

Flowforge UI – system low-code do budowy interfejsów API

Tryb ciemny

Konrad Firlej

Promotor: dr inż. Krzysztof Augustynek

Luty 2026



Agenda

- Wstęp: potrzeba biznesowa, cel, analiza rozwiązań
- Wymagania: funkcjonalne i нефunkcjonalne
- Technologie i narzędzia
- Projekt systemu: model danych, klasy, use case, UML
- Wnioski i dalszy rozwój
- Widoki główne (Workflows)
- Akcje na workflow (More → Versions/Rename/Import)
- Konfiguracja bloków (If, Switch, Calculation, HTTP itp.)
- Edytor workflow (canvas, palette, variables)
- Run drawer (open, result)
- Widoki główne (Executions)
- Detale wykonania (Execution details)
- Widoki główne (Scheduler)
- Harmonogram (create/edit)
- Widoki główne (Blocks)

Wstęp

- Potrzeba biznesowa: szybkie budowanie i uruchamianie przepływów API bez kodu back-end.
- Dlaczego: skrócenie time-to-market integracji oraz ujednolicenie operacji danych.
- Do czego ma służyć: orkiestracja bloków (HTTP, parser, transformacje, warunki) w jednolitym edytorze.
- Analiza podobnych rozwiązań: n8n, Zapier, Node-RED – różnicujemy się lekkim UI i kontrolą on-prem.

Wymagania

Funkcjonalne

- Tworzenie, wersjonowanie i import/eksport workflowów.
- Edycja bloków (If, Switch, HTTP, Parser, Calculation, itp.) na canvasie.
- Uruchamianie i podgląd wyników (Run drawer, Execution details).
- Planowanie zadań (Scheduler: create/edit/run now).

Niefunkcjonalne

- Responsywne UI (jasny/ciemny motyw).
- Szybki czas renderu i stabilność canvasu.

Technologie i narzędzia

- Języki: TypeScript (frontend), C#/.NET (API), SQL (SQLite dev).
- Biblioteki: React, React Flow, Vite, Playwright, NUnit, EF Core.
- Narzędzia: VS Code / Rider, latexmk (slidedeck), jq/rg (dev), Git.

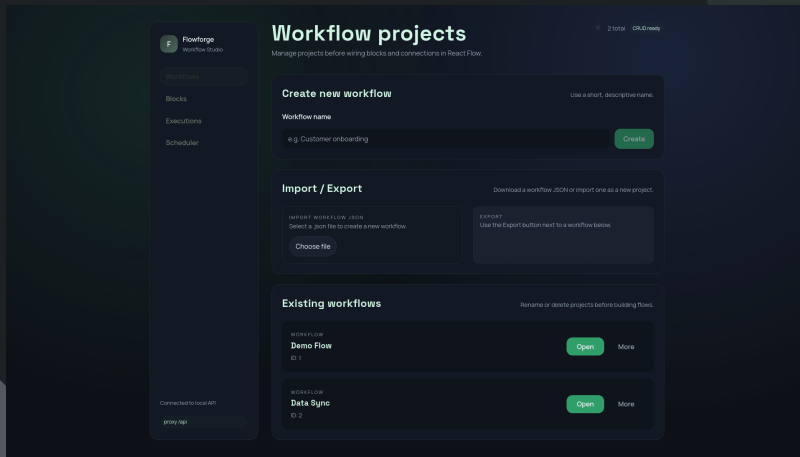
Projekt systemu

- Schemat bazy: tabele Workflow, Version, Block, Execution, Schedule (SQLite dev, migrowalne do produkcyjnej bazy).
- Schemat klas: DTO + modele EF Core; pary interfejs/implementacja dla serwisów i repozytoriów.
- Use case: tworzenie/edycja workflow, uruchomienie, schedulowanie, analiza przebiegu.
- UML (wysoki poziom): edytor (UI) → API (Controllers) → Services/Repositories → DB.

