

# Auth.it!

Reimagining the Keycloak  
Admin UI for B2B SaaS



<https://auth.it>

Based on Keycloak



Garth Patil

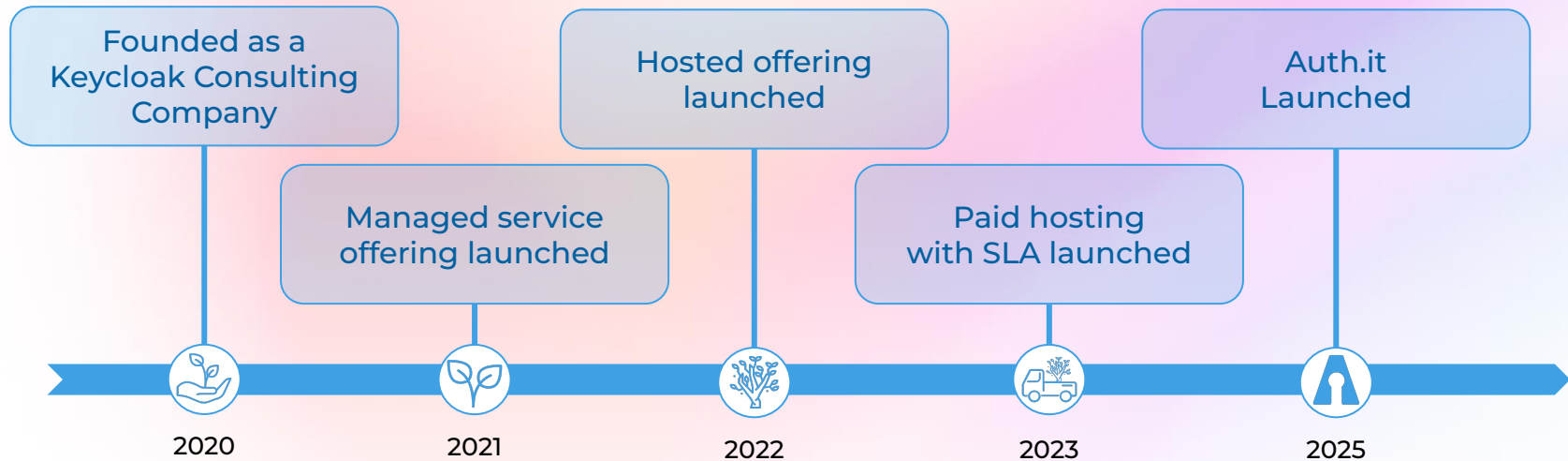
Jeff Patzer

Keycloak Dev Day [2025-03-06]

# Who is Phase Two?

*Keycloak extension, support and hosting.*

*Accelerate time-to-market and enterprise adoption for modern SaaS use cases.*



# What is happening in the IAM market?

- Incumbents (e.g. Auth0, Okta, Ping, OneLogin, MS, etc.) are marketing and sales driven, and no longer innovate.
- A second generation of companies is “skinning” IAM tools and calling them “Enterprise SSO” or “User Management”... and they’re winning the market. (e.g. Clerk, Frontegg, Stytch, WorkOS, etc.)
- **How?** By solving the problems that developers have with configuration and integration.

# But what about Keycloak?

- Keycloak is very mature, and handles 90% of use cases really well.
- Especially in the US, Keycloak is not known, and not even considered.
- The developer success story for getting configured and their apps protected is 🤩 for Keycloak
  - “Have a configuration question? **RTFM!**”
  - “Securing your app? Go to this ancient-looking list of unsupported OIDC libraries!”
- Keycloak is hard to operate, as it compares to the other “cloud native” open source apps.

# What is the problem we're trying to solve?

- **Awareness**

- Keycloak is largely unknown, especially in non-European markets, particularly among small-medium companies
- If you know of and understand how to configure Keycloak, you're already sold. If not, you're going elsewhere.

