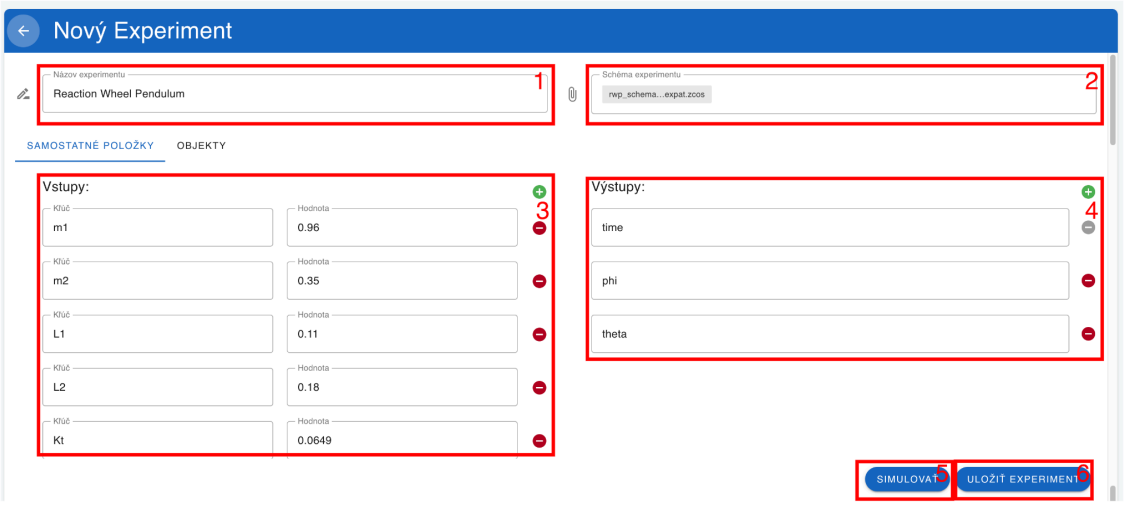
Vytváranie/simulovanie experimentu:

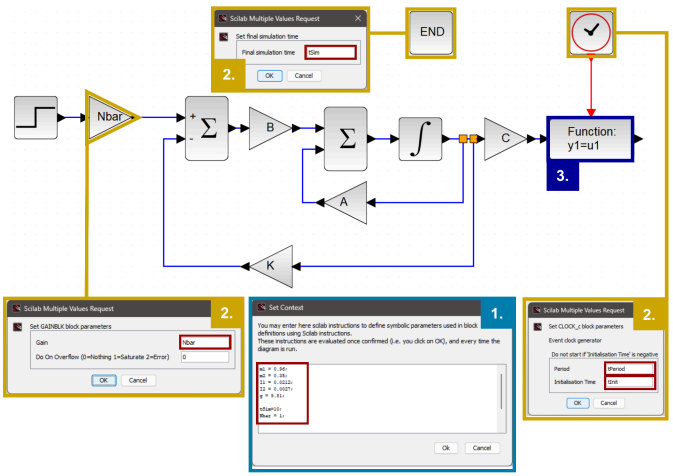


Obr. 1. Používateľské rozhranie aplikácie

1 - Pole pre názov experimentu
2 - Pole na vloženie súboru pre experiment
3 - Definovanie vstupov experimentu
4 - Definovanie názvov pre výstupy experimentu
5 - Tlačidlo na spustenie simulácie
6 - Tlačidlo na uloženie simulácie

Pre vytvorenie alebo simuláciu experimentu je nutné aby boli vyplnené polia názov, súbor a vstupy experimentu. Pri stlačení tlačidla simulovať sa zobrazia výsledky experimentu avšak sa neuložia do úložiska aplikácie. Pri uložení je možné z pamäte tento experiment simulovať viac krát bez nutnosti opätovného nahrávania povinných polí experimentu.

Vstupy experimentu - definujú mapovanie hodnôt v súbore, ktorý definuje správanie experimenu (body 1 a 2 na Obr. 2). Definovanie z aplikácie prebieha na základe pomenovaní premenných.



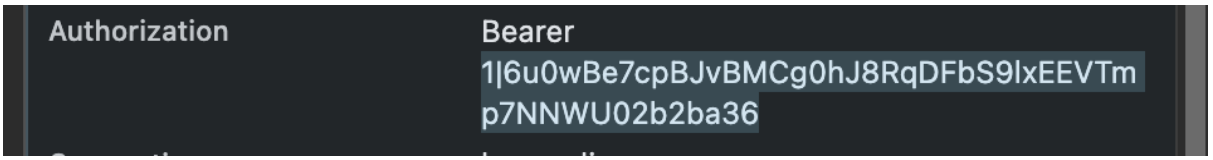
Obr. 2. Zobrazenie súboru experimentu pomocou programu Scilab

Výstupy experimentu - definujú názvy premenných, ktoré vychádzajú zo simulácie (bod 3. na Obr 2.)

