION	TAGS	ACTIONS
playback-order	audio-playback	user:guest	service	production	Playback Order	java playback	
searcher		user:guest	service	production	Searcher	go	
shuffle-api	audio-playback	user:guest	service	production	Shuffle API	go	

Below the table are several filter dropdowns: TYPE (All), PERSONAL (Owned: 3, Starred: 1), BACKSTAGE (All: 15), OWNER, LIFECYCLE, and TAGS. A "Filter" button is located at the top right of the table.

CONFIDENTIAL

System Model: describe entities and relationships



```

apiVersion: backstage.io/v1alpha1
kind: Component
metadata:
  name: artist-web
  description: The place to be, for great artists
  labels:
    example.com/custom: custom_label_value
  annotations:
    example.com/service-discovery: artistweb
    circlegci.com/project-slug: github/example-org/artist-website
  tags:
    - java
  links:
    - url: https://admin.example-org.com
      title: Admin Dashboard
      icon: dashboard
      type: admin-dashboard
spec:
  type: website
  lifecycle: production
  owner: artist-relations-team
  system: public-websites
  
```

<https://backstage.io/docs/features/software-catalog/system-model>

<https://backstage.io/docs/features/software-catalog/descriptor-format>

Getting started...



Getting started...?

```
$ docker run spotify/backstage
```

```
Unable to find image 'spotify/backstage:latest' locally
docker: Error response from daemon: pull access denied for spotify/backstage, repository does not exist
```

Backstage is **not an application** that you run.

Backstage is an **application framework** that you use to create applications.

Getting started with Backstage - create your “instance”

```
$ npx @backstage/create-app
```

Need to install the following packages:

 @backstage/create-app

Ok to proceed? (y) y

```
? Enter a name for the app [required] my-portal
```

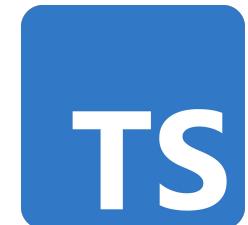
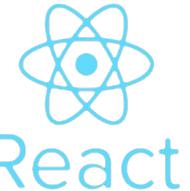
Creating the app...

Checking if the directory is available:

 checking my-portal ✓

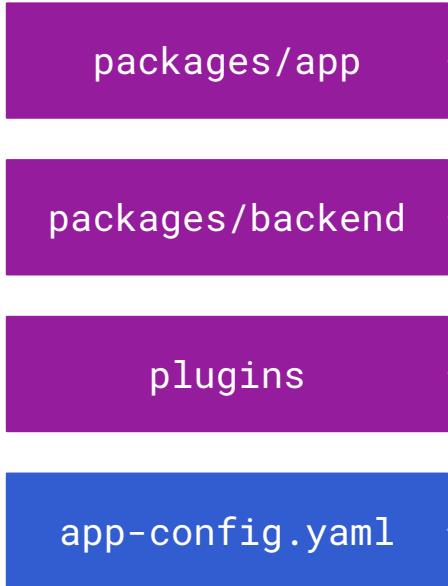
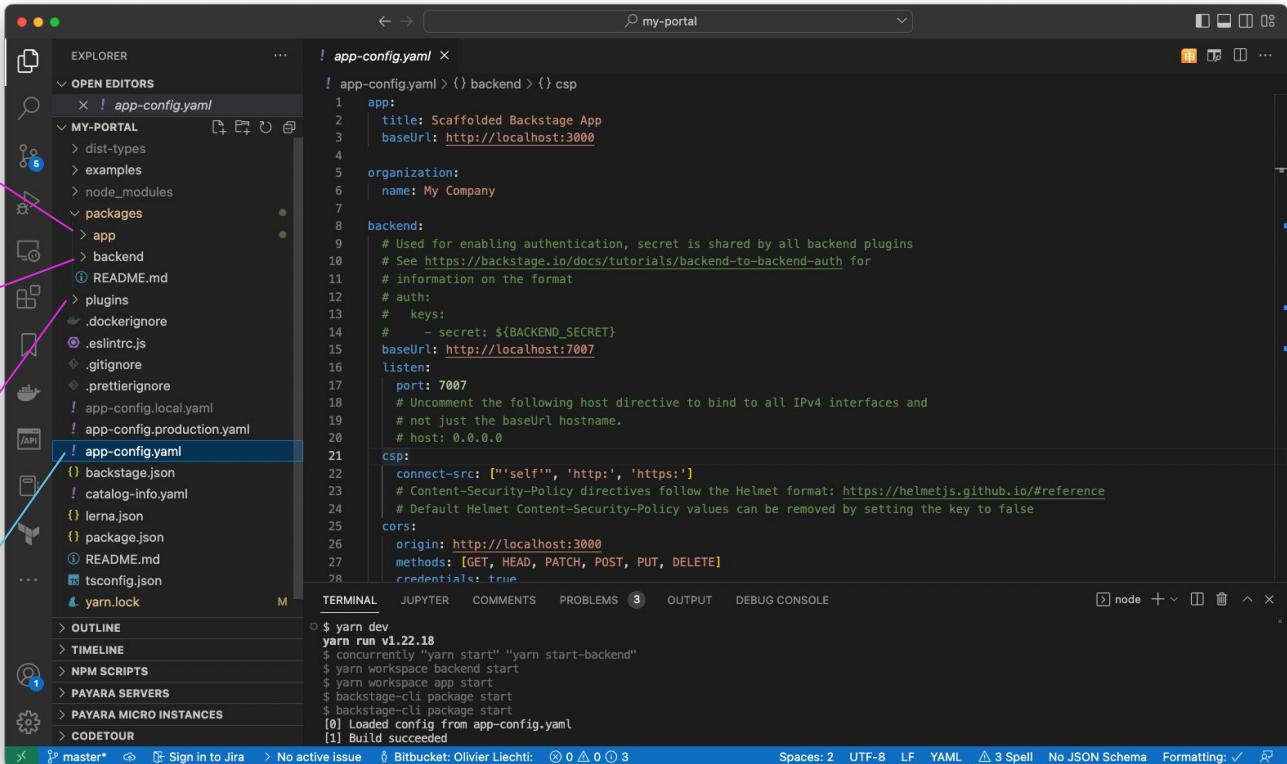
...