- **Ease of use**

- Setting up Keycloak for even simple Enterprise SaaS or CIAM use cases can be very **hard**
- Even though it has all the features, this drives lots of companies away from Keycloak (mostly to commercial products)

*Introducing...*



# How did we solve it (high-level)

- Listened to **customer stories** (both from customers and those who dropped Keycloak from consideration)
- Solved for the core configuration use cases with **use-case driven UI**
- Chose **conventions that hide the complexity** of Keycloak configuration
- Leveraged our assets
  - Phase Two's library of **custom extensions** that solve the underlying issues
  - **Competency in hosting Keycloak** at many scales for many use cases

# How did we solve it (the details) – and how did we do it in a SaaS app where installing custom extensions is impossible

- **Authentication** configuration in *one page*.
- **Branding** in *one page*.
- **Highlight** the **entities** people care about. Hide everything else.
- Externalize **events** as **webhooks**.
- “**API Keys**” for easy service accounts.
- Default “**Applications**” for most Client needs.



# Authentication - ON ONE PAGE

- **Limit** options (“convention over configuration”).
  - Show the 20% need, pre-set everything else.
- **Sensible defaults** for popular **social providers**.
  - For 80% pf Enterprise SaaS and CIAM use cases, only 4-5 are necessary.
  - Remove unnecessary configuration. Conventions for everything else.
- **Email is the username**. Welcome to 1999!
- **Strong passwords** are a necessity. Use sensible defaults, and make it **intuitive** to update.
- Make the use of **2fa** and **alternate credentials** simple and intuitive

# Authentication - NO FLOWS!

- Configuring Keycloak authentication flows are where most users **abandon**.
- The 80% flow is a solved problem, and most people just want to clone that.
- **One Authentication Flow to Rule Them All**
  - “Features” that allow a simple, intuitive mental model for the administrator
  - Conditional authenticator based on “Feature” attributes
  - Required actions that add other RAs based on “Feature” attributes and user state



# Branding - Important and FORGOTTEN

## Existing theme situation

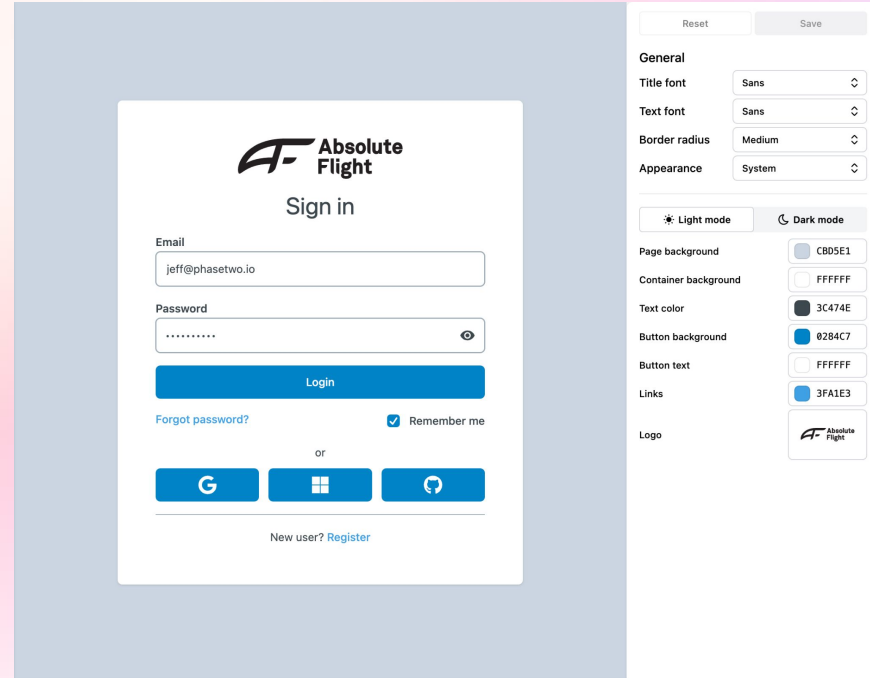
- Existing theme development is cumbersome in Keycloak
- Theme developers don't know FTL
- Base theme leaves much design to be desired
- No way to introduce simple theme changes

## Existing solutions

- Write/extend a new FTL theme 🤖
- Use [Keycloakify](#) 🙏
  - Works well, React (Js) is used by theme developers
  - Must know CSS/JS to use
- What's missing?

