

DASHBOARD

2021

8. DECEMBER

Api-BangGang
By: Karl Störzel



Introduction

This project is a dashboard for multiple services like github, spotify, reddit, teams, twitch and discord. You can find different information depending on your settings about these services and your connected accounts.

Installation

Docker

Firstly, be sure Docker is installed on your local machine by running “docker –version” in your terminal/powershell etc. If you get no version output click [this link](#) and follow the instructions to download and install docker.

Docker-Compose

After you have docker, now check for docker-compose by running “docker-compose –version” in your terminal/powershell etc. If you get no version output click [this link](#) and follow the instructions to download and install docker-compose.

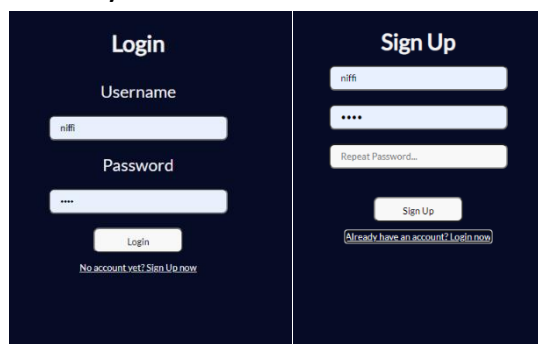
Dashboard installation

Now run “sudo docker-compose up” (linux) or “docker-compose up” (windows/mac) in the root of the project. The dashboard and database will initialize itself. When it is done, press ctrl+C and rerun the previous command. After that you can access the dashboard at “<http://127.0.0.1:3000>”.

Usage

Login

When you access the website, you can login or sign up at the bottom if you have not already created an account.



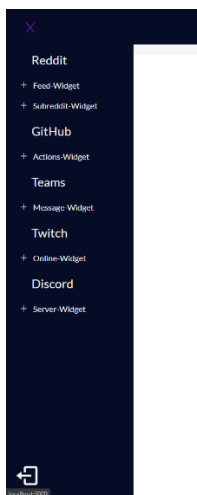
The image shows two side-by-side forms on a dark blue background. The left form is titled 'Login' and contains fields for 'Username' (with 'nifi' entered) and 'Password' (with four dots). Below these is a 'Login' button and a link that says 'No account yet? Sign Up now'. The right form is titled 'Sign Up' and contains fields for 'Email' (with 'nifi' entered), 'Password' (with four dots), and 'Repeat Password...'. Below these is a 'Sign Up' button and a link that says 'Already have an account? Login now'.

Authentication

By using our dashboard, you have to accept all the Authentication asked by the services. If you don't want to accept them, do not use the dashboard. You will get redirected to all the Services to authorize the dashboard to access your data. It will just access the data you want to see and only when you want to see it.

Widget Creation

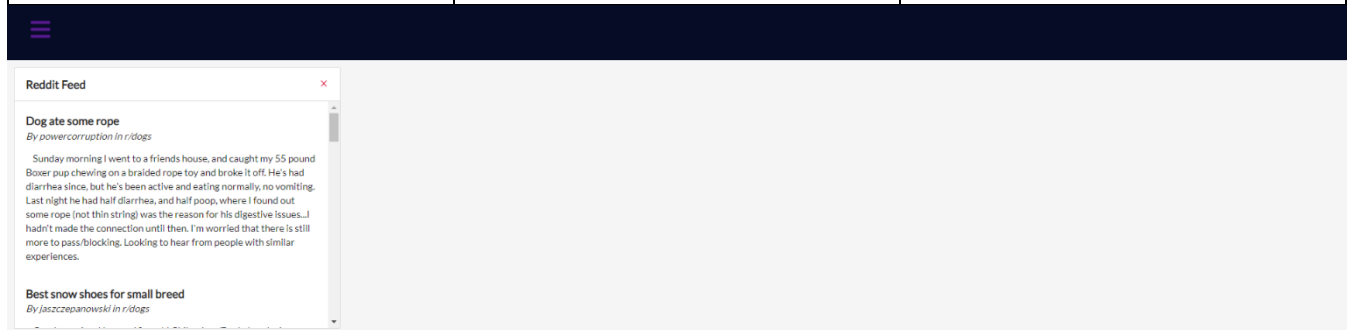
To create a new widget, click on the top-left the three lines to open the widget menu. You see the service names and the provided widgets. Just click on one widget to create it and set up the settings for it.



Design

Colors

Component	Colorcode	Reason
Main Color	#060b26	plain, serious, relaxing
Secondary Color	#F8F8F8	contrast, simple
Close buttons	DC143CFF	close is warning red



REST-API

The API is a rest API. When you want to do a request with user information you need an access-token which you get by creating an account.

Main path

The main path is <http://127.0.0.1:8080>.

