

Cvičení v prostředí Google Earth Engine

Úvod do práce v Google Earth Engine

Dálkový průzkum Země

OBSAH

1	TEORETICKÁ ČÁST	4
1.1	GOOGLE EARTH ENGINE	4
	Prostředí Google Earth Engine.....	4
2	DATA	6
2.1	SENTINEL-2	6
3	PRAKTICKÁ ČÁST.....	7
3.1	POSTUP PŘIHLÁŠENÍ DO GOOGLE EARTH ENGINE.....	7
3.2	ZÁKLADY JAZYKA JAVASCRIPT.....	11
	3.2.1 <i>Proměnné</i>	11
	3.2.2 <i>Seznamy a slovníky</i>	11
	Seznam.....	11
	Slovník.....	12
	3.2.3 <i>Definování funkcí</i>	12
	3.2.4 <i>Komentáře</i>	13
3.3	PRÁCE S IMAGE.....	14
	1. Založení nového repositáře a projektu.....	14
	2. Zakreslení zájmové oblasti.....	16
	3. Import dat.....	16
	4. Příprava dat	17
	5. Vizualizace snímku	18
	6. NDVI index.....	20
	7. Vizualizace NDVI indexu	21
	8. Export snímku.....	21
	9. Výběr bodů pro spektrální křivku.....	22
	10. Přejmenování pásem	25
	11. Vykreslení spektrální křivky	26
3.4	PRÁCE S IMAGECOLLECTION.....	28
	1. Založení nového skriptu	28
	2. Import potřebných dat	28
	3. Příprava dat	28

4.	Vizualizace kolekce.....	29
5.	NDVI.....	30
6.	Vizualizace NDVI.....	31
7.	Vývoj hodnot NDVI v celé zájmové oblasti.....	31
8.	Vývoj hodnot NDVI pro konkrétní pole.....	32

1 TEORETICKÁ ČÁST

1.1 Google Earth Engine

Google Earth Engine je cloudová platforma pro práci s družicovými snímky, která umožňuje efektivně vizualizovat výsledky naší práce. Pro práci v prostředí Google Earth Engine není nutné nic stahovat, instalovat a průběžně aktualizovat, ale stačí mít pouze připojení k internetu, webový prohlížeč a vlastní Google účet.

Jednou z překážek však může být nutnost psát vlastní skript, a to v jazyce JavaScript. I když psaní vlastních skriptů v jazyce JavaScript může mnoho lidí odradit, tak samotní vývojáři Google Earth Engine poskytují užitečné návody a rady, které jsou zdarma dostupné na internetu.

Značnou výhodou představuje zejména rozsáhlý datový katalog, který nám umožňuje nahrávat geografická data z cloudu přímo do našeho vlastního pracovního prostředí. Můžeme tak stahovat rozsáhlé datové sady a následně je zpracovávat. Lze provádět detekci změn nebo tvořit časové řady, což by bez silného hardwarového základu nebylo běžně možné. Zmíněný datový katalog v sobě skrývá satelitní snímky, meteorologická data, data o terénu nebo pokryvu země.

Díky bohatosti nástrojů a dat si tato platforma najde uplatnění v ekologii, lesnictví, zemědělství nebo v urbanistice.

Prostředí Google Earth Engine

Prostředí Google Earth Engine je velmi přehledné a poskytuje nám vše, co potřebujeme pro zpracování družicových snímků.



Pro naši práci bude stěžejní zejména vyhledávací okno, editor kódu, mapové okno a konzole. Vyhledávací okno nám umožňuje vyhledávat lokality v mapovém okně, ale hlavně vyhledávat datové sady v datovém katalogu a následně je nahrávat do našeho pracovního prostředí. Editor kódu slouží k psaní našeho vlastního kódu v programovacím jazyce JavaScript, který následně spouštíme pomocí ikony *Run*. Výsledky našeho skriptu se pak zobrazí v mapovém okně nebo v případě textového výstupu či grafu se zobrazí v konzoli.

Dále nás může zajímat v levé části pracovního prostředí záložka *Scripts*, kde se ukládají naše skripty, sekce *Docs*, ve které lze nalézt popis nástrojů Google Earth Engine.

S dalšími částmi prostředí Google Earth Engine se blíže se seznámíme v rámci praktické části cvičení.

2 DATA

V rámci tohoto cvičení budeme pracovat s daty z družice Sentinel-2.

2.1 Sentinel-2

Jedná se o misi programu Copernicus. Sentinel-2 nese multispektrální senzor, který snímá sluneční záření odražené od povrchu Země. Data jsou získávána v 13 spektrálních pásmech ve viditelném, blízkém infračerveném a krátkovlnném infračerveném spektru.

Na oběžné dráze se pohybují dvě identické družice Sentinel-2A a Sentinel-2B a díky této soustavě je časové rozlišení dat 5 dní. V mírných šířkách je časové rozpětí v případě dvou družic dokonce 2-3 dny.

V závislosti na spektrálním rozlišení jsou snímky poskytovány v prostorovém rozlišení 10, 20 a 60 m. Čtyři kanály v prostorovém rozlišení 10 m (B2, B3, B4, B8), šest kanálů v prostorovém rozlišení 20 m (B5, B6, B7, B8a, B11, B12) a tři kanály v prostorovém rozlišení 60 m (B1, B9, B10)

Sentinel-2 Bands	Central Wavelength (μm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

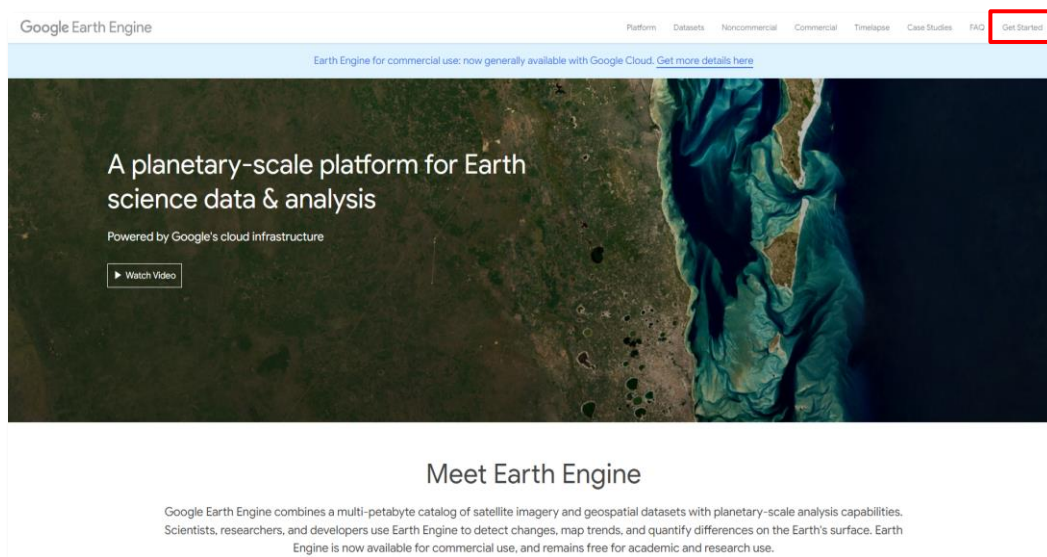
3 PRAKTICKÁ ČÁST

V rámci praktické sekce cvičení se postupně podíváme na postup přihlášení do prostředí Google Earth Engine, osvětlíme si základy jazyka JavaScript a samozřejmě si i vyzkoušíme práci s družicovými snímky. Veškeré skripty používané na dnešním cvičení si můžete zobrazit ve svém prostředí Google Earth Engine pomocí odkazu https://code.earthengine.google.com/?accept_repo=users/michaelasvehlikova/cviceni.

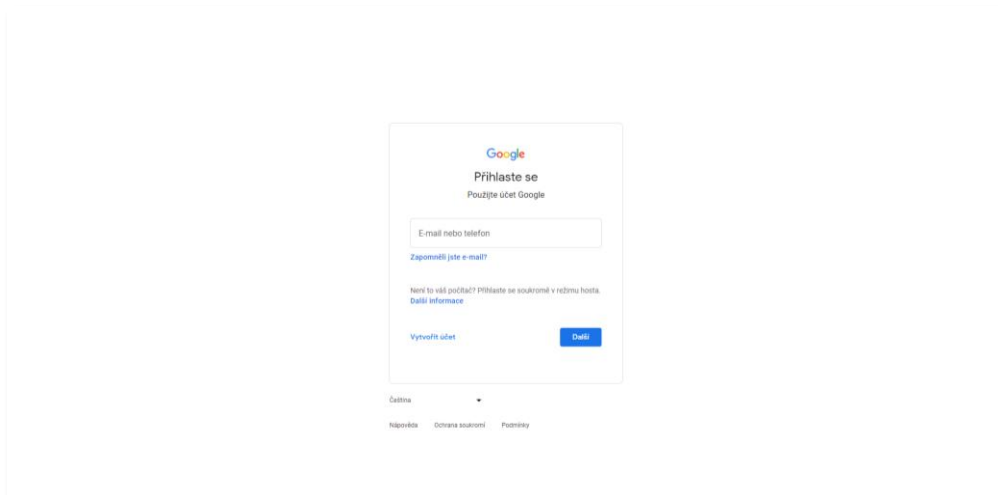
3.1 Postup přihlášení do Google Earth Engine

Jak již bylo zmíněno v teoretickém úvodu, tak pro práci v Google Earth Engine nám postačí připojení k internetu, webový prohlížeč a vlastní Google účet. Abychom tedy mohli začít v tomto prostředí pracovat je nutné se do něj přihlásit a založit si svůj první projekt.

