

Low-coding Petriflow apps in Netgrif

This guide will show you how to get started with Netgrif low-code platform. The guide will show you how to use Netgrif platforms to create and deploy applications in low-code language Petriflow.

An applications in Petriflow language is given by a set of Petriflow classes, also called Petriflow processes. When Petriflow application is deployed, a user can create objects of Petri flow classes, also called instances or cases of Petriflow processes.

Petriflow language was designed to make development of process-based web applications easier and faster when compared with traditional full stack development. To do this, Petriflow abstracts from traditional three-layer architecture, when data layer, application layer and presentation layer of web applications are developed separately and then interconnected. Instead, a Petriflow class contains definition of data layer, application layer and presentation layer.

More precisely, a Petriflow class consists of:

- data attributes,
- life cycle of the objects of the class given by a workflow process that consists of tasks,
- forms containing subsets of data attributes that are associated with tasks.

Netgrif low code platform consists of:

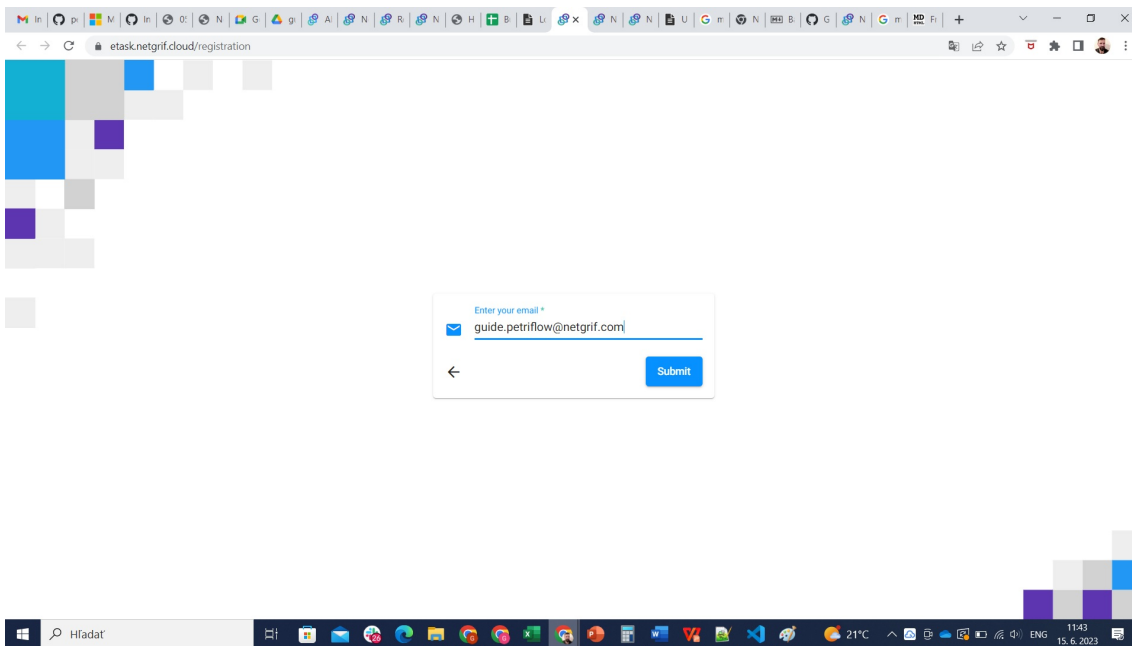
- Netgrif application builder, and
- Netgrif application engine.

Netgrif application builder available for free at <https://builder.netgrif.com> is a tool that supports graphical creation of Petriflow classes by drawing workflows and drag-and-dropping data forms.

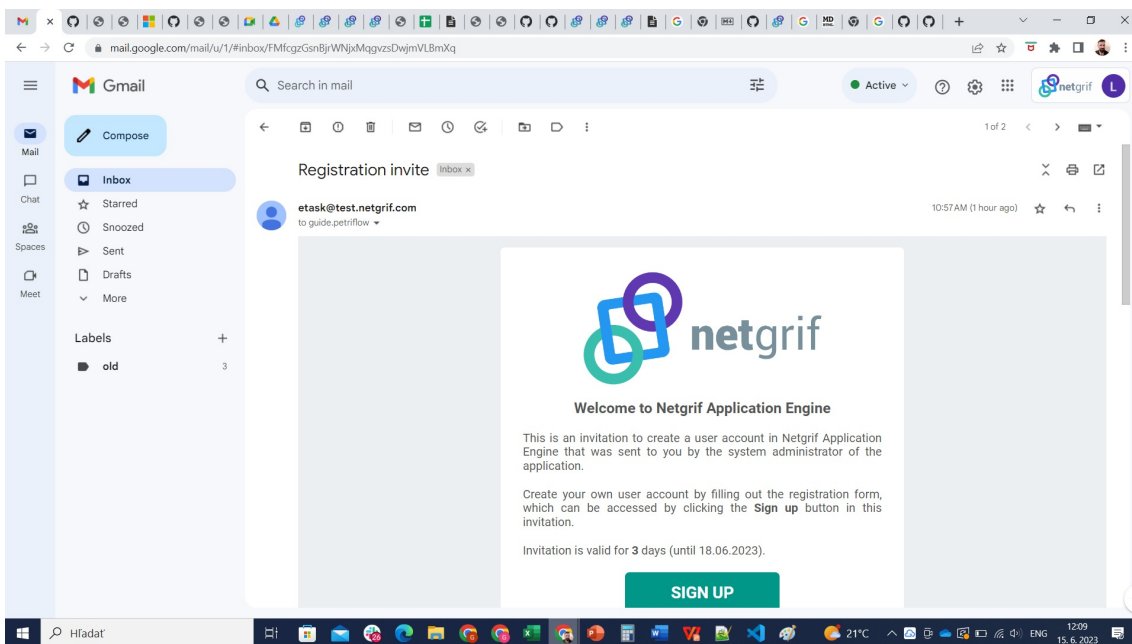
Netgrif application engine is an interpreter of Petriflow applications. It may either be downloaded from github <https://github.com/netgrif> or used as a free service after registration at <https://etask.netgrif.cloud/>.