Wnioski

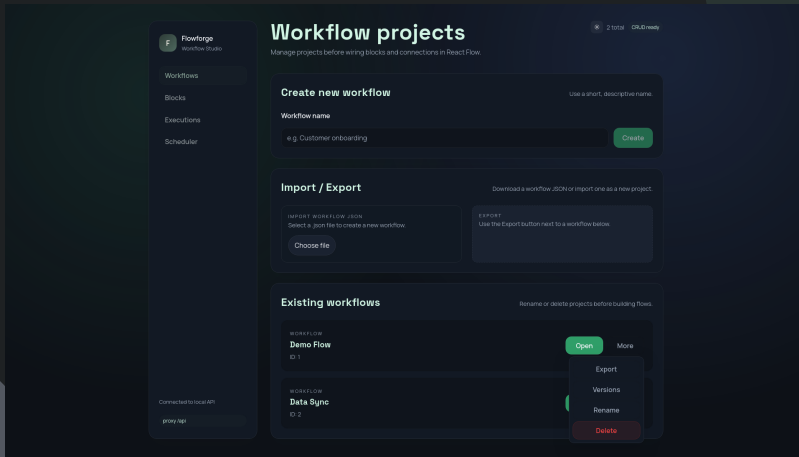
- Zrealizowano pełen zestaw widoków: edytor, akcje workflow, run drawer, execution details, scheduler.
- Kluczowe problemy: precyzyjne pozycjonowanie bloków na canvasie, spójność light/dark, czas ładowania przed screenshotem.
- Co się nie udało/plan: dołożenie realnej autoryzacji, rozbudowa monitoringu runtime, optymalizacja dla bardzo dużych diagramów.

Workflows



Lista projektów workflow – tryb ciemny

Workflow actions – More



Menu akcji "More" (ciemny)

Workflow actions – Versions

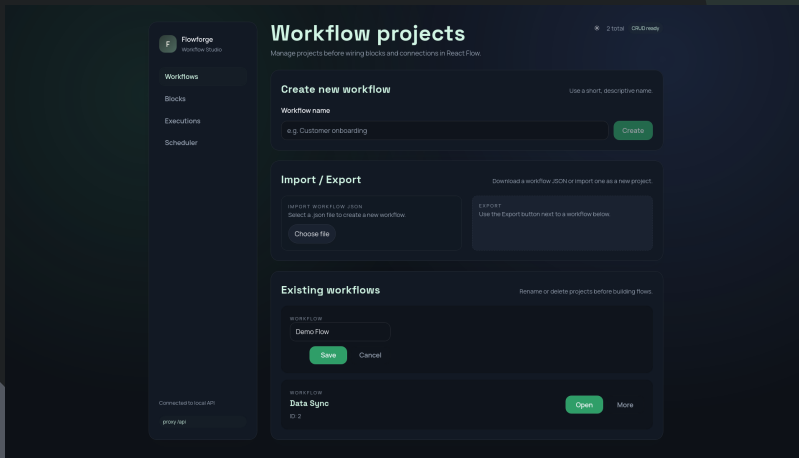
The screenshot displays the Flowforge Workflow Studio interface. A modal titled 'Versions (workflow #1)' is open, showing a list of workflow versions. The modal includes a 'Close' button in the top right corner. Below the title, there is a note: 'Migawki zapływywane automatycznie przy zapisie workflow.' and a toggle for 'Pokaż tylko aktywne' (Show only active) which is currently turned on. The versions are listed in a table with columns for version number, name, date, and actions ('Przywróć' - Restore and 'Usuń' - Delete).

Version	Name	Date	Actions
v5	v5	11/4/2025, 11:00:00 AM	Przywróć, Usuń
v4	v4	11/1/2025, 9:30:00 AM	Przywróć, Usuń
v3	v3	12/20/2025, 12:00:00 PM	Przywróć, Usuń
v2	v2	12/10/2025, 11:00:00 AM	Przywróć, Usuń
v1	v1	12/1/2025, 9:00:00 AM	Przywróć, Usuń

The background interface shows the 'Workflow projects' section with a sidebar containing 'Workflows', 'Blocks', 'Executions', and 'Scheduler'. The main area displays a list of workflow projects, including 'Data Sync' with an 'Open' button.