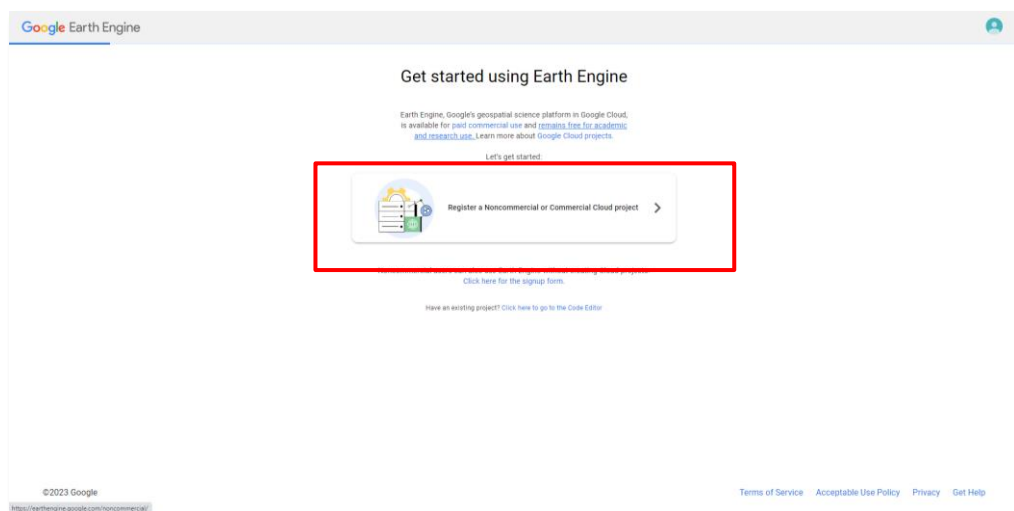
Jako první přejdeme na stránku <https://earthengine.google.com/>. Po načtení úvodní stránky Google Earth Engine najdeme v pravém horním rohu možnost *Get Started* a klikneme na ni.



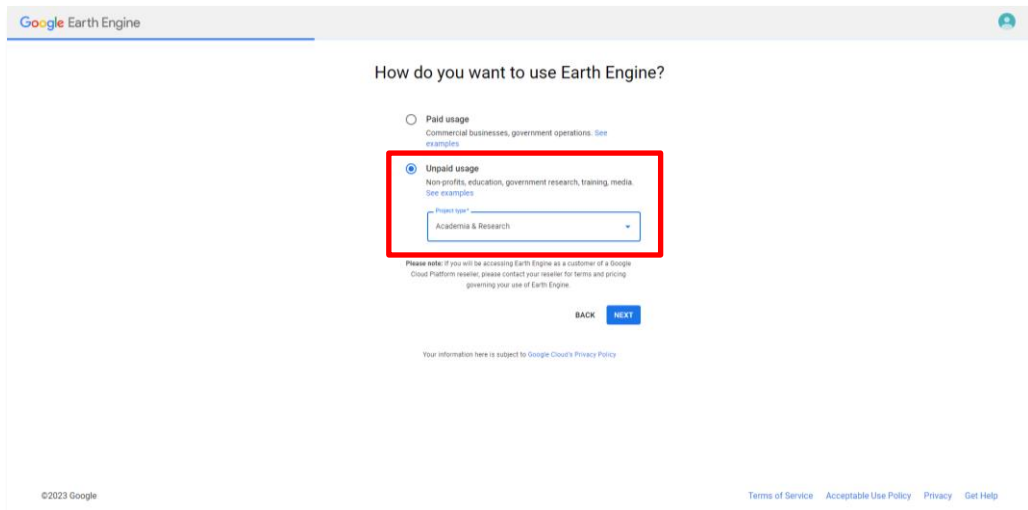
Budeme přesměrováni na přihlášení pomocí Google účtu. Vyplníme tedy náš e-mail a následně heslo. V případě, že Google účet nemáte, tak je nutné si jej vytvořit.



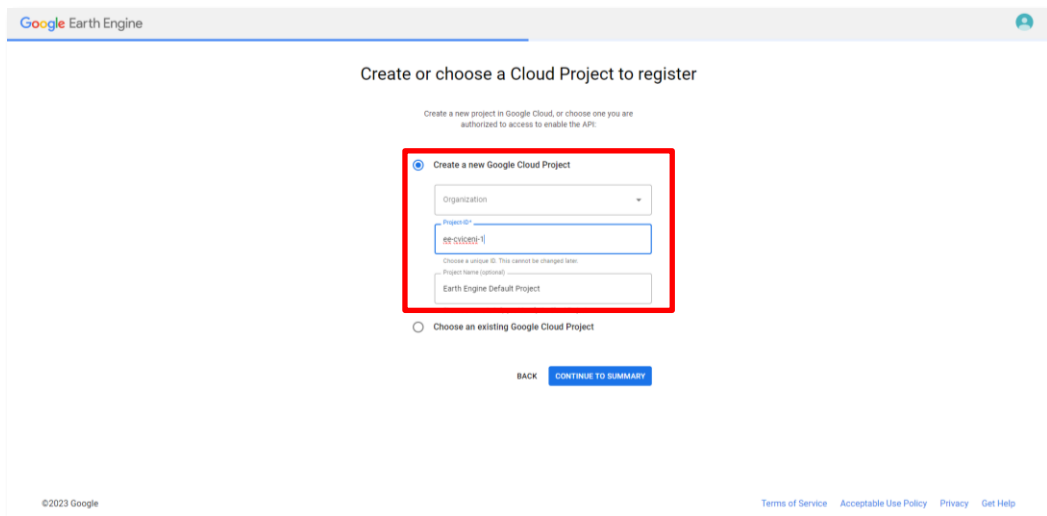
Po přihlášení do Googlu už následuje založení našeho projektu. Vybereme jedinou nabízenou možnost *Register a Noncommercial or Commercial Cloud project*.



Google Earth Engine je zdarma pro akademické účely. V dalším kroku vybereme možnost *Unpaid usage* a z rozbalovacího menu zvolíme možnost *Academia & Research*. Poté už jen klikneme na tlačítko *Next*, abychom přešli k dalšímu kroku.



Vzhledem k tomu, že jsme ještě v prostředí Google Earth Engine nepracovali, tak musíme vytvořit zcela nový projekt. Volíme tedy možnost *Create a new Google Cloud Project* a můžeme vyplnit ID projektu a název projektu, či ponechat defaultní nastavení. Po nastavení požadovaných parametrů, klikneme na tlačítko *Continue to summary*.



Na závěr už jen zkontrolujeme nastavení našeho projektu. Vše potvrdíme pomocí tlačítka *Confirm* a budeme přesměrováni do prostředí Google Earth Engine.

Confirm your Cloud Project information

Project usage
Academia & Research 

Project info
ee-civildm
Earth Engine Default Project 

BACK

CONFIRM

Project information cannot be changed later.

3.2 Základy jazyka JavaScript

Pro práci v Google Earth Engine je nutná znalost programovacího jazyka JavaScript. Je nezbytné si tedy představit jeho základy předtím, než začneme se zpracováním družicových snímků. Jednotlivé bloky kódu si zkopírujte či přepište do editoru kódu a spusťte.

3.2.1 Proměnné

K definování proměnných se používá klíčové slovo **var**, které je nutné uvést vždy před názvem proměnné. Po klíčovém slovu **var** následuje samotné jméno proměnné. Při pojmenování samozřejmě nepoužíváme diakritiku ani klíčová slova JavaScriptu. Ideálně si však volíme názvy, které jsou vypovídající a srozumitelné, tudíž po přečtení názvu proměnné získáme představu o tom, co daná proměnná reprezentuje.

V případě, že je hodnota naší proměnné textový řetězec je nutné ji uvést do uvozovek. Pokud se jedná o číselnou hodnotu, tak není třeba psát uvozovky.

V následující ukázce si můžete všimnout středníku na konci definice proměnných. Středník symbolizuje ukončení jednoho příkazu. Není nutné jej psát, ale je dobrou praxí jej používat. Samotný Google Earth Engine vás upozorní na chybějící středník, ale i tak skript proběhne bez chybových hlášek.

Následně si pomocí příkazu **print()** můžeme vytisknout hodnotu vybrané proměnné, která se nám zobrazí v konzoli.

```
1. var jmeno = "Ostrava"; // textový řetězec
2. var cislo = 1; // číselná hodnota
3.
4. print(jmeno);
5. print("Olomouc");
```

3.2.2 Seznamy a slovníky

Seznam

Jako první si definujeme seznam. Opět začínáme klíčovým slovem **var** a s názvem seznamu. Obsah našeho seznamu ukládáme do hranatých závorek a jednotlivé hodnoty v něm oddělujeme čárkou.

Položky v seznamu jsou indexovány od 0 a pomocí indexů lze k jednotlivým položkám v seznamu přistupovat.

Pomocí **print()** si můžeme nechat vypsat položku na zadaném indexu či celý seznam.

```
1. // Definování seznamu a přístup k jeho položkám
2. var seznam = ["Ostrava", "Brno", "Olomouc"];
3. print(seznam[0]); // první položka v seznamu
4. print(seznam); // celý seznam
```

Slovník

V případě slovníku ukládáme obsah do složených závorek. Do slovníku ukládáme vždy dvojici klíč a hodnota.

K jednotlivým položkám přistupujeme pomocí klíče, který uvádíme do hranatých závorek nebo můžeme využít tečkovou notaci.

```
1. // Definování slovníku a přístup k jeho položkám
2. var slovník = {"kočka": "cat", "pes": "dog", "kráva": "cow"};
3. print(slovník["kočka"]); // klíč v hranatých závorkách
4. print(slovník.pes); // tečková notace
```

3.2.3 Definování funkcí

Funkce představují bloky kódu, které se často opakují. Je totiž zbytečné psát stejný kód několikrát, a proto si definujeme funkci. Vede to k zjednodušení a větší přehlednosti kódu.

Pro definici funkce je nutné začít opět s klíčovým slovem **var** a názvem funkce. Poté použijeme klíčové slovo **function** a kulaté závorky, do kterých uvádíme parametry naší funkce. Samotné znění funkce udáváme do složených závorek. Pokud definujeme funkci s návratovou hodnotou, tak použijeme klíčové slovo **return**, za které uvedeme podobu návratové hodnoty. V případě ukázky se bude jednat o součet dvou čísel.

Funkci na sčítání si můžete vyzkoušet na vlastním zadání pomocí **print()** a vlastních vstupních parametrů.

```
1. var scitani = function (a, b) {
2.   return a+b;
3. };
4.
5. print(scitani(1,2));
```

3.2.4 Komentáře

Je vhodné náš kód průběžně komentovat pro lepší pochopení a orientaci. V JavaScriptu se pro jednořádkové komentáře používají dvě lomítka a po nich následuje komentář. V případě, že je náš komentář delší než jeden řádek, tak využíváme kombinaci lomítka a hvězdičky.