Allowed Header

Access-Control-Allow-Origin	*
Access-Control-Allow-Credentials	true
Access-Control-Allow-Methods	GET POST PUT DELETE
Access-Control-Allow-Headers	Access-Control-Allow-Headers Origin Authorization Accept X-Requested-With Content-Type Access-Control-Request-Method Access-Control-Request-Headers

About

Description	Get the information about all possible services and integrated widgets
Request Type	GET
Header	No header needed
Params	No params needed
Path	/about.json
Response	Type: json DataKeys: <ul style="list-style-type: none">• client<ul style="list-style-type: none">○ host• server<ul style="list-style-type: none">○ current_time○ services []<ul style="list-style-type: none">▪ name▪ widgets []<ul style="list-style-type: none">• name• description• params []<ul style="list-style-type: none">○ name○ type

User

Description	Get All Usernames
Request Type	GET
Header	No header needed
Params	No params needed
Path	/user
Response	Type: json DataKeys: <ul style="list-style-type: none">• users []<ul style="list-style-type: none">○ username

Description	Create a new user
Request Type	POST
Header	No header needed
Body	Keys: <ul style="list-style-type: none"> • name • password
Path	/user
Response	Type: json DataKeys: <ul style="list-style-type: none"> • access_token

Description	Delete a user
Request Type	DELETE
Header	Authorization <i>ADMIN_ACCESS_TOKEN</i>
Body	Keys: <ul style="list-style-type: none"> • name • user_id
Path	/user
Response	"User deleted"

Description	Get All users with their information (requires admin rules)
Request Type	GET
Header	Authorization <i>YOUR_ACCESS_TOKEN</i>
Params	Keys: <ul style="list-style-type: none"> • Name • Password
Path	/user/adm
Response	Type: json DataKeys: <ul style="list-style-type: none"> • users [] <ul style="list-style-type: none"> ○ username

Description	Get All Usernames		
Request Type	GET		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Params	No params needed		
Path	/user/widgets		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • user • subreddits [] <ul style="list-style-type: none"> ○ timer_seconds ○ servicename ○ id ○ subreddit ○ sorting_name • feeds [] <ul style="list-style-type: none"> ○ timer_seconds ○ servicename ○ id ○ sorting_name • actions [] <ul style="list-style-type: none"> ○ timer_secods ○ servicename ○ id ○ owner ○ repo • messages [] <ul style="list-style-type: none"> ○ timer_seconds ○ servicename ○ id ○ sorting_new • channels [] <ul style="list-style-type: none"> ○ timer_seconds ○ servicename ○ id ○ sorting_type • server [] <ul style="list-style-type: none"> ○ timer_seconds ○ servicename ○ id ○ admin_only 		

Login

Description	Check if username and password are right
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none">• name• password
Path	/login
Response	Type: json DataKeys: <ul style="list-style-type: none">• guid (access_token)

Teams

Description	Authenticate teams
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none">• access_token
Path	/teams/login
Response	redirection

Description	Create message widget	
Request Type	POST	
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>
Body	Keys: <ul style="list-style-type: none">• sorting_type• timer	
Path	/teams/widgets/messages/	
Response	Type: json DataKeys: <ul style="list-style-type: none">• config_id	

Description	Get message widget		
Request Type	GET		
Header	Authorization	YOUR_ACCESS_TOKEN	
Params	No params needed		
Path	/teams/teams/widgets/messages/:id		
Query	"Id" is an existing message-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• timer• sorting_type• messages []<ul style="list-style-type: none">○ type○ text○ name○ time		

Description	Change message widget		
Request Type	PUT		
Header	Authorization	YOUR_ACCESS_TOKEN	
Body	Keys: <ul style="list-style-type: none">• sorting_type• timer		
Path	/teams/widgets/messages/ :id		
Query	"Id" is an existing message-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• result		

Description	Delete message widget		
Request Type	DELETE		
Header	Authorization	YOUR_ACCESS_TOKEN	
Body	No body needed		
Path	/teams/widgets/messages/ :id		
Query	"Id" is an existing message-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• result		

Reddit

Description	Authenticate reddit
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none">• access_token
Path	/reddit/login
Response	redirection

Description	Create feed widget		
Request Type	POST		
Header	Authorization	YOUR_ACCESS_TOKEN	
Body	Keys: <ul style="list-style-type: none">• sorting_type• timer		
Path	/reddit/widgets/feed/		
Response	Type: json DataKeys: <ul style="list-style-type: none">• config_id		

Description	Get feed widget		
Request Type	GET		
Header	Authorization	YOUR_ACCESS_TOKEN	
Params	No params needed		
Path	/reddit/widgets/feed/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• timer• sorting_type• values [<ul style="list-style-type: none">○ name○ subreddit○ title○ text○ votes○ author]		

Description	Change feed widget		
Request Type	PUT		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	Keys: <ul style="list-style-type: none"> • sorting_timer • timer 		
Path	/reddit/widgets/feed/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Description	delete feed widget		
Request Type	DELETE		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	No body needed		
Path	/reddit/widgets/feed/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Description	Create subreddit widget		
Request Type	POST		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	Keys: <ul style="list-style-type: none"> • sorting_type • timer • subreddit 		
Path	/reddit/widgets/feed/		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • config_id 		