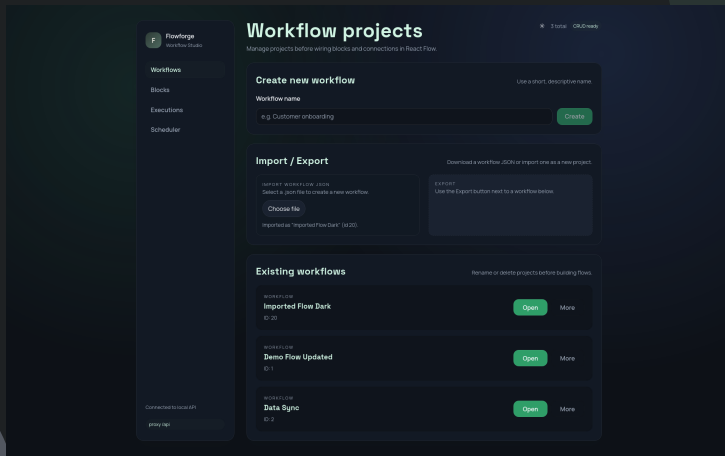
Lista wersji workflow (ciemny)

Workflow actions – Rename



Formularz zmiany nazwy (ciemny)

Workflow actions – Import



Import workflow z pliku (ciemny)

If block

The image shows the 'If Config' interface for React Flow in dark mode. The interface is divided into three main sections:

- Left Panel:** Contains a 'CANVAS' section with '+ Add block' and 'Variables' buttons, and a 'CONTEXT' section with a right-click action tip.
- Canvas:** Displays a workflow with two blocks: a 'Start block' (labeled 'Start') and an 'If block' (labeled 'If block'). They are connected by a dashed arrow. The 'If block' has two outgoing edges, one green and one red.
- Right Panel (If settings):** A configuration panel for the 'If block' with a 'Close' button. It includes:
 - First value:** A dropdown menu with the text 'Variable (e.g. amount) or literal'.
 - Second value:** A dropdown menu with the text 'Variable (e.g. total) or literal'.
 - Compare as:** A dropdown menu with 'String' selected.
 - Footer:** A note stating 'If evaluates to true, the first outgoing edge is taken; otherwise the second (error) edge.'

Konfiguracja bloku If – tryb ciemny

Switch block

The screenshot displays the 'Switch Config' interface for React Flow, set in a dark theme. The main workspace is a canvas where a 'Start block' (labeled 'Start') is connected to a 'Switch block' (labeled 'Route by case labels'). The 'Switch block' has three output ports labeled '1', '2', and 'default'. To the left of the canvas, there are two panels: 'CANVAS' with an 'Add block' button and 'CONTEXT' with a 'Variables' button. To the right, the 'Switch settings' panel is open, showing configuration options for the selected 'Switch block'. This panel includes a 'Switch settings' header, a 'Close' button, and sections for 'Expression / variable' (set to 'Variable or literal'), 'Cases' (containing 'Case 1' and 'Case 2', each with a 'Value (e.g. pending)' and a 'Remove' button), and an 'Add case' button. A descriptive text at the bottom of the settings panel states: 'Switch routes by matching case values to connection labels. Use the default handle for unmatched values.'

Konfiguracja bloku Switch – tryb ciemny

Calculation block

Calculation Config
Design blocks and connections with React Flow.

CANVAS

- + Add block
- Variables

CONTEXT
Right-click canvas or elements for quick actions.

START Ready

Start block
Start

CALCULATION Ready

Calculation block
Calculation

Calculation settings Close

Calculation block

Operation
Add

First variable
e.g. amount

Second variable
e.g. tax

Result variable
e.g. total

Saved as JSON config with CalculationOperation and variable names.