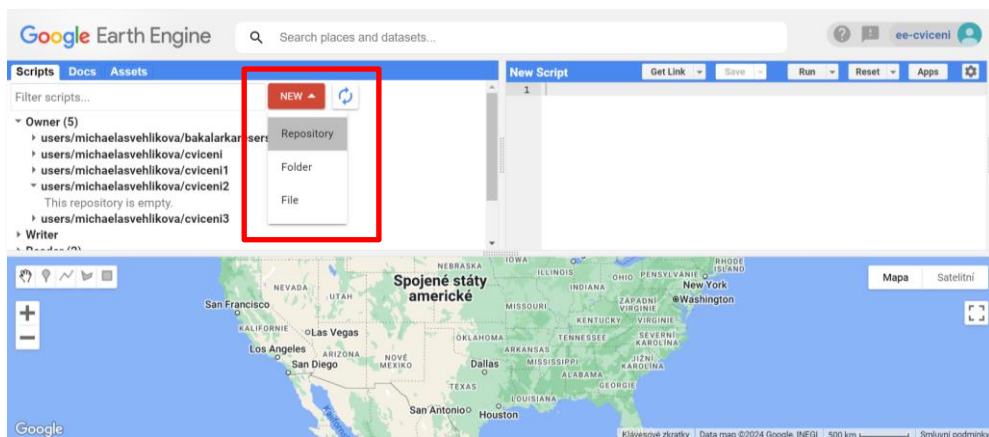
```
1. // Jednořádkové komentáře v JavaScriptu
2.
3. /*
4. Pokud chceme zakomentovat delší blok kódu, tak využijeme kombinaci hvězdičky a lomítka.
5. Můžeme tedy tvořit komentáře, které jsou delší jak jeden řádek
6. */
```

3.3 Práce s Image

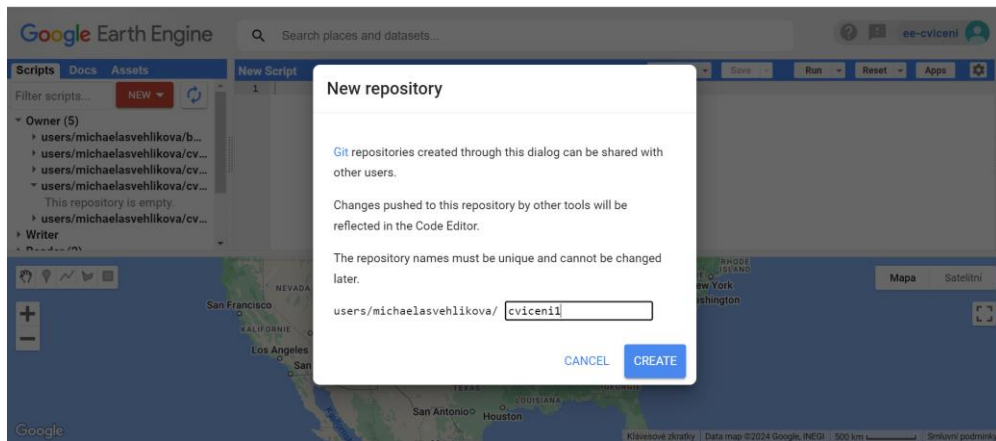
Už jsme se seznámili s prostředím Google Earth Engine a vyzkoušeli si základy jazyka JavaScript, a tudíž nám nic nebrání v tom, abychom se vrhli na práci s družicovými snímky. Jako první si vyzkoušíme práci s Image, tedy s jedním družicovým snímkem. Doporučuji jednotlivé bloky kódu kopírovat a následně upravit podle svých potřeb. Není důležité umět napsat celý kód z hlavy, ale pochopit o co se v něm jedná a jak docílit požadovaného výsledku.

1. Založení nového repozitáře a projektu

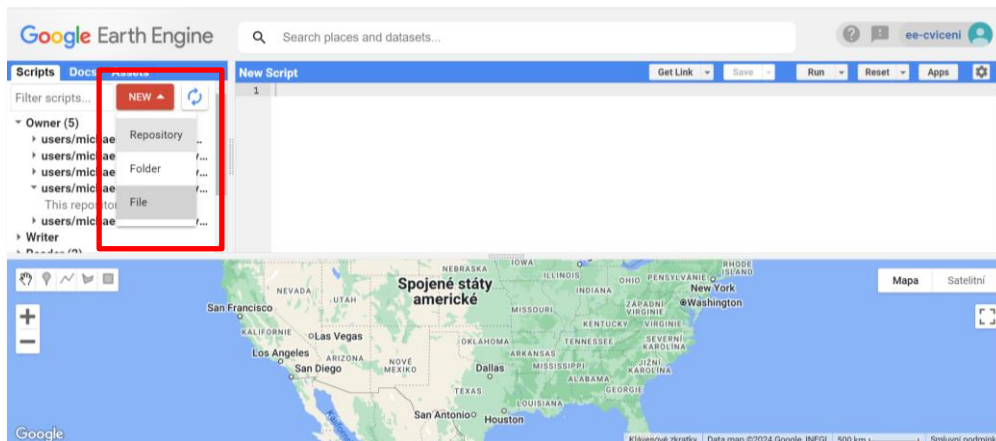
Před zahájením práce si vytvoříme nový repozitář, do kterého budeme ukládat skripty z prvního cvičení v prostředí Google Earth Engine. Vybereme ikonu *NEW* v záložce *Scripts* v levé části pracovního prostředí. Z rozbalovacího menu zvolíme možnost *Repository*.



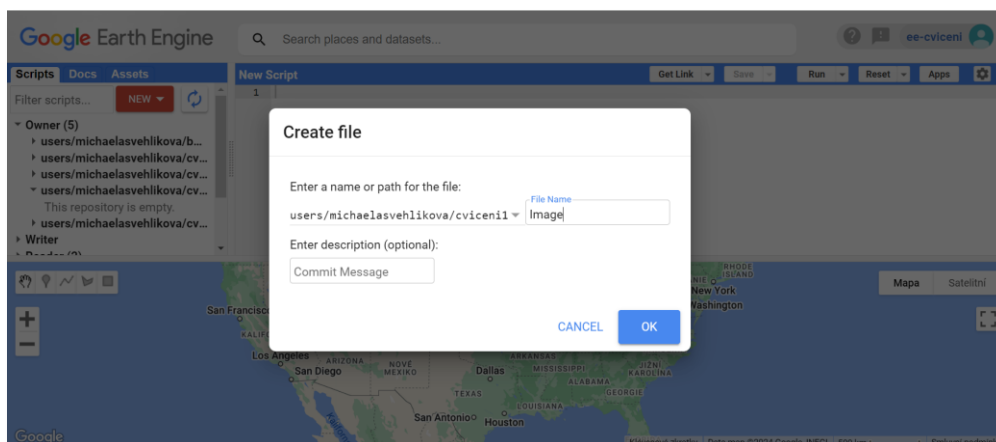
Teď už jen pojmenujeme náš nový repozitář. Vzhledem k tomu, že se jedná o první cvičení v Google Earth Engine, tak volíme název **cviceni1**. Vytvoření nového repozitáře nakonec potvrdíme pomocí tlačítka *CREATE*.



Do našeho nového repositáře vytvoříme nový skript a můžeme zahájit práci. Opět přejdeme na ikonu *New* v sekci *Scripts* a zvolíme možnost *File*.





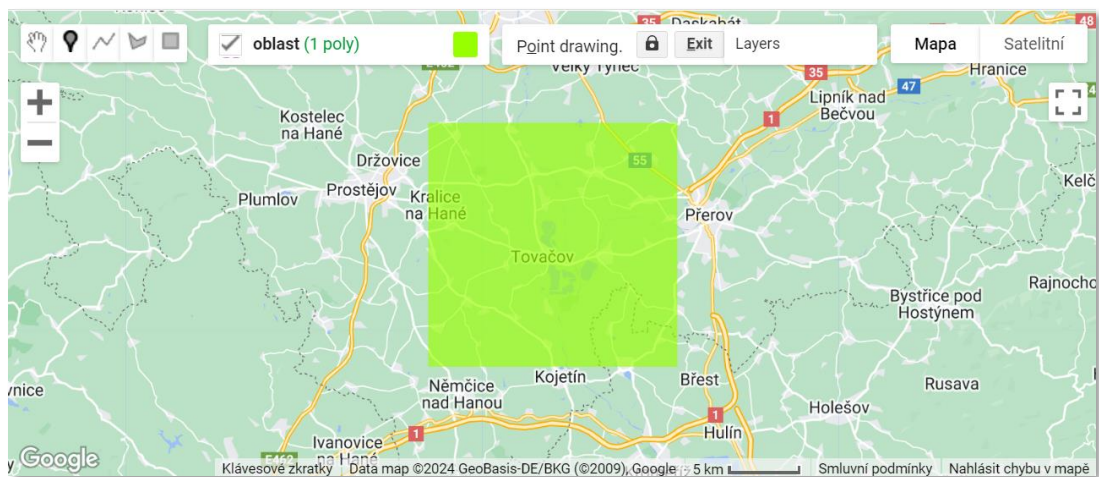
Zkontrolujeme, že se náš nový skript uloží do požadovaného repositáře a skript pojmenujeme. Tvorbu nového skriptu dokončíme kliknutím na tlačítko *OK*.



2. Zakreslení zájmové oblasti

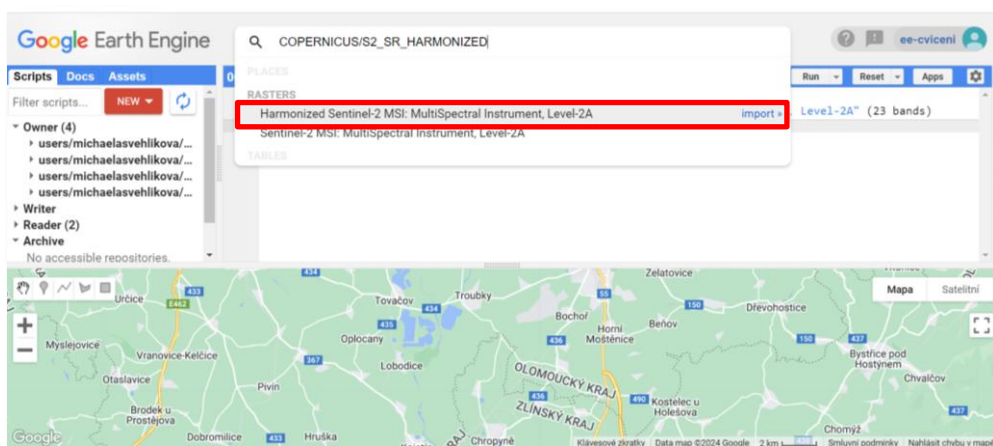
Jako první bude naším úkolem zakreslit zájmovou oblast na mapě. Je zcela na vás, jakou oblast si vyberete, můžete si zvolit např. místo svého bydliště či studia. Hlavní je, aby se ve vybrané oblasti nacházely veškeré druhy povrchu, a to pole, vegetace, zástavba a vodní plochy.