🥇 Successfully created my-portal
Run the app: `cd my-portal && yarn dev`



Lerna

To customize your instance, you work in the scaffolded code

The screenshot shows a VS Code interface with the following details:

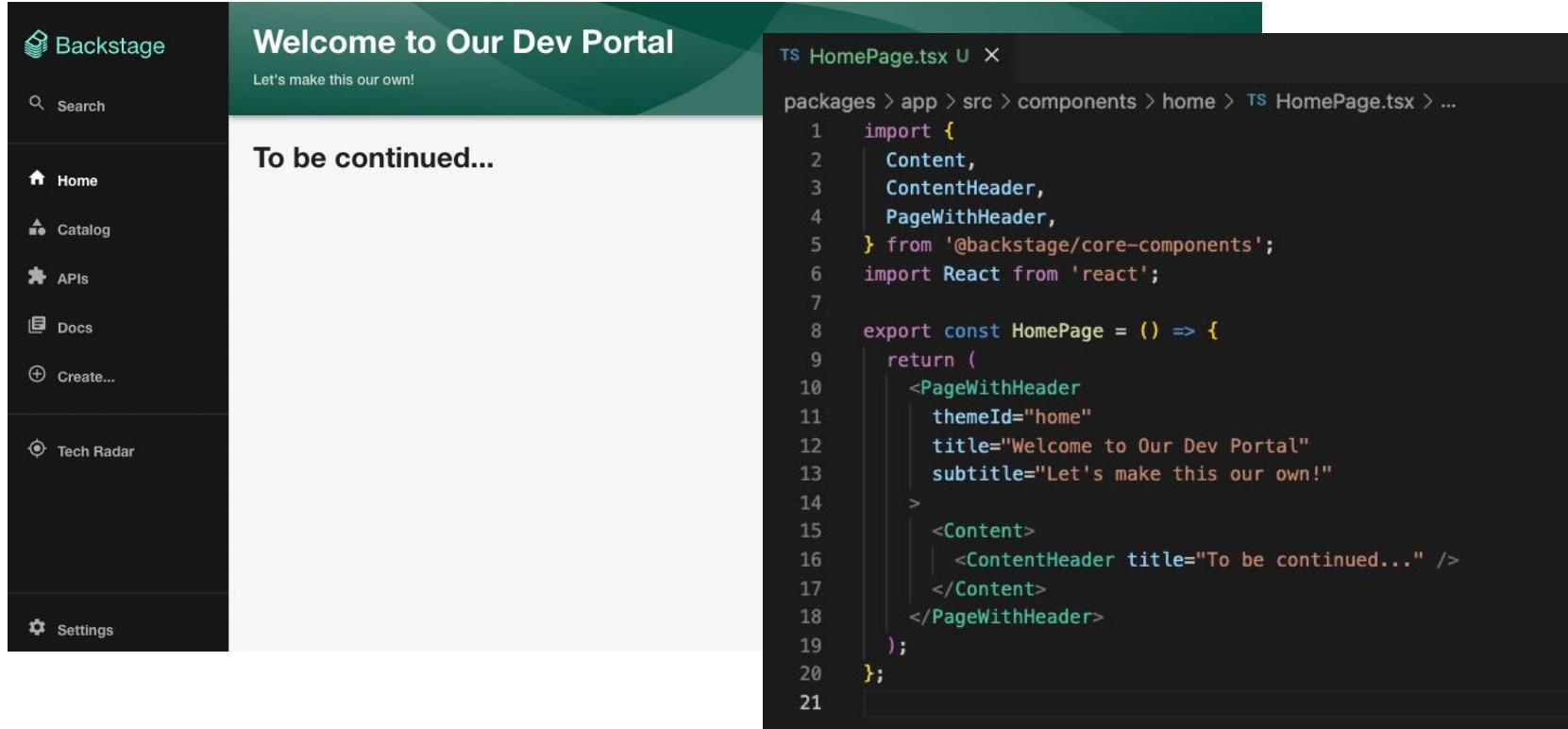
- File Structure (Explorer):**
 - MY-PORTAL
 - > dist-types
 - > examples
 - > node_modules
 - > packages
 - > app
 - > backend
 - ! README.md
 - > plugins
 - .dockerignore
 - .eslintrc.js
 - .gitignore
 - .prettierignore
 - ! app-config.local.yaml
 - ! app-config.production.yaml
 - Code Editor (app-config.yaml):**

```

! app-config.yaml <-- my-portal

! app-config.yaml > {} backend > {} csp
1 app:
2   title: Scaffolded Backstage App
3   baseUrl: http://localhost:3000
4
5 organization:
6   name: My Company
7
8 backend:
9   # Used for enabling authentication, secret is shared by all backend plugins
10  # See https://backstage.io/docs/tutorials/backend-to-backend-auth for
11  # information on the format
12  # auth:
13  #   keys:
14  #     - secret: ${BACKEND_SECRET}
15  baseUrl: http://localhost:7007
16 listen:
17   port: 7007
18   # Uncomment the following host directive to bind to all IPv4 interfaces and
19   # not just the baseUrl hostname.
20   # host: 0.0.0.0
21 csp:
22   connect-src: ['"self"', 'http:', 'https:']
23   # Content-Security-Policy directives follow the Helmet format: https://helmetjs.github.io/#reference
24   # Default Helmet Content-Security-Policy values can be removed by setting the key to false
25 cors:
26   origin: http://localhost:3000
27   methods: [GET, HEAD, PATCH, POST, PUT, DELETE]
28   credentials: true
    
```
 - Terminal:**
 - \$ yarn run v1.22.18
 - \$ concurrently "yarn start" "yarn start-backend"
 - \$ yarn workspace backend start
 - \$ yarn workspace app start
 - \$ backstage-cli package start
 - \$ backstage-cli package start
 - [0] Loaded config from app-config.yaml
 - [1] Build succeeded
 - Bottom Status Bar:**
 - Spaces: 2
 - UTF-8
 - LF
 - YAML
 - 3 Spell
 - No JSON Schema
 - Formatting: ✓