Konfiguracja bloku Calculation – tryb ciemny

Text Transform block

The screenshot displays the 'TextTransform Config' window, which is used for designing blocks and connections with React Flow. The interface is divided into three main sections:

- CANVAS:** Located on the left, it contains two buttons: '+ Add block' and 'Variables'. Below these buttons is the 'CONTEXT' section, which includes the text 'Right-click canvas or elements for quick actions'.
- CANVAS AREA:** The central workspace shows a flow diagram. It starts with a 'Start block' (labeled 'Start') connected by a dashed arrow to a 'Text Transform block' (labeled 'Text transform'). Both blocks have a 'Ready' status and a settings gear icon.
- Text Transform settings:** A panel on the right side of the canvas area. It includes a 'Close' button at the top right. The settings are organized into sections:
 - Operation:** A dropdown menu currently set to 'Trim'.
 - Input text (literal):** A text input field containing 'Sample text'.
 - Or variable (overrides literal):** A dropdown menu currently set to '\$text'.
 - Result variable:** A dropdown menu currently set to '\$result'.

At the bottom of the settings panel, a note states: 'Variable input takes priority over literal. Output is stored in the chosen result variable.'

Konfiguracja bloku Text Transform – tryb ciemny

Text Replace block

The screenshot displays the 'TextReplace Config' interface in a dark theme. The main canvas shows a workflow starting with a 'START' block (labeled 'Start block') connected to a 'TEXT REPLACE' block (labeled 'Text Replace block'). Both blocks are marked as 'Ready' and have a settings gear icon. The 'TEXT REPLACE' block is configured with 'foo' as the 'From' value and 'bar' as the 'To' value. The 'Replacements' section on the right shows these values and includes a 'Result variable' set to '\$result'. The 'Text Replace settings' panel on the right provides detailed configuration options, including 'Input text (literal)' set to 'foo bar', 'Or variable (overrides literal)' set to '\$text', and 'Replacements' with 'From' as 'foo' and 'To' as 'bar'. The 'Use regex' option is disabled, and 'Ignore case' is enabled. A note at the bottom states: 'Rules are applied in order. Regex rules are capped for safety; invalid patterns are skipped.'

TextReplace Config
Design blocks and connections with React Flow

CANVAS
+ Add block
= Variables

CONTEXT
Right-click canvas or elements for quick actions

START Ready

Start block
Start

TEXT REPLACE Ready

Text Replace block
Text replace

Text Replace settings Close

Text replace block

Input text (literal)
foo bar

Or variable (overrides literal)
\$text

Replacements Add rule

1 rule(s)

From
foo

To
bar

☐ Use regex ☒ Ignore case

Result variable
\$result

Rules are applied in order. Regex rules are capped for safety; invalid patterns are skipped.

Konfiguracja bloku Text Replace – tryb ciemny

HTTP request block

The screenshot displays the 'HttpRequest Config' interface in a dark theme. On the left, a sidebar contains 'CANVAS' with '+ Add block' and 'Variables' buttons, and 'CONTEXT' with a right-click action tip. The main canvas shows a workflow: a 'Start block' (labeled 'Start') connects via a dashed arrow to an 'HTTP request block' (labeled 'HTTP request'). Both blocks have 'Ready' status and gear icons. On the right, the 'HTTP request settings' panel is open, showing configuration for the 'httprequest block'. It includes fields for 'Method' (set to GET), 'URL' (https://api.example.com/data), 'Response variable' (e.g. responseBody), 'Authorization' (None), and 'Headers' (Accept: application/json). Buttons for 'Remove' and 'Add header' are at the bottom of the headers section.

HttpRequest Config
Design blocks and connections with React Flow

CANVAS
+ Add block
Variables

CONTEXT
Right-click canvas or elements for quick actions

Start block
Start

HTTP request block
HTTP request

HTTP request settings
httprequest block

Method: GET

URL: https://api.example.com/data

Response variable: e.g. responseBody

Authorization: None

Headers: Accept: application/json

Remove
Add header

Konfiguracja bloku HTTP request – tryb ciemny

Parser block

Parser Config
Design blocks and connections with React Flow.