V mapovém okně nalezneme zvolenou zájmovou oblast a pomocí nástroje na zakreslení libovolného tvaru  či obdélníku  v levé horní části mapového okna ji vyznačíme. V mapovém okně se následně zobrazí zvolená oblast a bude zároveň přidána do našeho pracovního prostředí.

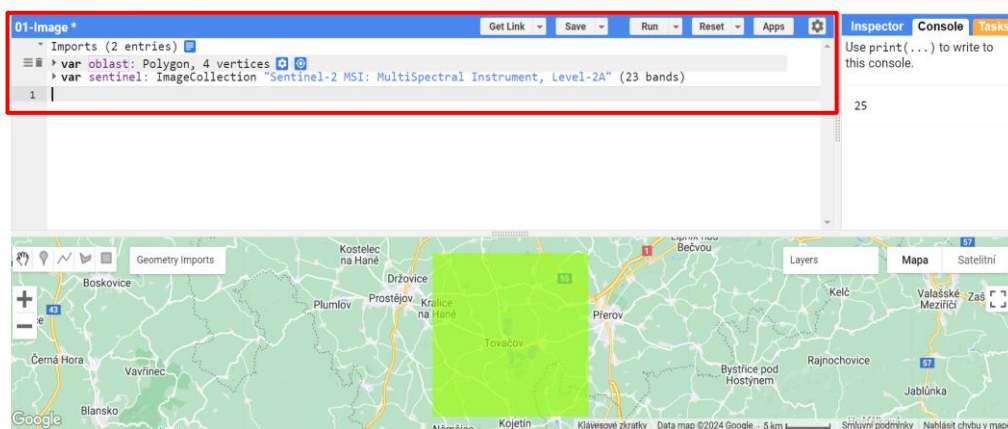


3. Import dat

Pomocí vyhledávací lišty najdeme datovou sadu s názvem COPERNICUS/S2_SR_HARMONIZED a importujeme ji do našeho pracovního prostředí.



Po zakreslení oblasti a nahrání dat ze Sentinelu-2 by se nám v horní části editoru kódu měly objevit dvě nové proměnné. Jednotlivé proměnné si můžeme přejmenovat nebo ponechat tak, jak jsou.



4. Příprava dat

Zatím jsme si nahráli celou kolekci snímků z družice Sentinel-2 bez ohledu na území, časové období nebo oblačnost. V této sekci je, ale našim cílem naučit se pracovat pouze s jedním snímkem pro zvolené území.

Nejdříve si vyfiltrujeme celou nahranou kolekci snímků pomocí funkce **filterDate()** a do závorek uvedeme požadované časové období. Dále potřebujeme, aby snímky pokrývaly naše zájmové území, proto využijeme funkci **filterBounds()** a jako parametr uvedeme název naší zakreslené zájmové oblasti. Snímky seřadíme podle oblačnosti od nejmenší po největší. Pro seřazení snímků využijeme funkci **sort()** a atribut snímků s názvem

CLOUDY_PIXEL_PERCENTAGE. Ze seřazených snímků vybereme ten s nejmenší oblačností pomocí funkce **first()**.

```
1. // Nalezení snímku pro dané období a oblast a seřazení dle oblačnosti
2. var snimekSentinel = sentinel
3.     .filterDate("2023-09-01", "2023-10-31")
4.     .filterBounds(oblast)
5.     .sort("CLOUDY_PIXEL_PERCENTAGE")
6.     .first();
```

Posledním krokem před vizualizací bude ořez vybraného snímku na rozsah zájmové oblasti. Náš snímek s nejmenší oblačností ořežeme díky metodě **clip()**, které zadáme zájmovou oblast.

```
1. // Ořezání zvoleného snímku na zájmovou oblast
2. var orezSnimek = snimekSentinel.clip(oblast);
```

5. Vizualizace snímku

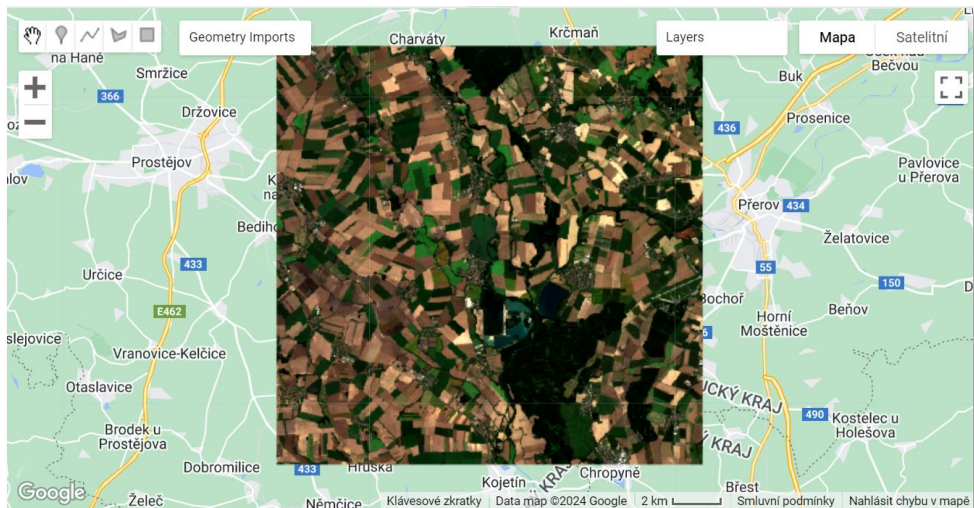
Náš snímek už je připravený k tomu, abychom si mohli zobrazit kompozice v pravých a nepravých barvách.

Pomocí **Map.addLayer()** říkáme mapě, že má vložit novou vrstvu. Mezi parametry této funkce patří snímek, který chceme zobrazit, parametry zobrazení, a nakonec název nové vrstvy v mapovém okně, který uvádíme do uvozovek.

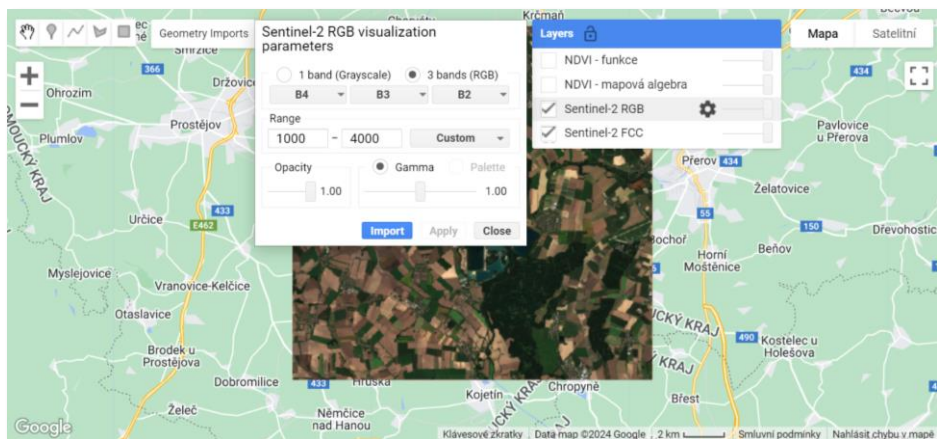
Map.centerObject() nám zajistí, že při vykreslování vrstev v mapě dojde zároveň k jejich přiblížení. Jako parametr vkládáme objekt, který chceme přiblížit a velikost zoomu. Zoom lze zadat v rozsahu hodnot 0-24, kdy 0 představuje nejmenší možné přiblížení.

Výsledkem následujícího bloku kódu je kompozice v pravých a nepravých barvách.

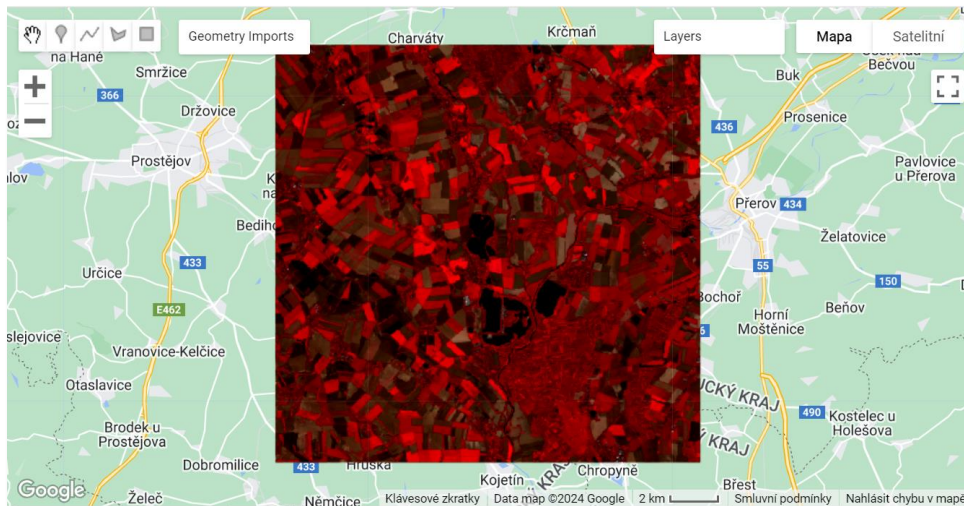
```
1. // Vizualizace snímku v pravých a nepravých barvách
2. Map.addLayer(snimekSentinel, {bands: ["B8", "B4", "B3"], min: 1200, max: 6000}, "Sentinel-2 FCC");
3. Map.addLayer(snimekSentinel, {bands: ["B4", "B3", "B2"], min: 1000, max: 4000}, "Sentinel-2 RGB");
4.
5. // Nastavení zobrazení a přiblížení snímku v mapovém okně
6. Map.centerObject(snimekSentinel, 10);
```



V případě, že si potřebujeme pohrát s vizualizačními parametry, třeba proto, že se snímek zdá být moc tmavý, tak najedeme na záložku *Layers*, přejedeme na vrstvu, kterou chceme upravit a vybereme ozubené kolečko. Můžeme upravit rozsah vizualizačních hodnot či průhlednost.