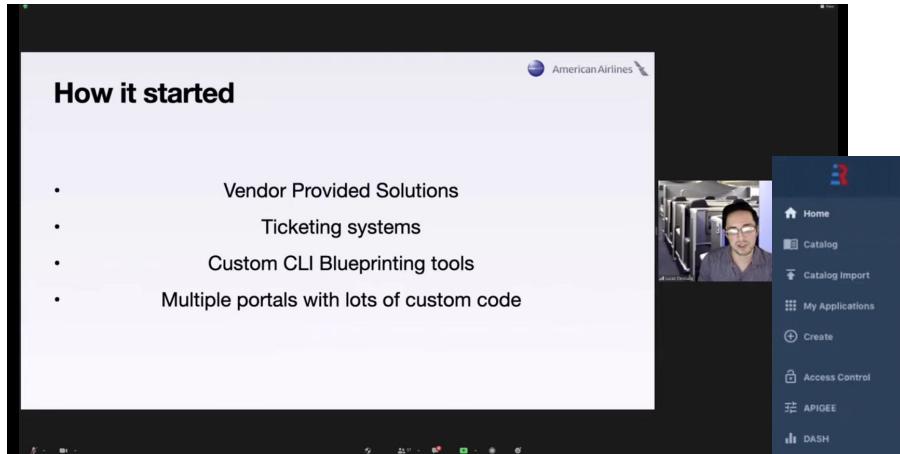
To customize the frontend, you use React components and plugins



The image shows a screenshot of the Backstage Dev Portal. On the left is a dark sidebar with various navigation items: Backstage (selected), Search, Home, Catalog, APIs, Docs, Create..., Tech Radar, and Settings. The main content area has a teal header bar with the title "Welcome to Our Dev Portal" and a subtitle "Let's make this our own!". Below the header, the page content reads "To be continued...". To the right of the content, a code editor window displays the source code for `HomePage.tsx`. The code imports `Content`, `ContentHeader`, and `PageWithHeader` from the backstage/core-components package, and `React` from react. It defines a `HomePage` component that returns a `PageWithHeader` component with the themeId "home", title "Welcome to Our Dev Portal", and subtitle "Let's make this our own!". Inside the `PageWithHeader`, there is a `Content` component containing a `ContentHeader` with the title "To be continued...".

```
TS HomePage.tsx U X
packages > app > src > components > home > TS HomePage.tsx > ...
1 import {
2   Content,
3   ContentHeader,
4   PageWithHeader,
5 } from '@backstage/core-components';
6 import React from 'react';
7
8 export const HomePage = () => {
9   return (
10     <PageWithHeader
11       themeId="home"
12       title="Welcome to Our Dev Portal"
13       subtitle="Let's make this our own!"
14     >
15     <Content>
16       <ContentHeader title="To be continued..." />
17     </Content>
18   </PageWithHeader>
19 );
20
21 
```

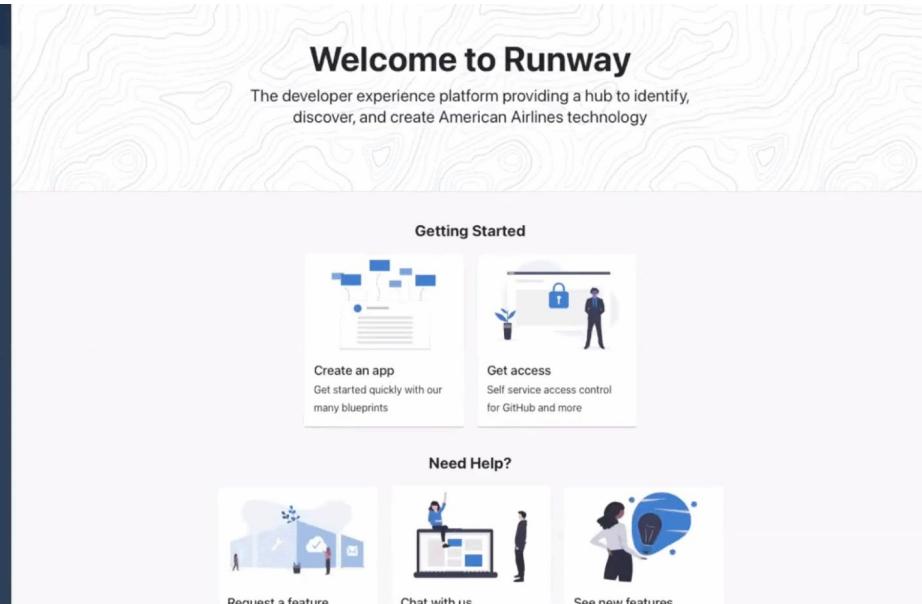
Runway: Backstage at American Airlines



How it started

- Vendor Provided Solutions
- Ticketing systems
- Custom CLI Blueprinting tools
- Multiple portals with lots of custom code

A video call window shows a person in a blue shirt and glasses, with the text "Meet Runway" visible.