In this guide we will use <https://builder.netgrif.com> to build examples of Petriflow classes, to describe generated code in Petriflow language and then deploy generated Petriflow classes using <https://etask.netgrif.cloud/>.

If you want to try yourself just sign up using your preferred email at <https://etask.netgrif.cloud/registration>.



You will receive an email



If you click on SING UP button, you will be redirected to the web page, where zou can finish your registration.

etask.netgrif.cloud/signup/Z3VpZGUucGV0cm1mbG93QG5ldGdyYWYy29tOjFhNG0ydw82czBjcHFcWd1azh0aWE4OHROwW3dDhuamNscWduMXRpN29zbWthb2oxNm0

Your email
guide.petriflow@netgrif.com

Name *
Gabriel

Surname *
Juhas

Enter your password *

Repeat your password *

Workspace *
Petriflow Guide

Workspace ID
petriflow-guide

☒ I agree to the [Terms and Conditions](#)*

☒ I would like to receive marketing and sales communication from Netgrif

Register

Windows taskbar: Hfadat, 21°C, 12:17, 15. 6. 2023

As <https://etask.netgrif.cloud/> is a multi-tenant service, here you can choose a unique Wokspace name for your account. We have chosen Petriflow Guide as a name for the account. The Workspace ID will automatically be generated from your Workspace name, in our case the generated Workspace ID is petriflow-guide.

After you fill your name, surname, create your password, agree with the Terms and Condiitiions and press Register button, you will be redirected to the login page.

etask.netgrif.cloud/login

guide.petriflow@netgrif.com

.....

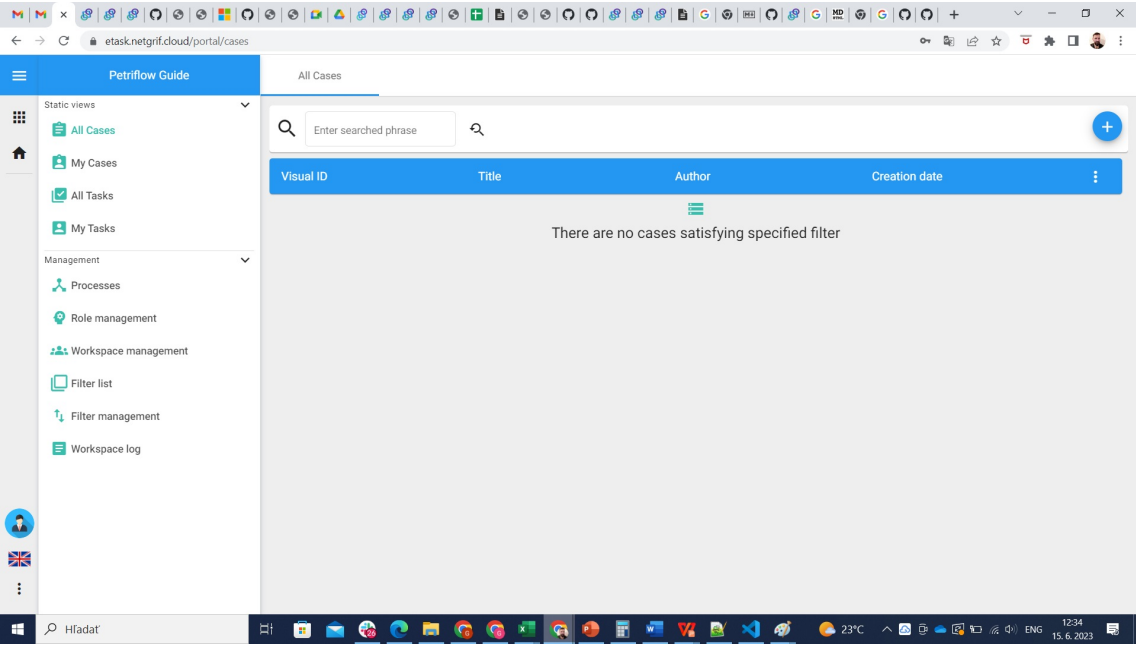
Sign Up Log in

[Reset password](#)

Windows taskbar: Hfadat, 23°C, 12:29, 15. 6. 2023

Here you enter your email and password to Log In your account of Netgrif Etask, which enables you to upload Petriflow classes. Further, users registered to your group (to your workspace) van create

Petriflow objects and perform tasks according to their workflow processes, fill the form associated to tasks etc.



In following paragraphs, we will illustrate how to crate, deploy and use Petriflow applications starting with the most simple case.

Petriflow Web application with form

Let us start with a web application in Petriflow language that consist of just one Petriflow class.

Consider

Wokflowe process -connected by directed arcs with explicit place holders that can be marked by tokens to determine whether tha

bla

bla

```
<data type="number">
  <id>vratit</id>
  <title>Vrátit</title>
  <init>0</init>
</data>
```