Záložka *Layers* nám také umožňuje spravovat viditelnost vrstev. Díky vypnutí vrstvy s kompozicí v pravých barvách se nám zobrazí kompozice v nepravých barvách.



6. NDVI index

V rámci práce s Image si vyzkoušíme ještě výpočet indexu NDVI. Podíváme se jak na výpočet pomocí mapové algebry, tak i na výpočet pomocí integrované funkce. Pro veškeré indexy není v prostředí Google Earth Engine integrovaná funkce, a tak není na škodu vyzkoušet si výpočet i pomocí mapové algebry.

V případě využití mapové algebry si musíme pásma snímku potřebná pro výpočet NDVI uložit do oddělených proměnných – metoda `select()`. Definujeme si proměnnou **nir**, která představuje blízké infračervené pásmo, a proměnnou **cervene**, reprezentující pásmo s červenou částí viditelného spektra.

Výsledek výpočtu NDVI uložíme do vlastní proměnné **algebraNDVI**. Pro výpočet využíváme funkce **subtract()** neboli odčítání, **divide()** tedy dělení a **add()** jako sčítání.

```
1. // Výpočet NDVI pomocí mapové algebry
2. var nir = snimekSentinel.select("B8");
3. var cervene = snimekSentinel.select("B4");
4. var algebraNDVI = nir.subtract(cervene).divide(nir.add(cervene));
```

Výpočet s využitím integrované funkce je jednodušší. Na náš ořezaný snímek aplikujeme funkci **normalizedDifference()** a jako parametry ji zadáme název blízkého infračerveného pásma a červeného pásma.

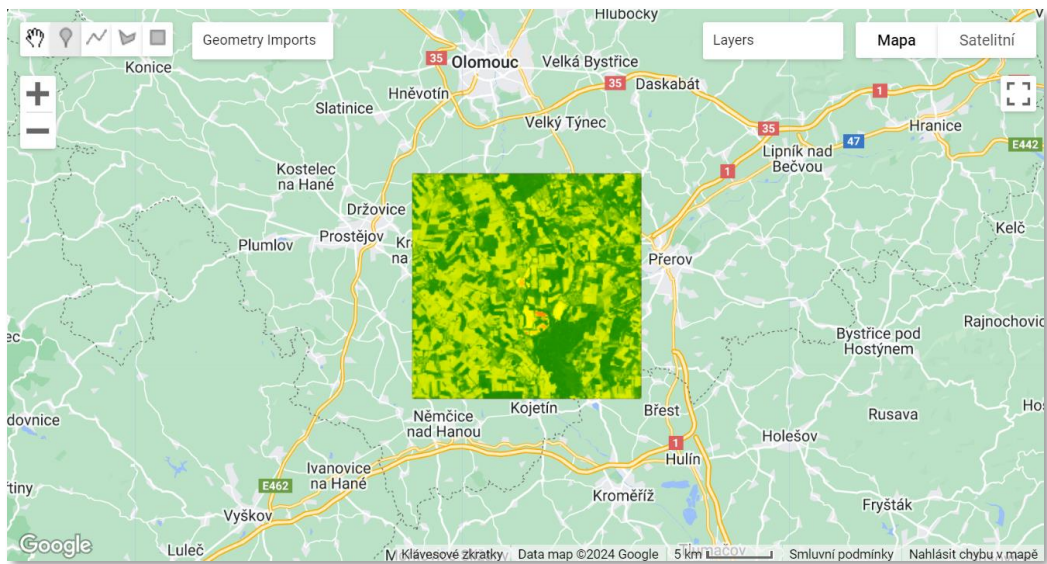
```
1. // Výpočet NDVI na základě vestavěné funkce
2. var NDVI = snimekSentinel.normalizedDifference(["B8", "B4"]);
```


7. Vizualizace NDVI indexu

Vypočtené NDVI indexy si opět zobrazíme v mapě pomocí **Map.addLayer()**. Jako vizualizační parametry zadáme možný rozsah hodnot NDVI indexu (-1,1) a barevnou paletu. Pro definici barev lze využít přímo anglické názvy barev či hexadecimální kódy.

```
1. // Vizualizace NDVI snímků
2. Map.addLayer(algebraNDVI, {min:-1, max: 1, palette: ["red", "yellow", "green"]}, "NDVI -
mapová algebra");
3. Map.addLayer(NDVI, {min:-1, max: 1, palette: ["red", "yellow", "green"]}, "NDVI - funkce");
```

V mapovém okně se nám zobrazí NDVI vypočítané jak podle mapové algebry, tak pomocí integrované funkce. Pomocí záložky *Layers* se mezi nimi můžeme přepínat a zjistíme, že jsou výsledky totožné.



8. Export snímku

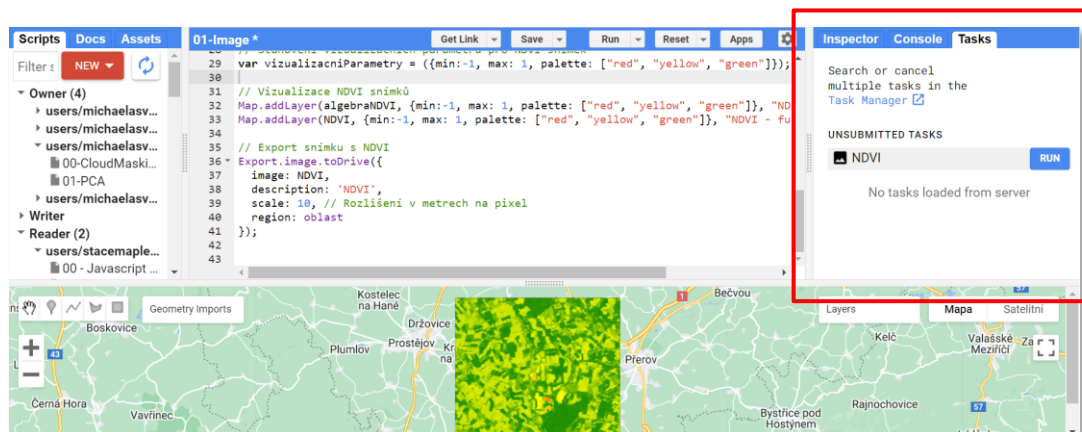
Výsledek naší práce si můžeme kdykoliv vyexportovat ve formátu GeoTIFF a otevřít v jiném programu.

Pro vyzkoušení exportu využijeme vypočítané NDVI a formuli **Export.image.toDrive()**, která říká, že si chceme vyexportovat snímek na Google Disk. Uvnitř **Export.image.toDrive()** specifikujeme, který snímek chceme exportovat, popis snímku, rozlišení našeho snímku a oblast, kterou pokrývá. Poté už jen spustíme skript.

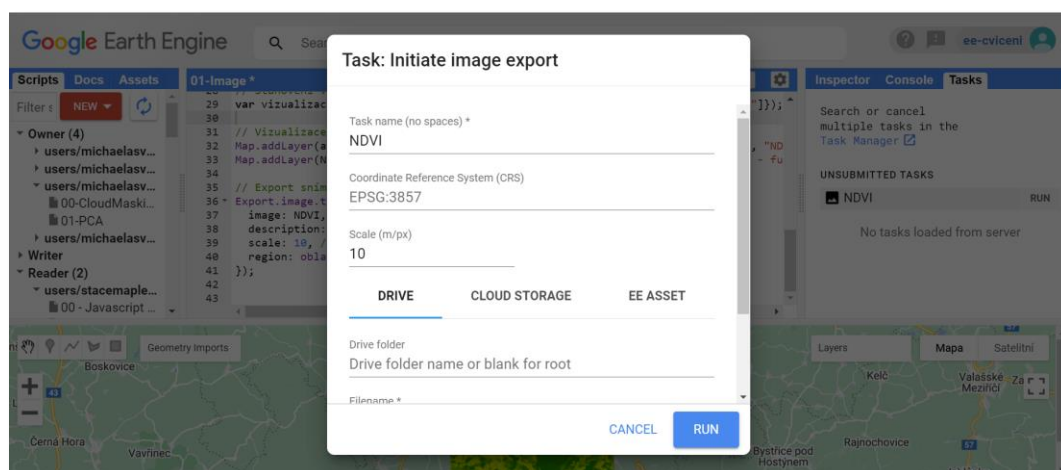
```
1. // Export snímku s NDVI
2. Export.image.toDrive({
3.   image: NDVI,
4.   description: 'NDVI',
5.   scale: 10, // Rozlišení v metrech na pixel
6.   region: oblast
```

7. });

Přejdeme do záložky *Tasks* v pravé části Google Earth Engine a potvrdíme export našeho snímku pomocí tlačítka *RUN*.



Otevřou se nám informace o exportovaném snímku a v případě zájmu lze upravit detaily exportu. Pokud vše odpovídá, tak export potvrdíme tlačítkem *RUN* a bude zahájen.



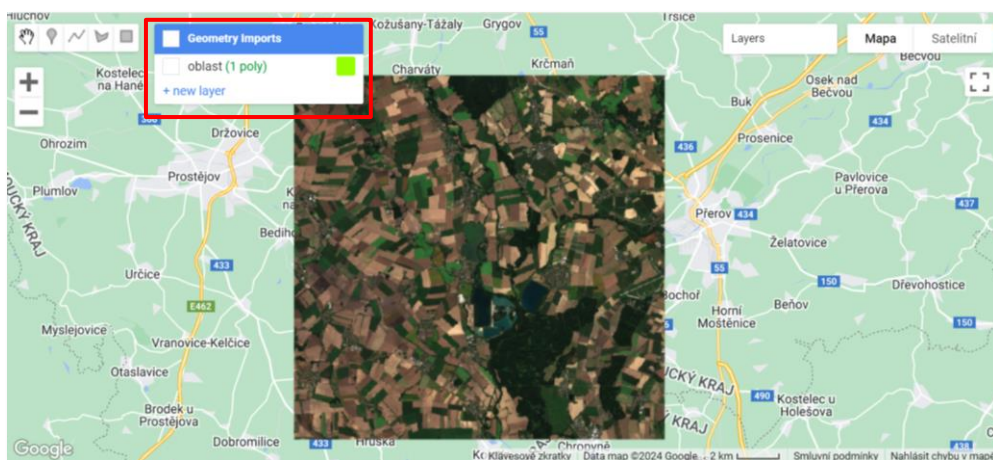
Po dokončení si můžeme náš snímek otevřít na Google Disku a stáhnout k sobě do počítače.