# Branding - Visual Theme Editor

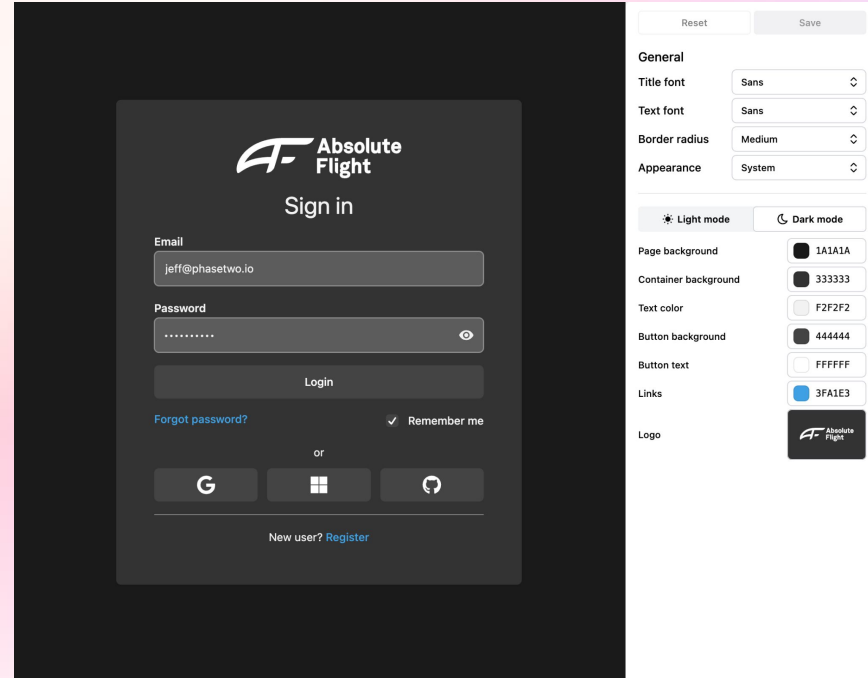
- Clean, modern design
- **WYSIWYG** editor
- Updates in real time for **instant design feedback**
- Implemented with Keycloakify
- CSS vars stored in realm attributes
- Email theme support\*
- Export theme for use in any Keycloak deployment\*



\*coming soon

# Branding - Custom Templates

- **React** components. Use in Auth.it UI and the Theme for consistency
- Custom UI components (not Patternfly)
- Tailwind + Emotion + CSS vars
- **Different UI layouts** than base KC theme
- Supports all necessary templates
  - Standard Keycloak Templates
  - P// Org and Magic Link Extensions
- Light and Dark mode support
- Quickly upload logos

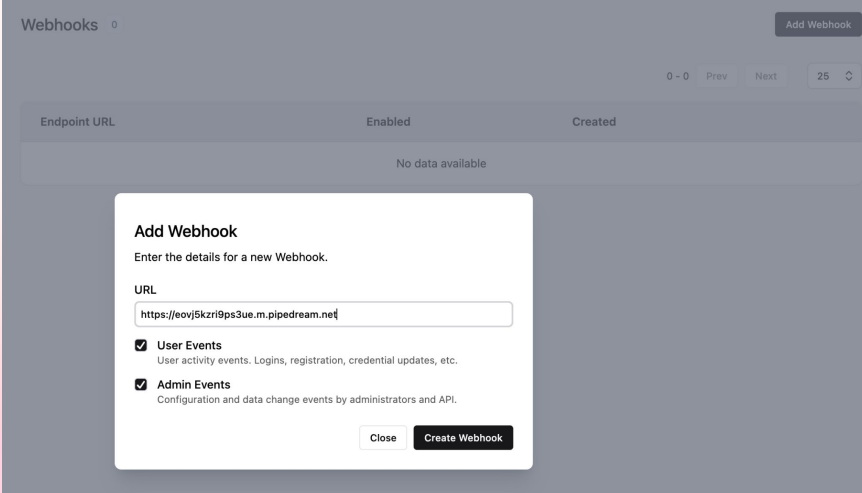


# *Users & Organizations (the only entities most admins care about)*

- **Limit** the data and actions displayed to the most common use cases
- **Don't hide** common needs, but make the **results clear** (eg. Impersonate, Delete)
- Get the administrator **quickly** to the need
- Required actions should be **implicit** (set up and reset credentials)
- “**Enterprise SSO**” identity providers are **associated** with the **Organization**

# Webhooks - FREE the DATA

- Getting event data out of Keycloak today requires **extensions**
- Stripe is the gold standard. We modeled it after them.
- All **User or Admin events** can be **sent** to a **Webhook endpoint**.
- **No** additional extension **installation** required.
- See presentation from [KeyConf24](#) on all of the extensions