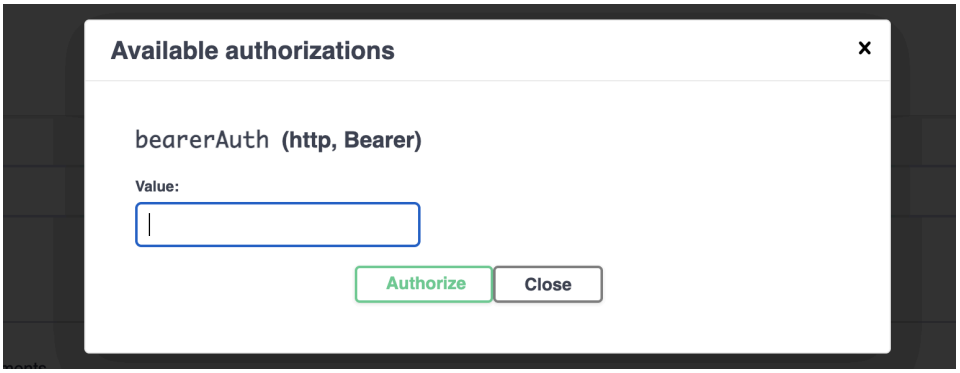
Rest Api:
pre simuláciu je použitý endpoint **/api/experiment**
parametre:

Parameter	Typ	Formát	Povinný	Popis
file	string	binary	✅	Súbor obsahujúci experimentálny
name	string	string	✅	Názov experimentu.
context	string	json	✅	Kontext experimentu vo forme JSON reťazca. Obsahuje parametre simulácie (napr. čas, veličiny, konštanty). Príklad [{"key":"m1","value":"0.96","order":1}, …]
output	string	json	✅	Pole (JSON array ako string) určujúce, ktoré veličiny majú byť súčasťou výstupu. Pri swagger je nutné zadať [{}], v inom prípade ["time", "R", "W", "P"]
save	integer	integer	✅	Určuje, či sa má výsledok uložiť do databázy. Hodnoty: `0` = neukladať, `1` = uložiť.

Pri swagger dokumentácii nastáva problém pri posielaní json hodnôt (context, output), kde swagger neposiela polia ako súvislé reťazce. Pri nastavení hodnoty save na 0 tento endpoint vráti výsledok simulácie. Pre použitie tohto enpointu je nutné zadať Bearer token do hlavičky volania.



Pri použití swagger je tiež nutné skopírovať vyznačenú hodnotu a vložiť do požadovaného pola.



Následne swagger umožní použitie endpointu avšak so spomínaným znevýhodnením.

Príklad volania api cez CURL:

```
curl -X 'POST' \
  'http://localhost:8000/api/experiments' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -H 'X-CSRF-TOKEN: ' \
  -F 'file=@rwp_schema_plus180_expat.zcos' \
  -F 'name=Experiment' \
-F'context=[{"key":"m1","value":"0.96","order":1},{"key":"m2","value":"0.35","order":2},{"k
ey":"L1","value":"0.11","order":3},{"key":"L2","value":"0.18","order":4},{"key":"Kt","value
":"0.0649","order":5},{"key":"Ke","value":"0.0649","order":6},{"key":"Ng","value":"1","orde
r":7},{"key":"Rm","value":"6.83","order":8},{"key":"I1","value":"0.0212","order":9},{"key":
"I2","value":"0.0027","order":10},{"key":"g","value":"9.81","order":11},{"key":"a","value":
"m1*L1*L1 + m2*L2*L2 + I1","order":12},{"key":"b","value":"(m1*L1 + m2*L2) *
g","order":13},{"key":"b2","value":"-Kt*Ng/(a*Rm)","order":14},{"key":"b4","value":"(a+I2)*
Kt*Ng/(a*I2*Rm)","order":15},{"key":"a44","value":"-b4*Ke*Ng","order":16},{"key":"A","value
":"[0 1 0 0;b/a 0 0 Kt*Ke*Ng*Ng/(a*Rm);0 0 0 1;-b/a 0 0
a44]","order":17},{"key":"B","value":"[0;b2;0;b4]","order":18},{"key":"C","value":"[1 0 0
0;0 0 1
0]","order":19},{"key":"D","value":"[0;0]","order":20},{"key":"K","value":"[-601.0636
-99.5773 -0.9211
-1.5320]","order":21},{"key":"Nbar","value":"1","order":22},{"key":"x0","value":"[90; 0; 0;
0]","order":23},{"key":"tSim","value":"15","order":24},{"key":"tInit","value":"0","order":2
5}]' \
  -F 'output=[“time”,”velocity”,”height”']’ \
  -F 'save=0'
```

Pri programatickom riešení je nutné vykonať na akúkoľvek prácu s aplikáciou 2 volania:

Príklad použitia v php:

1. Prihlásenie sa do aplikácie:

```
$url = $environment->url . '/api/auth/login';
$login = YOUR_SCILAB_LOGIN;
$password = YOUR_SCILAB_PASSWORD;

$response = Http::post($url, [
      'email' => $login,
      'password' => $password,
]);
$token = $response->json()['token']
```