9. Výběr bodů pro spektrální křivku

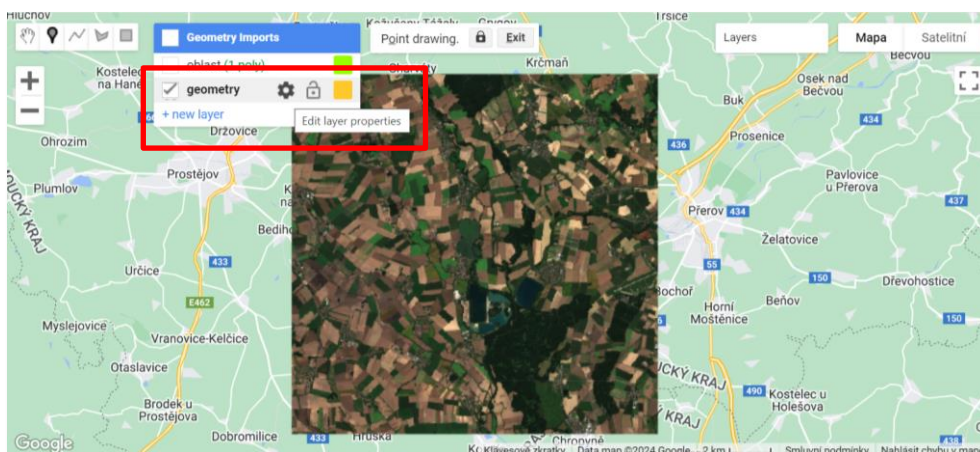
Na závěr práce s Image si ještě ukážeme, jak vytvořit spektrální křivku.

Jako první je nutné vybrat body, které budou reprezentovat jednotlivé typy povrchu. Postupně vyznačíme body pro vodstvo, vegetaci, pole a zástavbu.

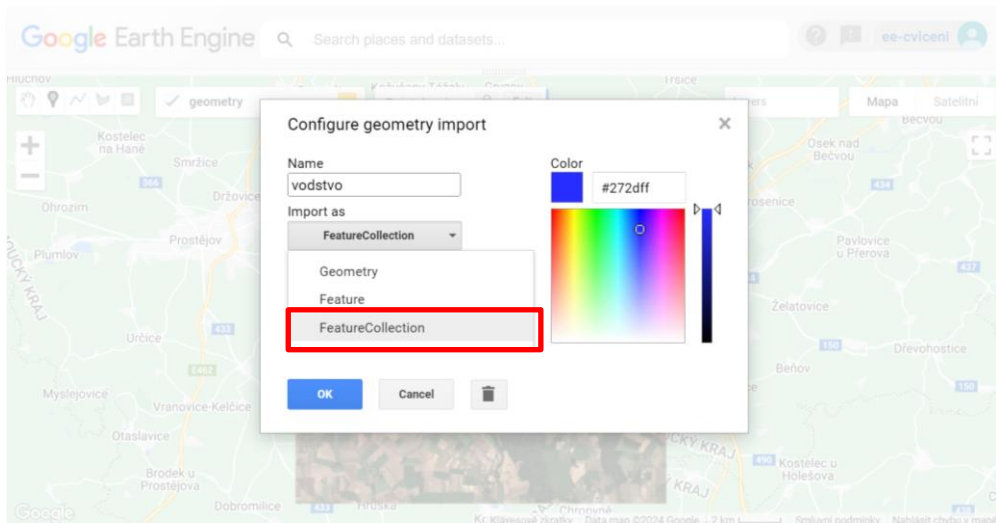
V mapovém okně přejdeme na záložku *Geometry Imports* a přidáme novou vrstvu pomocí *+ new layer*.




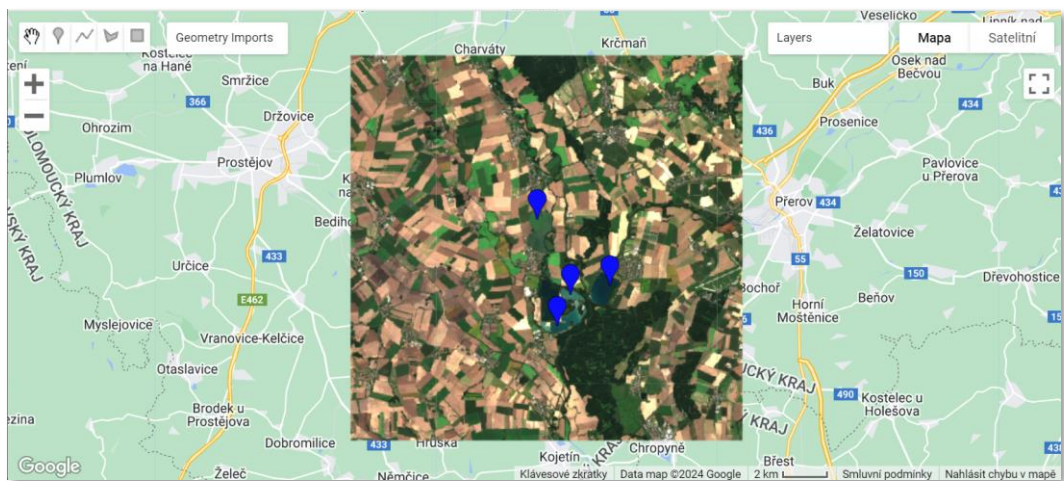
Vznikla nová vrstva s názvem **geometry**, do které budeme následně přidávat body reprezentující vodstvo. Před přidáním bodů je ale třeba upravit nastavení nové vrstvy, a proto přes ozubené kolečko otevřeme vlastnosti vrstvy.



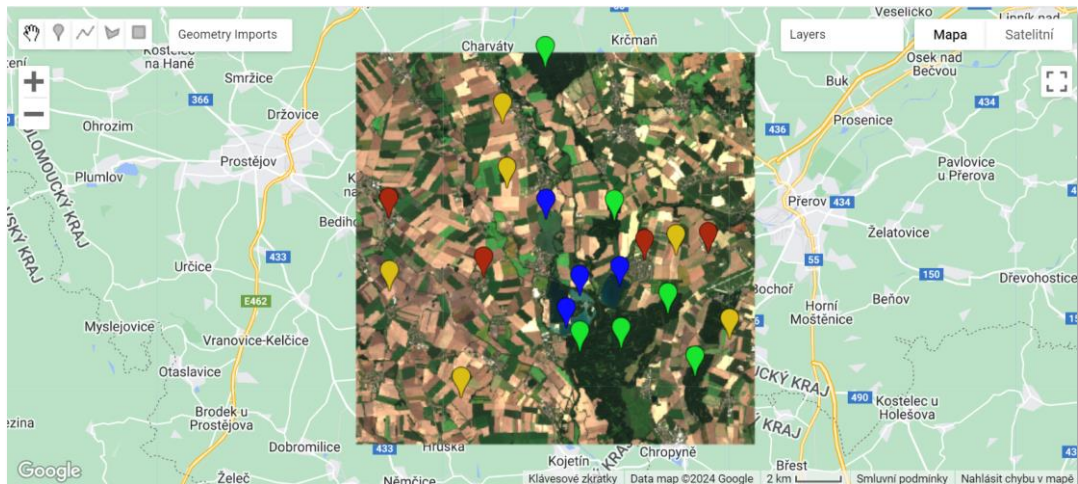
Změníme název vrstvy na vodstvo a v rozbalovacím menu zvolíme možnost *FeatureCollection*, protože naše vrstva bude kolekce prvků. Dále můžeme změnit barvu výplně pinů v mapě, tak aby korespondovala s typem povrchu. V případě vodstva tedy můžeme vybrat modrou barvu. Pro potvrzení našich změn už jen klikneme na tlačítko *OK*.



Po úspěšném nastavení nové vrstvy aktivujeme nástroj na přidávání pinů  v levém horním rohu mapového okna a postupně vyznačíme několik pinů reprezentujících vodstvo. Snažíme se piny rozmístit po celém snímku, a ne je koncentrovat pouze v jednom místě.



Stejným postupem přidáme piny reprezentující vegetaci, pole a zástavbu.



Ze všech vyznačených pinů vytvoříme souhrnnou kolekci pomocí `ee.FeatureCollection()`.

```
1. // Tvorba kolekce prvků, které reprezentují různé druhy povrchu
2. var body = ee.FeatureCollection([vegetace, pole, vodstvo, zastavba]);
```

10. Přejmenování pásem

Pro korektní tvorbu spektrální křivky je nutné přejmenovat pásma snímku. Google Earth Engine jinak chybně zařazuje pásma B11 a B12 na počátek osy X.

Jako první vytvoříme seznam s pásmy, která vstupují do tvorby spektrální křivky. Jedná se o pásma B1, B2, B3, B4, B5, B6, B7, B8, B8A, B9, B11, B12.

Následně vytvoříme seznam, do kterého vložíme nová označení pásem. Nová označení jsou 1, 2, 3, 4, 5, 6, 7, 8, 8A, 9, 11, 12. Před téměř všechna označení kromě názvů pro pásma B11 a B12 přidáváme mezeru. Díky tomu zabráníme chybnému znázornění pásem B11 a B12 ve spektrální křivce.

```
1. // Seznam pásem, které budou vstupovat do tvorby spektrální křivky
2. var pasma = ["B1", "B2", "B3", "B4", "B5", "B6", "B7", "B8", "B8A", "B9", "B11", "B12"];
3.
4. // Seznam nových označení pásem
5. var pasmaNove = [" 1", " 2", " 3", " 4", " 5", " 6", " 7", " 8", " 8A", " 9", "11", "12"];
```

Dále už jen vytvoříme nový snímek s přejmenovanými pásmy. Pro jeho tvorbu využíváme náš ořezaný snímek, ze kterého vybereme požadovaná pásma s využitím `select()` a tyto pásma přejmenujeme pomocí `rename()`.

```
1. // Snímek s požadovanými pásmy a jejich přejmenování
2. var pasmaSnímek = snímekSentinel1.select(pasma).rename(pasmaNove);
```

11. Vykreslení spektrální křivky

Před definicí spektrální křivky si definujeme její vizualizační parametry. Zvolíme si titulek spektrální křivky, názvy os, tloušťku linie v grafu, velikost bodů v grafu, a nakonec barevné zobrazení.

```
1. // Definování parametrů pro spektrální křivku
2. var zobrazeniParametry = {
3.   title: 'Sentinel-2 spektrální křivka',
4.   hAxis: {title: 'Vlnové délky [nm]'},
5.   vAxis: {title: 'Odráživost'},
6.   lineWidth: 1,
7.   pointSize: 3,
8.   series: {
9.     0: {color: 'green'},
10.    1: {color: 'red'},
11.    2: {color: 'blue'},
12.    3: {color: 'black'},
13.  };
```

Vytvoříme seznam, který představuje hodnoty vlnových délek jednotlivých pásem. Pokud bychom tento krok vynechali, byly by na ose X vypsány názvy jednotlivých pásem.