Webhooks 0 Add Webhook

0 - 0 Prev Next 25 ↕

Endpoint URL	Enabled	Created
No data available		

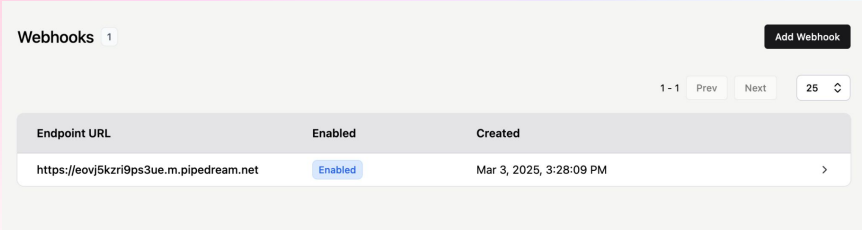
**Add Webhook**  
Enter the details for a new Webhook.

URL

**User Events**  
User activity events. Logins, registration, credential updates, etc.

**Admin Events**  
Configuration and data change events by administrators and API.

Close Create Webhook



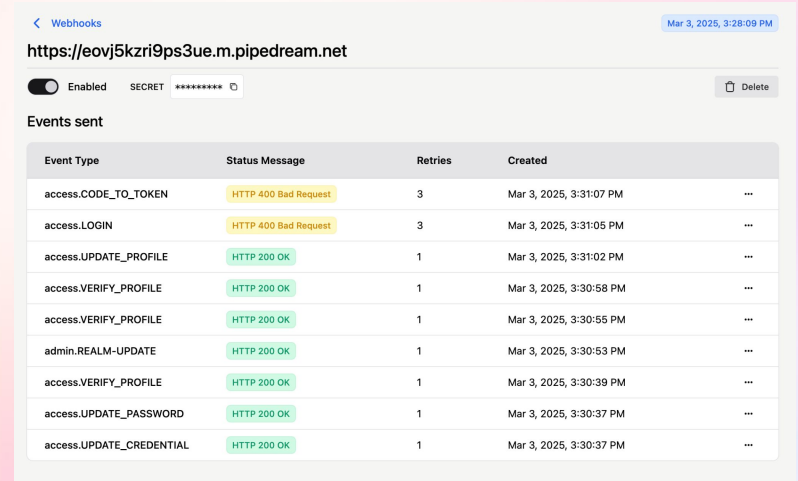
Webhooks 1 Add Webhook

1 - 1 Prev Next 25 ↕

Endpoint URL	Enabled	Created
https://eovj5kzri9ps3ue.m.pipedream.net	<span>Enabled</span>	Mar 3, 2025, 3:28:09 PM

# Webhooks - Easy to Send, Easy to See

- Quickly **view data** that goes to a specific endpoint
- Send data to **multiple locations**
- View attempts and **debug easily**
- **Resend** on failures
- Payload **presented** in a useful way
- Payload is a **unified payload** that doesn't separate endpoints