2. Simulácia experimentu bez ukladania do scilab databázy

```
$url = $environment->url . '/api/experiments'
$response = Http::withHeaders([
      'Authorization' => 'Bearer ' . $token,
      'accept' => 'application/json'
])->attach(
      'file',base64_decode(file), 'simulation.zcos'
)->post($url, [
      'name' => $experiment->title,
      'context' => json_encode(YOUR_CONTEXT),
      'output' => json_encode(YOUR_OUTPUTS),
      'save' => 0,
```

```
    ]);
```

Príklad použitia v JavaScript-e:
fetch:

```
async function login(environment, login, password) {
    const url = `${environment.url}/api/auth/login`;

    const response = await fetch(url, {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify({
            email: login,
            password: password
        })
    });

    const data = await response.json();
    return data.token;
}

async function simulateExperiment(environment, token, experiment,
context, outputs, base64file) {
    const url = `${environment.url}/api/experiments`;

    const form = new FormData();

    form.append("file", Buffer.from(base64file, "base64"),
"simulation.zcos");
    form.append("name", experiment.title);
    form.append("context", JSON.stringify(context));
    form.append("output", JSON.stringify(outputs));
    form.append("save", 0);

    const response = await fetch(url, {
        method: "POST",
        headers: {
            "Authorization": `Bearer ${token}`,
            "accept": "application/json"
        },
        body: form
    });

    const data = await response.json();
    return data;
```

```
}

(async () => {
    const environment = { url: "https://tvoja-api-url.com" };

    const loginName = "YOUR_SCILAB_LOGIN";
    const password = "YOUR_SCILAB_PASSWORD";

    const token = await login(environment, loginName, password);

    const experiment = { title: "Moja simulácia" };
    const context = {};
    const outputs = {};

    const base64file = "BASE64_OBSAH_ZCOS";

    const result = await simulateExperiment(
        environment,
        token,
        experiment,
        context,
        outputs,
        base64file
    );

    console.log(result);
})();
```

axios:

```
const axios = require('axios');
const FormData = require('form-data');

async function login(environment, login, password) {
    const url = `${environment.url}/api/auth/login`;
    const response = await axios.post(url, {
        email: login,
        password: password
    });
    return response.data.token;
}

async function simulateExperiment(environment, token, experiment,
context, outputs, base64file) {
    const url = `${environment.url}/api/experiments`;
    const form = new FormData();
    form.append('file', Buffer.from(base64file, 'base64'),
'simulation.zcos');
```

```
    form.append('name', experiment.title);
    form.append('context', JSON.stringify(context));
    form.append('output', JSON.stringify(outputs));
    form.append('save', 0);

    const response = await axios.post(url, form, {
        headers: {
            ...form.getHeaders(),
            Authorization: `Bearer ${token}`,
            Accept: 'application/json'
        }
    });

    return response.data;
}

(async () => {
    const environment = { url: "https://tvoja-api-url.com" };
    const loginName = "YOUR_SCILAB_LOGIN";
    const password = "YOUR_SCILAB_PASSWORD";

    const token = await login(environment, loginName, password);

    const experiment = { title: "Moja simulácia" };
    const context = { };
    const outputs = { };

    const base64file = "BASE64_OBSAH_ZCOS";

    const result = await simulateExperiment(
        environment,
        token,
        experiment,
        context,
        outputs,
        base64file
    );

    console.log(result);
})();
```

Príklad použitia v Python-e:

```
import base64
import requests

def login(environment, login, password):
    url = f"{environment['url']}/api/auth/login"
```

```
    response = requests.post(url, json={
        "email": login,
        "password": password
    })
    response.raise_for_status()
    return response.json().get("token")

def simulate_experiment(environment, token, experiment, context,
outputs, base64file):
    url = f"{environment['url']}/api/experiments"

    file_bytes = base64.b64decode(base64file)

    headers = {
        "Authorization": f"Bearer {token}",
        "accept": "application/json"
    }

    files = {
        "file": ("simulation.zcos", file_bytes)
    }

    data = {
        "name": experiment["title"],
        "context": json.dumps(context),
        "output": json.dumps(outputs),
        "save": 0
    }

    response = requests.post(url, headers=headers, files=files,
data=data)
    response.raise_for_status()
    return response.json()

if __name__ == "__main__":
    import json

    environment = { "url": "https://tvoja-api-url.com" }

    login_name = "YOUR_SCILAB_LOGIN"
    password = "YOUR_SCILAB_PASSWORD"

    token = login(environment, login_name, password)

    experiment = { "title": "Moja simulácia" }
    context = {}
    outputs = {}
```

```
base64file = "BASE64_OBSAH_ZCOS"

result = simulate_experiment(
    environment,
    token,
    experiment,
    context,
    outputs,
    base64file
)

print(result)
```