CANVAS
+ Add block
F Variables

CONTEXT
Right-click canvas or elements for quick actions.

Start block
Start

Parser block
Parser JSON/XML

Parser settings Close

Parser block

Format JSON (e.g. HTTP response) or XML payload to parse.

JSON

Source variable
e.g. responseBody

Point to the payload variable (e.g. \$http.body or \$responseBody). For JSON arrays, include the array name in the path (e.g. \$.responseBody[0].id).

Mappings (path → variable)

JSON: \$.field\$.items[0].id, XML: /root/items/id. Each mapping writes the matched value into the target variable.

\$ payload id

Variable (e.g. itemId)

Remove

Add mapping

Paths: JSON uses dot/ \$.field style. XML uses XPath-like /root/items/id. Each mapping writes the matched value into the target variable.

Konfiguracja bloku Parser – tryb ciemny

Loop block

The screenshot displays the 'Loop Config' window in a dark-themed application. The interface is divided into three main sections:

- Left Panel:** Contains a 'CANVAS' section with '+ Add block' and 'Variables' buttons, and a 'CONTEXT' section with a right-click action tip.
- Center Canvas:** Shows a workflow starting with a 'START' block (labeled 'Start block') connected by a dashed arrow to a 'LOOP' block (labeled 'Loop block'). The 'LOOP' block has two output ports: 'loop' and 'exit'.
- Right Panel:** Titled 'Loop settings', it features a 'Loop block' header and a 'Iterations' input field set to '3'. Below this, a descriptive text states: 'Loop sends one branch through the loop output per iteration, then exits via the exit output.'

Konfiguracja bloku Loop – tryb ciemny

Wait block

Wait Config
Design blocks and connections with React Flow.

CANVAS

- + Add block
- [-] Variables

CONTEXT
Right-click canvas or elements for quick actions.

Canvas Content:

- START** Ready [Settings] Start block Start
- WAIT** Ready [Settings] Wait block Wait

Wait settings Close

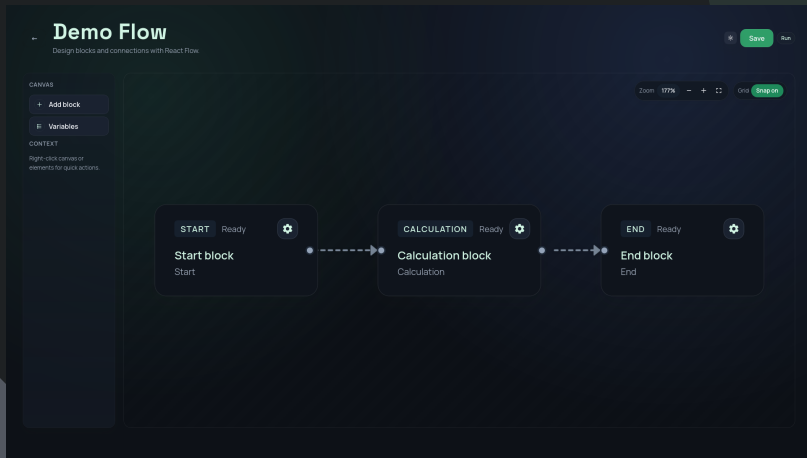
WAIT BLOCK

- Delay (ms)**
2000
- Delay from variable (optional)**
\$delay

If a variable is provided, its value (ms) overrides the numeric delay. Delay is clipped to a safe maximum on execute.