Webhooks

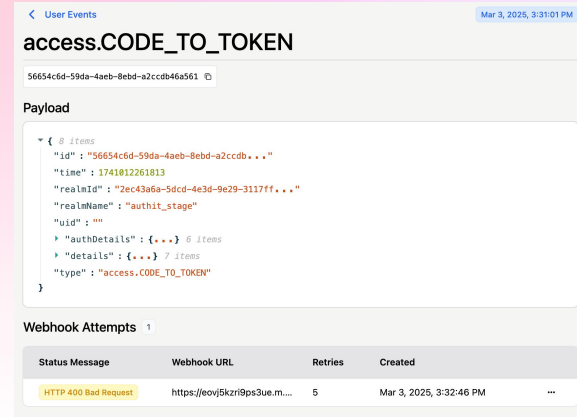
Mar 3, 2025, 3:28:09 PM

https://eovj5kzri9ps3ue.m.pipedream.net

Enabled SECRET \*\*\*\*\* Delete

Events sent

Event Type	Status Message	Retries	Created
access.CODE_TO_TOKEN	HTTP 400 Bad Request	3	Mar 3, 2025, 3:31:07 PM
access.LOGIN	HTTP 400 Bad Request	3	Mar 3, 2025, 3:31:05 PM
access.UPDATE_PROFILE	HTTP 200 OK	1	Mar 3, 2025, 3:31:02 PM
access.VERIFY_PROFILE	HTTP 200 OK	1	Mar 3, 2025, 3:30:58 PM
access.VERIFY_PROFILE	HTTP 200 OK	1	Mar 3, 2025, 3:30:55 PM
admin.REALM-UPDATE	HTTP 200 OK	1	Mar 3, 2025, 3:30:53 PM
access.VERIFY_PROFILE	HTTP 200 OK	1	Mar 3, 2025, 3:30:39 PM
access.UPDATE_PASSWORD	HTTP 200 OK	1	Mar 3, 2025, 3:30:37 PM
access.UPDATE_CREDENTIAL	HTTP 200 OK	1	Mar 3, 2025, 3:30:37 PM



User Events

Mar 3, 2025, 3:31:01 PM

access.CODE\_TO\_TOKEN

56654c6d-59da-4aeb-8ebd-a2ccdb46a561

Payload

```
{
  "@_items": [
    {
      "id": "56654c6d-59da-4aeb-8ebd-a2ccdb...",
      "time": 1741012261813,
      "realmId": "2ec43a6a-5dc4-4e3d-9e29-3117ff...",
      "realmName": "auth1_stage",
      "uid": "",
      "authDetails": {
        "items": 6
      },
      "details": {
        "items": 7
      },
      "type": "access.CODE_TO_TOKEN"
    }
  ]
}
```

Webhook Attempts

Status Message	Webhook URL	Retries	Created
HTTP 400 Bad Request	https://eovj5kzri9ps3ue.m...	5	Mar 3, 2025, 3:32:46 PM



# “API Keys”

- The **familiar** phrase used to reference API access credentials.
- Make it **easy** to make a confidential client with a service account that has the appropriate permissions.
- It’s still client id/secret, but it’s **simple** to understand and use.

### Add API key

Enter the details for a new API key.

**Name**

**Permissions**

**Manage Configuration**

View Configuration

**Manage Organizations**

View Organizations

**Manage Users**

View Users

**Manage applications**

View Applications

**Manage Events and Webhooks**

View Events and Webhooks

**Manage Identity Providers**

View Identity Providers

# Applications

- Another source of abandonment
  - “What is a client and why should I care?”
- Use familiar language. **Application** instead of Client.
- “How do I protect a frontend or backend application?”
  - Collect the **minimum** information required to configure the application in a secure manner.
- We really only care about OIDC
- By presenting an opinionated view, we can encourage best practices

# Applications

- Even the Keycloak Client “Wizard” is too much information and doesn’t explain the use case
- Set a reasonable default for frontend (public) and backend (confidential) applications to get the admin successful

**NAME**  
Backend Application backend \*\*\*\*\*

For applications running on a server (e.g., Node.js, Java, Python, or Go). Can securely store client secrets. Acts as a resource server if validating access tokens from a frontend. May directly obtain tokens on behalf of users or other services.

**Home URL**  
  
Fully qualified url for the location of your application

**Redirect URIs** + Add URI  
  
Redirect URIs are the URLs that Authit will redirect to after a user logs in or out.

**Allowed Origins (CORS)** + Add Origin  
 Same as Redirect URIs

Allowed Origins are the URLs that are allowed to make requests to your application.

Reset Update

# Future (roadmap)

- General availability, public release
- **Make APIs Great Again!**
  - Great API documentation and ergonomic SDKs
- Additional enhancements to Branding UI
  - Download theme JAR to use in other places
- Eject to Keycloak
  - Open source the runtime extensions
- Release “new” (existing Keycloak) features as we understand the developer use case for them and can simplify configuration and use

# Special thanks...

- The Keycloak **maintainers, authors** and **contributors**
- @dasniko & @srose for this **great event**
- @garronej for **keycloakify**
- @dteleguin for **beercloak**
- @thomasdarimont for **Awesome Keycloak** and so many excellent **examples**
- **Phase Two's community contributors**
- **The entire Keycloak community!**

# Questions?

**THANK YOU!**

More resources:

- Auth.it: <https://auth.it>
- Phase Two: <https://phasetwo.io>
- GitHub: <https://github.com/p2-inc>