Welcome to Runway
The developer experience platform providing a hub to identify, discover, and create American Airlines technology

Getting Started

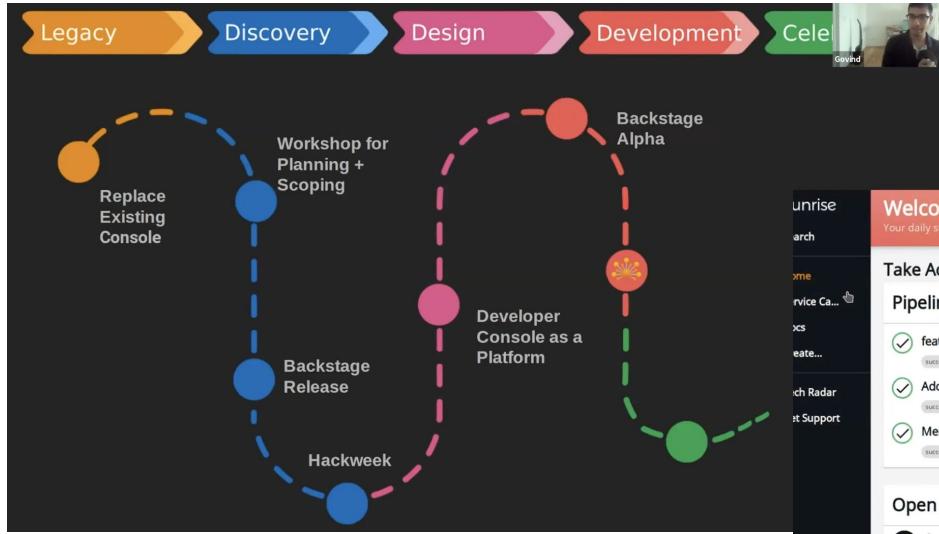
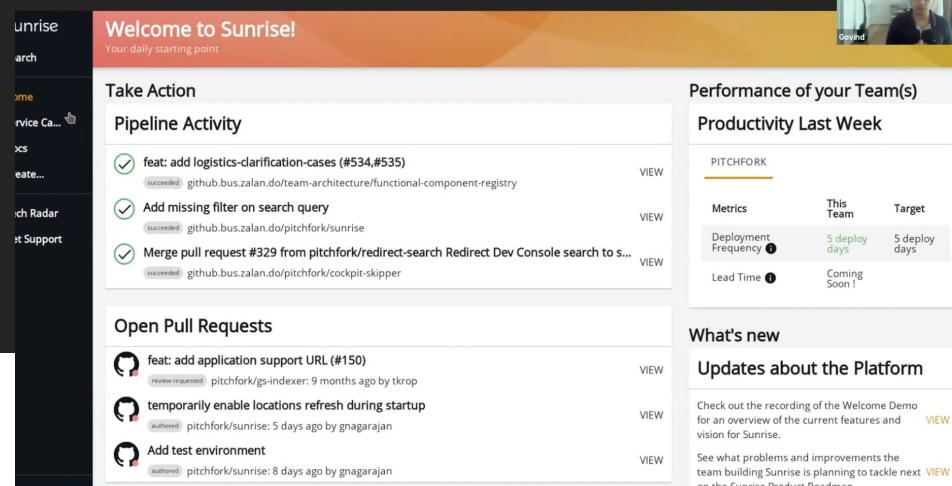
- Create an app**
Get started quickly with our many blueprints
- Get access**
Self-service access control for GitHub and more

Need Help?

- Request a feature**
- Chat with us**
- See new features**



Sunrise: Backstage at Zalando

Welcome to Sunrise!
Your daily starting point

Take Action

Pipeline Activity

- feat: add logistics-clarification-cases (#534,#535) [VIEW](#)
- Add missing filter on search query [VIEW](#)
- Merge pull request #329 from pitchfork/redirect-search Redirect Dev Console search to s... [VIEW](#)

Open Pull Requests

- feat: add application support URL (#150) [VIEW](#)
- temporarily enable locations refresh during startup [VIEW](#)
- Add test environment [VIEW](#)

Performance of your Team(s)

Productivity Last Week

Metrics	This Team	Target
Deployment Frequency	5 deploy days	5 deploy days
Lead Time	Coming Soon!	

What's new

Updates about the Platform

Check out the recording of the Welcome Demo for an overview of the current features and vision for Sunrise. [VIEW](#)

See what problems and improvements the team building Sunrise is planning to tackle next [VIEW](#) on the Sunrise Product Roadmap.