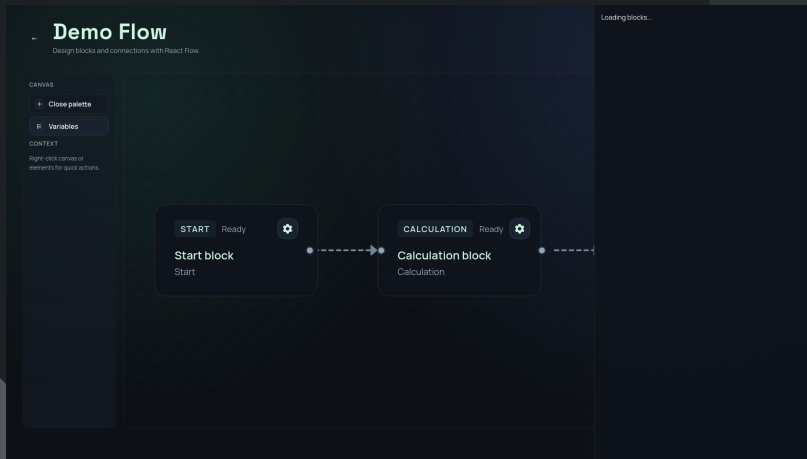
Konfiguracja bloku Wait – tryb ciemny

Workflow editor



Początkowy widok edytora – tryb ciemny

Palette



Paleta dodawania bloków – tryb ciemny

Variables

The screenshot displays the 'Demo Flow' interface, which is a workflow editor. On the left, there is a sidebar with a 'CANVAS' section containing '+ Add block' and 'Hide variables' buttons, and a 'CONTEXT' section with a right-click menu option. The main canvas shows a workflow with two blocks: 'Start block' (labeled 'START' and 'Ready') and 'Calculation block' (labeled 'CALCULATION' and 'Ready'). They are connected by a dashed arrow. On the right, a 'Workflow variables' panel is open, showing a list of variables. The 'limit' variable is highlighted, with a default value of 25. The panel includes an 'Add variable' button and a 'Close' button.

Demo Flow
Design blocks and connections with React Flow.

CANVAS
+ Add block
Hide variables

CONTEXT
Right-click canvas or elements for quick actions.

Workflow variables
Used by Calculation blocks. Close

Add variable
Name: e.g. total
Default value: Optional
Add variable

All variables
limit
Default: 25
Edit Delete

Lista zmiennych (ciemny)

Variables – edycja

The screenshot displays the React Flow editor interface in dark mode. The main canvas shows a workflow with two blocks: a 'Start block' (labeled 'START' and 'Start') and a 'Calculation block' (labeled 'CALCULATION' and 'Calculation'). They are connected by a dashed arrow. The left sidebar contains a 'CANVAS' section with 'Add block' and 'Hide variables' buttons, and a 'CONTEXT' section with a right-click action. The right sidebar is titled 'Workflow variables' and 'All variables'. The 'Workflow variables' section has a 'Close' button and an 'Add variable' button. Below it, there are input fields for 'Name' (with a hint 'e.g. total') and 'Default value' (with a hint 'Optional'). The 'All variables' section shows a list of variables: 'tempVarDark' with a default value of '456'. There are 'Save' and 'Cancel' buttons at the bottom of the 'All variables' section.

Demo Flow
Design blocks and connections with React Flow.

CANVAS
+ Add block
Hide variables

CONTEXT
Right-click canvas or elements for quick actions.

Workflow variables
Used by Calculation blocks. Close

Add variable
Name: e.g. total
Default value: Optional
Add variable

All variables
1 total
Name: tempVarDark
Default value: 456
Save Cancel

Edycja zmiennej (ciemny)

Run – panel danych

Demo Flow
Design blocks and connections with React Flow

CANVAS

- + Add block
- Variables

CONTEXT

Right-click canvas or elements for quick actions

START Ready

Start block
Start

CALCULATION Ready

Calculation block
Calculation

Run workflow Close

Demo Flow

API snippet

HTTP call template

- 1) Replace «API_BASE» with your backend URL.
- 2) Fill variable values in the JSON body.
- 3) Run in terminal (curl) or any HTTP client (Postman).

```
curl -X POST \
  «API_BASE»/api/Workflow/1/run \
  -H "Content-Type: application/json" \
  -d "{}"
```

Response returns execution id, path, actions, and output variables as JSON.