Description	Get subreddit widget		
Request Type	GET		
Header	Authorization	YOUR_ACCESS_TOKEN	
Params	No params needed		
Path	/reddit/widgets/subreddit/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• timer• sorting_type• values [<ul style="list-style-type: none">○ name○ subreddit○ title○ text○ votes○ author		

Description	Change subreddit widget		
Request Type	PUT		
Header	Authorization	YOUR_ACCESS_TOKEN	
Body	Keys: <ul style="list-style-type: none">• sorting_timer• subreddit• timer		
Path	/reddit/widgets/feed/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• result		

Description	Delete message widget		
Request Type	DELETE		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	No body needed		
Path	/reddit/widgets/feed/ :id		
Query	"Id" is an existing feed-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Github

Description	Authenticate teams
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none"> • access_token
Path	/github/login
Response	redirection

Description	Create actions widget		
Request Type	POST		
Header	<table> <tr> <td>Authorization</td><td><i>YOUR_ACCESS_TOKEN</i></td></tr> </table>	Authorization	<i>YOUR_ACCESS_TOKEN</i>
Authorization	<i>YOUR_ACCESS_TOKEN</i>		
Body	Keys: <ul style="list-style-type: none"> • owner • repo • timer 		
Path	/github/widgets/actions/		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • config_id 		

Description	Get actions widget		
Request Type	GET		
Header	<table> <tr> <td>Authorization</td><td><i>YOUR_ACCESS_TOKEN</i></td></tr> </table>	Authorization	<i>YOUR_ACCESS_TOKEN</i>
Authorization	<i>YOUR_ACCESS_TOKEN</i>		
Params	No params needed		
Path	/github/widgets/actions/:id		
Query	"Id" is an existing actions-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • timer • sorting_type • actions [] <ul style="list-style-type: none"> ○ name ○ event ○ conclusion ○ commit ○ time 		

Description	Change actions widget		
Request Type	PUT		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	Keys: <ul style="list-style-type: none"> • owner • repo • timer 		
Path	/github/widgets/actions/ :id		
Query	"Id" is an existing actions-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Description	Delete actions widget		
Request Type	DELETE		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	No body needed		
Path	/github/widgets/actions/ :id		
Query	"Id" is an existing actions-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Twitich

Description	Authenticate twitch
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none"> • access_token
Path	/twitch/login
Response	redirection

Description	Create channels widget
Request Type	POST
Header	Authorization <i>YOUR_ACCESS_TOKEN</i>
Body	Keys: <ul style="list-style-type: none"> • sorting_type • timer
Path	/twitch/widgets/channels
Response	Type: json DataKeys: <ul style="list-style-type: none"> • config_id

Description	Get channels widget
Request Type	GET
Header	Authorization <i>YOUR_ACCESS_TOKEN</i>
Params	No params needed
Path	/twitch/widgets/channels/:id
Query	"Id" is an existing channels-widget-id
Response	Type: json DataKeys: <ul style="list-style-type: none"> • timer • sorting_type • data [] <ul style="list-style-type: none"> ○ username ○ game_name ○ title ○ viewer_count

Description	Change channels widget		
Request Type	PUT		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	Keys: <ul style="list-style-type: none"> • sorting_type • timer 		
Path	/twitch/widgets/channels/:id		
Query	"Id" is an existing channels-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Description	Delete channels widget		
Request Type	DELETE		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	No body needed		
Path	/twitch/widgets/channels/:id		
Query	"Id" is an existing channels-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Discord

Description	Authenticate teams
Request Type	GET
Header	No header needed
Params	Keys: <ul style="list-style-type: none">• access_token
Path	/discord/login
Response	redirection

Description	Create message widget		
Request Type	POST		
Header	Authorization	YOUR_ACCESS_TOKEN	
Body	Keys: <ul style="list-style-type: none">• timer• admin_only		
Path	discord/widgets/server/		
Response	Type: json DataKeys: <ul style="list-style-type: none">• config_id		

Description	Get message widget		
Request Type	GET		
Header	Authorization	YOUR_ACCESS_TOKEN	
Params	No params needed		
Path	discord/widgets/server/ :id		
Query	"Id" is an existing message-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none">• server_number• sorting_type• server_names [<ul style="list-style-type: none">○ name○ owner		

Description	Change server widget		
Request Type	PUT		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	Keys: <ul style="list-style-type: none"> • timer • admin_only 		
Path	discord/widgets/server/ :id		
Query	"Id" is an existing server-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		

Description	Delete server widget		
Request Type	DELETE		
Header	Authorization	<i>YOUR_ACCESS_TOKEN</i>	
Body	No body needed		
Path	discord/widgets/server/ :id		
Query	"Id" is an existing server-widget-id		
Response	Type: json DataKeys: <ul style="list-style-type: none"> • result 		