Demo: getting a 360° view on your assets with Avalia DX Hub

DX Hub

GROUP — ORGANIZATION
avalia-systems ⭐

OVERVIEW **VIBES**

Avalia Systems  

Organization for core team members (@avalia.systems)

 info@avalia.systems

Ownership

Direct Relations Aggregated Relations

7 SYSTEMS	3 TASKS
3 BACKENDS	3 FRONTENDS
2 SERVICES	1 WEBSITE

Members (15)

of Avalia Systems

 Alessandra Rauh alessandra@avalia.systems	 Ana Marques ana@avalia.systems	 Anderson Soffa soffa@avalia.systems
---	--	---

Settings

DX Hub

SEARCH

SYSTEM
time-machine-system ⭐

OVERVIEW **VIBES** **DIAGRAM**

About

DESCRIPTION
No description

OWNER: avalia-systems DOMAIN: No Domain

LIFECYCLE: experimental TAGS: No Tags

Relations



[View graph →](#)

Has components

Name	Owner	Type	Lifecycle	Des
time-machine-backend	avalia-systems	backend	experimental	
time-machine-frontend	avalia-systems	frontend	experimental	

APIs

Name	Owner	Type	Lifecycle	Des
------	-------	------	-----------	-----

This system does not contain any APIs.
[Learn how to change this.](#)

Demo: creating and documenting IP assets on Avalia DX Hub

DX Hub

- Search
- Avalia Systems
- Feel the Vibe
- Explore
- Catalog
- APIs
- Docs
- Create...
- Tech Radar
- Patterns
- Risks
- People
- Analytics
- Exoscale
- Settings

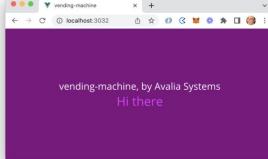
time-machine

Introduction

Edit this documentation locally

You can easily edit this documentation and get a preview on your local machine. You only have to clone the [time-machine](#) git repository and run a script that we provide. You will get a preview both in a Backstage-like layout and in a mkdocs layout.

Edit the markdown files, and you will immediately get the results. Commit your changes and the information will automatically be updated in DX Hub!



Functional description

Purpose

The time-machine system was generated with a DX Hub template. At the moment, it still shares the same purpose of the [template](#): demonstrate how design, build and deploy a system following the [Avalia](#) reference architecture.

Up to you!

This is obviously only the starting point: if you have created this new system, it is because you want to solve a

Next →

DX Hub

time-machine

Home

Component
system:time-machine-system

Owner
avalia-systems

Lifecycle
experimental

Table of contents

Functional description
Purpose
Target users
Features
Getting started

Architecture

The **time-machine system** is based on our standard architecture, where a single-page application interacts with a service over a REST API.



Backend

The **time-machine-backend component** is implemented in Java, on top of the [SpringBoot](#) framework. In most of our backend components, we use [Spring MVC](#) and [Spring Data](#).

If you have never used SpringBoot, we recognize to start with these tutorials:

- [Building a RESTful Web Service](#)
- [Accessing Data with JPA](#)

Describe what is specific to your component...

Frontend

The **time-machine-frontend component** is implemented in Javascript, with the [Vue 3](#) framework. In most of our frontend components, we use [Vue Router](#), [Vuex](#) and [PrimeVue](#).

Describe what is specific to your component...

← Previous
Introduction →

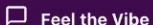
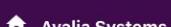
Next →
Deployment

 Search Avalia Systems Feel the Vibe Explore Catalog APIs Docs Create... Tech Radar Patterns Risks People Analytics Exoscale Settings



Welcome to DX Hub

Powered by Backstage - extended by Avalia Systems



Revenue

Quality

Cost

Time-to-market

Client satisfaction

Team satisfaction



3,038

●

2

●

18

●

26

1

+10

1

1

?

!

Good morning Olivier,

Things are moving well on Project X. Fabio has updated the playbook with his notes on the architecture deep dive.

You should have a look at the assess prod environment: the number of errors seems to grow.

Your Vue.js tutorial got 3 other likes, but one step has been flagged as broken.

Today, I will be happy if...

submit

Stream

Poll

Spotlight

Trending repositories today

TeamNe... / NewPipe 20.3k

A libre lightweight streaming front-end for Android.