Inputs

0 variables

No workflow variables defined.

☒ Estimated waits ☐ Skip wait blocks (UI only)

Run workflow

Panel Run z rozwiniętymi danymi wejściowymi i snippetem API

Run – wynik uruchomienia

The screenshot displays the 'Run' interface for a workflow named 'Demo Flow'. The interface is divided into three main sections:

- Canvas:** Shows the workflow design with two blocks: 'Start block' (labeled 'START') and 'Calculation block' (labeled 'CALCULATION'). They are connected by a dashed arrow. The 'Start block' has a status of 'Ready' and the 'Calculation block' also has a status of 'Ready'.
- Run workflow panel:** Located on the right, it contains a 'Run workflow' button. Below it, the 'Result' section shows the execution details for 'Execution #900'.
- Result section:** Displays the flow and output variables.
 - FLOW:** A visual representation of the execution path, showing the 'Start' block (labeled 'Start block') and the 'End' block (labeled 'End block') connected by a solid line. Both blocks are marked as 'Executed'.
 - OUTPUT VARIABLES:** A table showing the results of the workflow execution.

Variable	Value	Action
\$ID	900	Copy
RESULT	OK	Copy

Panel Run po wykonaniu workflow – widoczne wyniki i logi

Executions

The screenshot shows the 'Executions' page in the Flowforge interface. On the left is a sidebar with navigation links: 'Workflows', 'Blocks', 'Executions' (highlighted), and 'Scheduler'. The main header area contains the title 'Executions' and a subtitle 'Recent workflow runs and timestamps.' Below this is the 'Execution history' section, which includes a search bar, filters for 'Workflow' (set to 'All workflows') and 'Sort' (set to 'Newest first'), and date range selectors. The history table lists two executions: 'Text Transform' and 'Data Sync', each with a 'View details' button and an 'Open workflow' link. At the bottom left, a status bar indicates 'Connected to local API' and 'proxy: http'.

Flowforge
Workflow Studio

Workflows
Blocks
Executions
Scheduler

Executions

Recent workflow runs and timestamps.

2 total History

Execution history

Search and filter workflow runs by name, date, or recency.

Search
Search by workflow name or id...

Workflow
All workflows

Executed between
mm / dd / yyyy to mm / dd / yyyy

Sort
Newest first

Last 24h Last 7 days Last 30 days

Showing 2 Reset filters

Prev Page 1 of 1 Next Rows 10

Text Transform	Executed: 2/5/2026, 7:09:48 PM	View details	Open workflow
Data Sync	Executed: 12/13/2025, 3:19:00 PM	View details	Open workflow

Connected to local API
proxy: http

Historia wykonania skryptów – tryb ciemny

Execution details

The screenshot displays the 'Execution #36' details page in a dark-themed interface. On the left, a sidebar contains navigation links: 'Workflows', 'Blocks', 'Executions' (which is highlighted), and 'Scheduler'. The main content area is titled 'Execution #36' with a subtitle 'Full run details and outputs'. It includes a timestamp '202006-17 09:18 PM' and a 'Run' button. Below the title is an 'Overview' section with a table showing workflow statistics. The 'Inputs' section displays a JSON object:

```
{  "workflow": "test workflow",  "result": ""}
```

. The 'Results' section shows the output of the workflow, including a 'result' field with the value 'test result 2020'. The 'Path' section visualizes the execution flow from 'Start' to 'Text Transform' to 'End'. The 'Actions' section lists the steps: 'Start block', 'Text Transform (lower -> result)', and 'End block'. At the bottom left, it indicates 'Connected to local API' and '2020/180'.

Pełne szczegóły przebiegu – tryb ciemny

Scheduler

F Flowtange
Workflow Studio

Workflows

Blocks

Executions

Scheduler

Connected to local API

prony.nlp

Scheduler

Planowanie automatycznych uruchomień workflowów.

Nowy harmonogram

Wybierz workflow i ustaw czas startu oraz interwał.