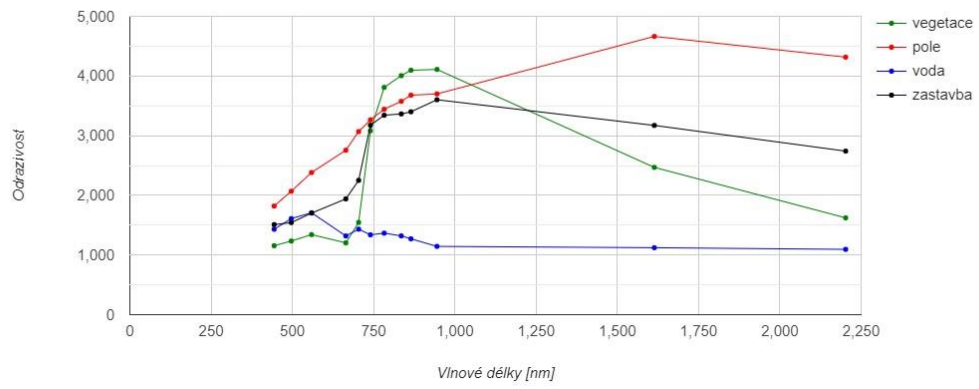
```
1. // Stanovení hodnot vlnových délek pro jednotlivá pásma pro vyšší přehlednost grafu
2. var vlnoveDelky =[443.9, 496.6, 559, 664.5, 703.9, 740.2, 782.5, 835.1, 864.8, 945, 1613.7, 2202.4];
```

Předposledním krokem tvorby spektrální křivky je její definice, kdy využíváme funkce **ui.Chart.image.regions()**. Mezi parametry této funkce patří snímek s přejmenovanými pásmy, kolekce bodů reprezentující jednotlivé typy povrchu, způsob agregace hodnot pinů pro jednotlivé typy povrchu, které se zobrazí na ose Y, rozlišení a seznam hodnot pro osu X. **SetChartType()** nám určí typ grafu a **setOptions()** nastaví naše definované vizualizační parametry.

Samotnou spektrální křivku už jen vykreslíme pomocí **print()** do konzole a následně si ji můžeme stáhnout ve formátu CSV, PNG nebo SVG.

```
1. // Vykreslení spektrální křivky
2. var spektralniKrivka= ui.Chart.image.regions(
3.   pasmaSnimek, body, ee.Reducer.mean(), 30, "nazev", vlnoveDelky)
4.   .setChartType('ScatterChart')
5.   .setOptions(zobrazeniParametry);
6. print(spektralniKrivka);
```

Sentinel-2 spektrální křivka



3.4 Práce s ImageCollection

Poslední sekce tohoto cvičení se týká práce s ImageCollection. ImageCollection je kolekce několika snímků, které pokrývají zájmové území v delším časovém úseku. Lze tak efektivně detekovat změny v území.

1. Založení nového skriptu

Začneme vytvořením nového skriptu. Postupujeme stejným způsobem jako při tvorbě skriptu pro práci s Image.

2. Import potřebných dat

Dále si vyznačíme zájmovou oblast a nahrajeme stejnou datovou sadu jako při práci s Image – COPENICUS/S2_SR_HARMONIZED.

3. Příprava dat

Naši kolekci je třeba opět vyfiltrovat, aby obsahovala pouze snímky pokrývající zájmovou oblast ve vybraném časovém období stejně tak jako při práci Image. Pro filtraci používáme také metodu **filterMetadata()**, která nám umožňuje vyfiltrovat kolekci snímků na základě hodnot atributů. Naším cílem je vybrat snímky s oblačností nižší než 20 %, a proto jako parametr funkce vkládáme název atributu s informací týkající se oblačnosti **CLOUDY_PIXEL_PERCENTAGE**, dále výraz **less_than**, který říká, že budou vybrány hodnoty menší než zadaná hraniční hodnota, a nakonec hraniční hodnota.

```
1. // Filtrace snímků pro zvolené období, oblast a filtrace podle oblačnosti
2. var sentinelSnímky = sentinel
3.     .filterBounds(oblast)
4.     .filterDate("2022-01-01", "2022-12-31")
5.     .filterMetadata("CLOUDY_PIXEL_PERCENTAGE", "less_than", 20)
6.     ;
```

Obsah kolekce je třeba ořezat na rozsah zájmové oblasti. Tentokrát nestačí pouze použít metodu **clip()** na celou kolekci, ale je nutné definovat funkci, který provede ořezání jednotlivých snímků v kolekci. Následující funkce provede ořez každého snímku v kolekci na rozsah zájmové oblasti. Funkce předpokládá, že zakreslené zájmové území má název **oblast**.

```
1. // Funkce na ořez kolekce snímků na rozsah zájmové oblasti
2. var orezKolekce = function(snimek){
3.     return snimek.clip(oblast);
4. };
```

Aby byla kolekce snímků ořezána využijeme metodu **map()**, která mapuje naši funkci na každý snímek obsažený v kolekci.

```
1. // Ořez snímků na rozsah zájmové oblasti
2. var sentinelOrez = sentinelSnimky.map(orezKolekce);
```

4. Vizualizace kolekce

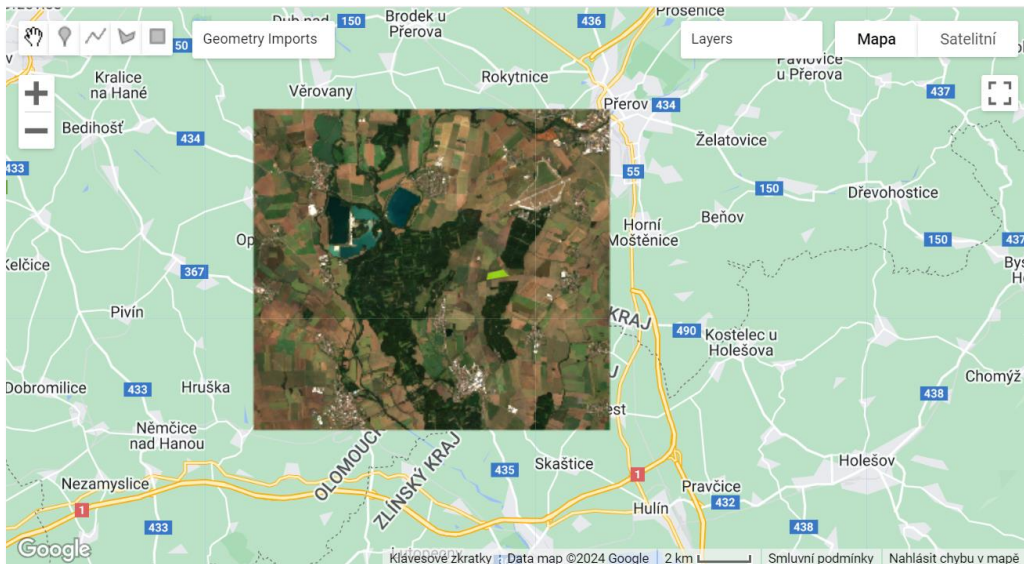
Kolekce snímků může obsahovat několik desítek či stovek snímků a je tedy nemožné zobrazit každý snímek v kolekci. Kolekci musíme agregovat s využitím minimálních, maximálních, průměrných či mediánových hodnot pixelů.

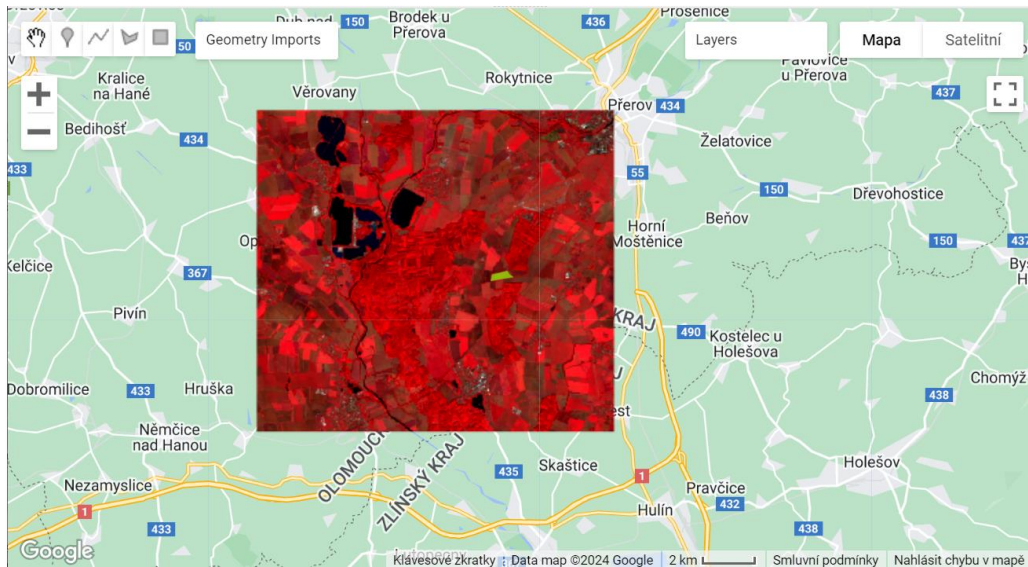
Snímky v naší kolekci agregujeme pomocí mediánových hodnot jednotlivých pixelů s využitím metody **median()**. Vznikne nám mediánový snímek.

```
1. // Výpočet mediánu pro každý pixel, pro každé pásmo nad kolekcí snímků -> vznik snímku, jehož
   // pixely nesou hodnoty mediánu pro dané pásma
2. var sentinelMedian = sentinelOrez.median();
```

Po vytvoření mediánového snímku nám už nic nebrání ve vizualizaci naší kolekce, i když jen v podobě souhrnného snímku. Do mapového okna přidáme vrstvy s kompozicemi v pravých a nepravých barvách.

```
1. // Vizualizace mediánové snímku v pravých a nepravých barvách
2. Map.addLayer(sentinelMedian, {bands: ["B4", "B3", "B2"], min:100, max: 2000}, "Sentinel-2
   // median RGB");
3. Map.addLayer(sentinelMedian, {bands: ["B8", "B4", "B3"], min:500, max: 4000}, "Sentinel-2
   // median FCC");
```





5. NDVI

Postup výpočtu NDVI pro kolekci snímků je podobný postupu u Image. U ImageCollection je nutné definovat funkci, která provede výpočet NDVI na každém snímku.