Co... / stable-diffusion 2.3k

p.../Summer2023-I... 11.6k

Collection of Summer 2023 tech internships!

sh... / shadowsock... 54.9k

A C# port of shadowsocks

[See more trending repositories >](#)



Team awareness



DX Hub

Welcome to DX Hub

Powered by Backstage - extended by Avalia Systems

Search

Avalia Systems

Feel the Vibe

Explore

Catalog

APIs

Docs

Create...

Tech Radar

Patterns

Risks

People

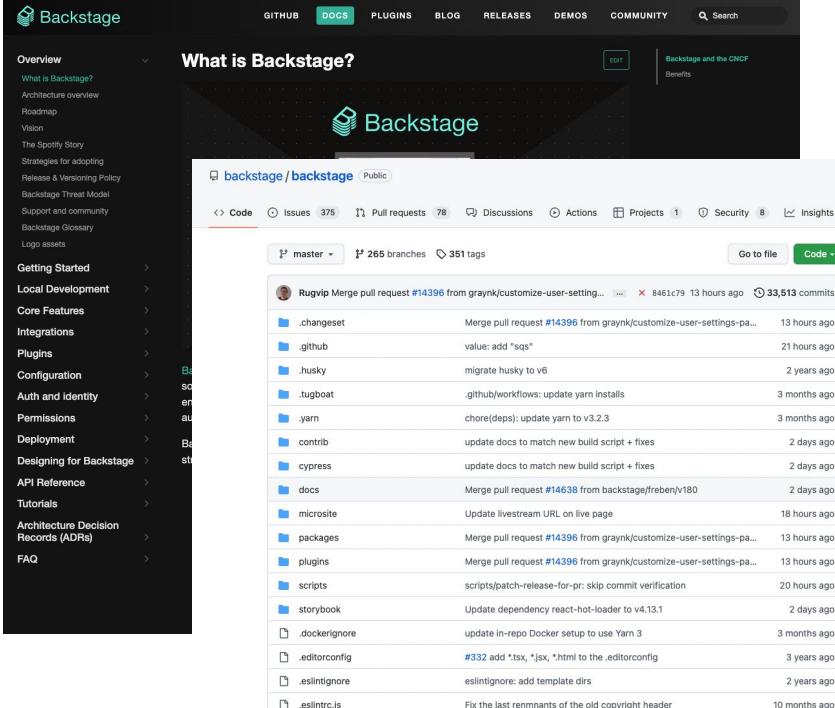
Analytics

Exoscale

Settings



References



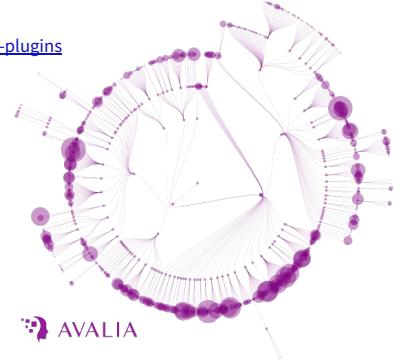
The screenshot shows the Backstage.io interface. On the left, there's a sidebar with sections like Overview, What is Backstage?, Architecture overview, Roadmap, Vision, The Spotify Story, Strategies for adopting, Release & Versioning Policy, Backstage Threat Model, Support and community, Backstage Glossary, and Logo assets. The main content area is titled "What is Backstage?" and features a "Code" tab showing a GitHub repository for "backstage/backstage". The repository has 375 issues, 78 pull requests, and 8 discussions. It lists 265 branches and 351 tags. A specific merge request by "Rugvip" is highlighted, showing a commit history with 33,513 commits.

<https://backstage.io/docs/overview/what-is-backstage>
<https://github.com/backstage/backstage>
<https://backstage.spotify.com/learn/>



The slide has a dark blue background with purple wavy lines. At the top is the AVALIA logo and website address. Below it is the title "A Picture is Worth a Thousand Words" and the subtitle "Bring Data Visualization in Backstage". In the center, it says "BACKSTAGE | CON, DETROIT OCTOBER 2022". A green box contains the "Backstage | con" logo. At the bottom right is a circular network diagram with nodes and connections.

<https://www.youtube.com/watch?v=riOXPNyp0eo>
<https://github.com/AvaliaSystems/avalia-backstage-plugins>



CONFIDENTIAL



Thank you



Olivier Liechti

Co-founder and CTO

olivier@avalia.io

www.avalia.io