Nazwa	Workflow
np. Nightly ETL	Wybierz...
Typ	Interwał (min)
Interval	60
Start (UTC)	Opis
mm / dd / yyyy, - - - -	np. codziennie o 2:00
Aktywny	
<input checked="" type="checkbox"/>	
Dodaj	

Harmonogramy

Lista zaplanowanych uruchomień.

Harmonogram

Daily

Uruchom teraz

Edytuj

Usuń

Daily sync

Workflow #1

Następnie: —

Ostatnie: —

Lista zaplanowanych zadań – tryb ciemny

Scheduler action – create

The screenshot displays the FlowForge Scheduler interface in dark mode. On the left is a sidebar with navigation links: 'FlowForge workflow studio', 'Workflows', 'Blocks', 'Executions', and 'Scheduler' (which is highlighted). The main area is titled 'Scheduler' with the subtitle 'Planowanie automatycznych uruchomeń workflowów'. It features a 'Nowy harmonogram' (New Schedule) form and a 'Harmonogramy' (Schedules) list.

Nowy harmonogram
Wzrost workflow i ustaw czas startu oraz interwał.

Nazwa	Workflow
np. Nightly ETL	Demo Flow
Typ	Interwał (min)
Interwał	45 <input type="checkbox"/>
Start (UTC)	Czas
min / dd / yyyy, - : -	np. codziennie o 2:00
Aktywny <input checked="" type="checkbox"/>	<button>Dodaj</button>

Harmonogramy
Lista zaplanowanych uruchomeń.

Nightly build Workflow #1 Nazwa: — Czas: —	Interwał: 45 min	Uruchom teraz	Edytuj	<button>Usuń</button>
Daily sync Workflow #1 Nazwa: — Czas: —	Daily	Uruchom teraz	Edytuj	<button>Usuń</button>

Connected to local API
play api

Tworzenie wpisu schedulera – tryb ciemny

Scheduler action – edit

The screenshot displays the 'Scheduler' interface in a dark theme. On the left is a sidebar with navigation links: 'Workflow', 'Backs', 'Executions', and 'Scheduler'. The main area is titled 'Scheduler' and contains two sections: 'Nowy harmonogram' (New schedule) and 'Harmonogramy' (Schedules).

The 'Nowy harmonogram' section is active, showing a form for editing a workflow. The form includes the following fields:

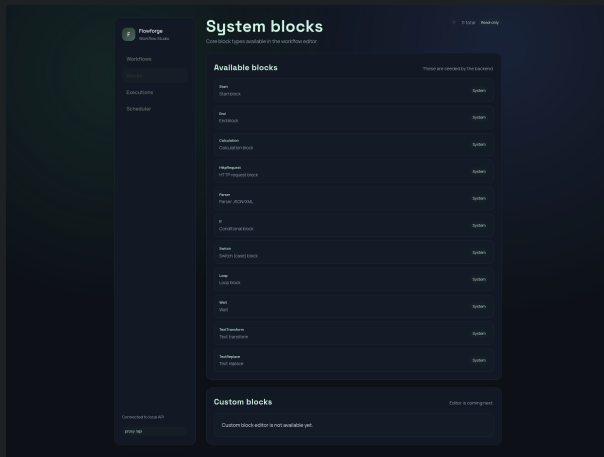
- Name:** Daily sync updated
- Workflow:** Demo Flow
- Type:** Internal (selected)
- Interval (min):** 100
- Start (UTC):** 2024-01-01T00:00:00
- Stop:** 2024-01-01T00:00:00
- Active:** ☒

A green 'Save' button is located at the bottom right of the form.

The 'Harmonogramy' section below shows a list of existing schedules. The first entry is 'Daily sync' with a status of 'Active'. Below this entry, the same form fields are visible, but they are partially obscured by a 'Daily sync' header and a 'Save' button.

Edycja istniejącego wpisu – tryb ciemny

Blocks



Widok bloków systemowych – tryb ciemny