Definujeme si funkci, která pro každý snímek provede výpočet NDVI a zároveň přidá každému snímku nové pásmo obsahující NDVI. K výpočtu NDVI v rámci funkce na přidání NDVI pásma ke snímku využijeme integrovanou funkci **normalizedDifference()** a pásma B8 a B4.

```
1. // Funkce na přidání vypočteného NDVI jako dalšího pásma ke snímkům
2. var pridatNDVI = function(snimek) {
3.   var ndvi = snimek.normalizedDifference(["B8", "B4"]).rename("NDVI");
4.   return snimek.addBands(ndvi);
5. };
```

Následně naši funkci použijeme na ořezanou kolekci.

```
1. // Použití funkce na přidání NDVI pásma nad naší kolekcí snímků
2. var snimkyNDVI = sentinelOrez.map(pridatNDVI);
```

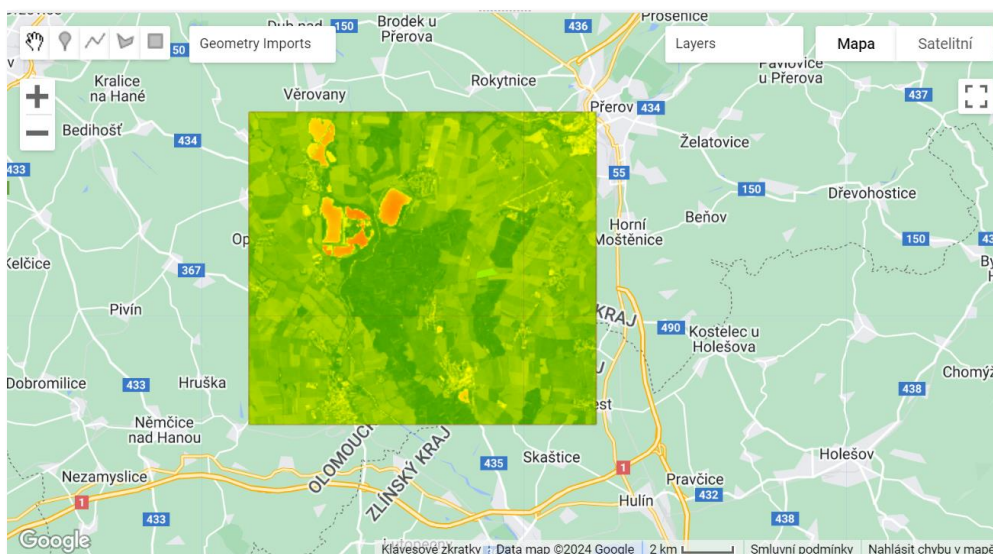
Pokud si chceme ověřit, že ke snímkům v naší kolekci bylo přidáno pásmo s hodnotami NDVI, využijeme metodu **getInfo()**, která v konzoli vypíše informace o naší kolekci, a tedy i informace o jednotlivých snímcích a jejich pásmech.

```
1. print(snimkyNDVI.getInfo());
```


6. Vizualizace NDVI

Pro vizualizaci NDVI je nutné opět agregovat hodnoty. Agregaci provedeme v rámci přidání nové vrstvy do mapového okna pomocí **Map.addLayer()**. Tentokrát si však pro agregaci vyzkoušíme metodu **mean()**.

```
1. // Vizualizace vypočteného průměrného NDVI
2. Map.addLayer(snimkyNDVI.mean(), {bands: ["NDVI"], min:-1, max: 1, palette: ["red", "yellow", "green"]}, "NDVI - průměr");
```



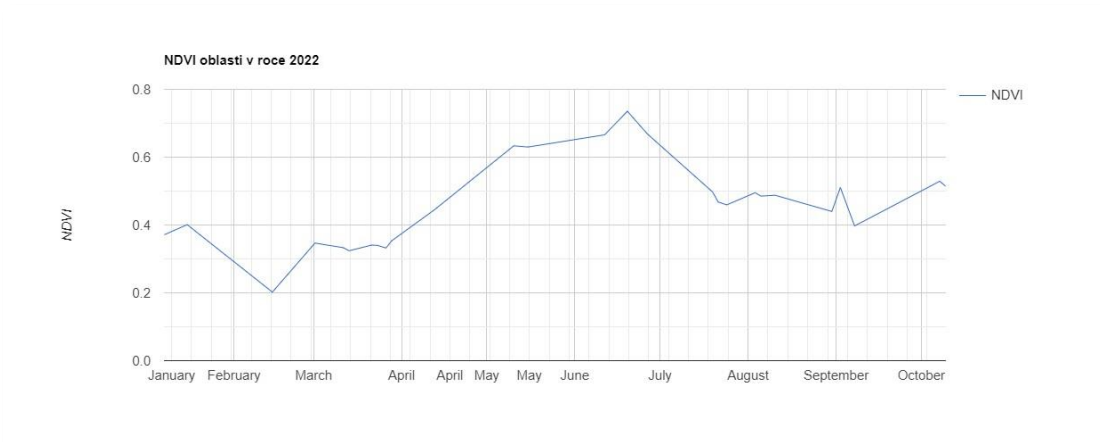
7. Vývoj hodnot NDVI v celé zájmové oblasti

Na závěr celého cvičení si vytvoříme grafy reprezentující vývoj hodnot NDVI v čase.

Pro tvorbu grafu reprezentujícího vývoj NDVI hodnot pro celé zájmové území využíváme funkce **ui.Chart.image.series()**. Vybereme u každého snímku z kolekce pouze pásmo obsahující vypočítaný NDVI indexem pomocí **select()**. Stanovíme oblast, pro kterou je graf tvořen a určíme způsob agregace hodnot – **ee.Reducer.mean()**. Náš graf tedy bude reprezentovat vývoj průměrných hodnot NDVI v čase. Pak už jen stanovíme vizualizační parametry našeho grafu a samotný graf nakonec vykreslíme do konzole pomocí **print()**.

```
1. // Tvorba grafu, který zobrazuje průměrnou hodnotu NDVI v čase pro celé území
2. var grafOblast = ui.Chart.image.series({
3.   imageCollection: snimkyNDVI.select("NDVI"), // Výběr NDVI pásem
4.   region: oblast, // Výběr oblasti
5.   reducer: ee.Reducer.mean(), // Způsob agregace hodnot
6.   scale: 20
7. }).setOptions({
8.   lineWidth: 1,
9.   title: "NDVI oblasti v roce 2022",
10.  interpolateNulls: true,
11.  vAxis: {title: "NDVI"},
12.  hAxis: {title: "", format: "MMMM"}
```

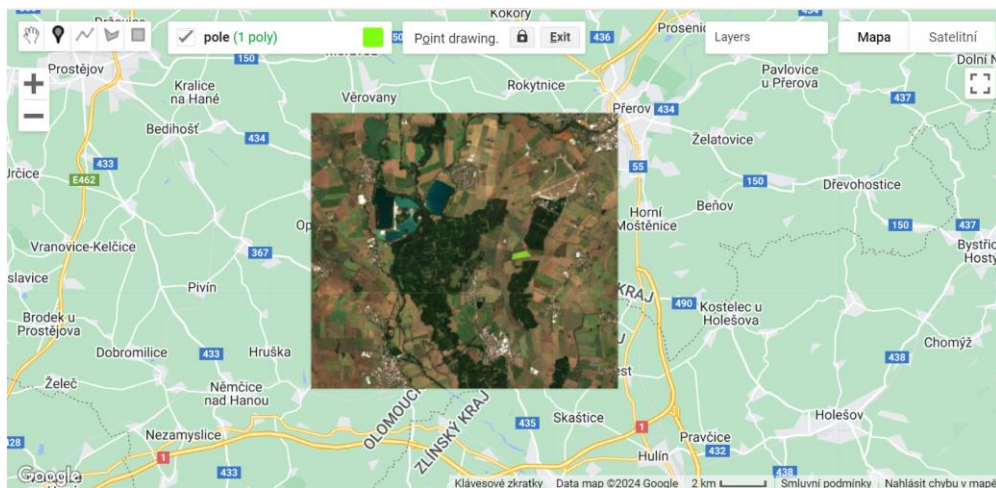
```
13. });  
14. print(grafOblast);
```



8. Vývoj hodnot NDVI pro konkrétní pole

V prostředí Google Earth Engine je možné vykreslit graf vývoje NDVI hodnot i pro nějaké konkrétní pole či například les. Můžeme tak efektivně pozorovat stav vegetace v průběhu celého časového období.

Před tvorbou grafu pro konkrétní pole je nutné si jej vyznačit v mapovém okně. Ponecháme si aktivní vrstvu se snímkem v pravých barvách a přidáme novou vrstvu v *Geometry Imports*, která bude obsahovat naše zakreslené pole. Následně už jen pole zakreslíme.



Při tvorbě grafu postupujeme stejně jako v případě grafu pro celé zájmové území. Jediným rozdílem je název grafu a území, pro které se má graf vytvořit. Místo celé oblasti uvedeme do části region pouze vrstvu pole. Nakonec graf vykreslíme do konzole.

```
1. // Tvorba grafu, který zobrazuje průměrnou hodnotu NDVI v čase pro vybrané pole
2. var grafPole = ui.Chart.image.series({
3.   imageCollection: snimkyNDVI.select("NDVI"),
4.   region: pole,
5.   reducer: ee.Reducer.mean(),
6.   scale: 20
7. }).setOptions({
8.   lineWidth: 1,
9.   title: "NDVI pole v roce 2022",
10.  interpolateNulls: true,
11.  vAxis: {title: "NDVI"},
12.  hAxis: {title: "", format: "MMMM"}
13. });
14. print(grafPole);
```

