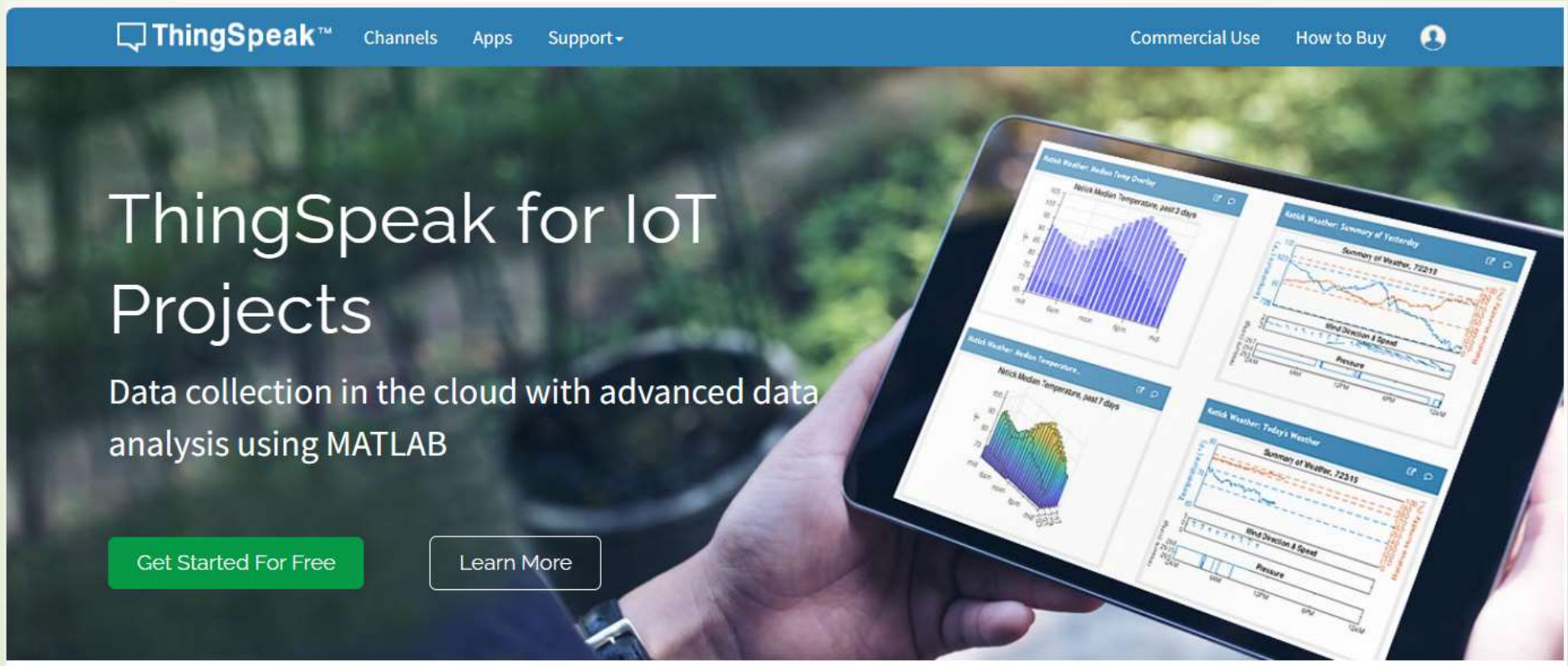
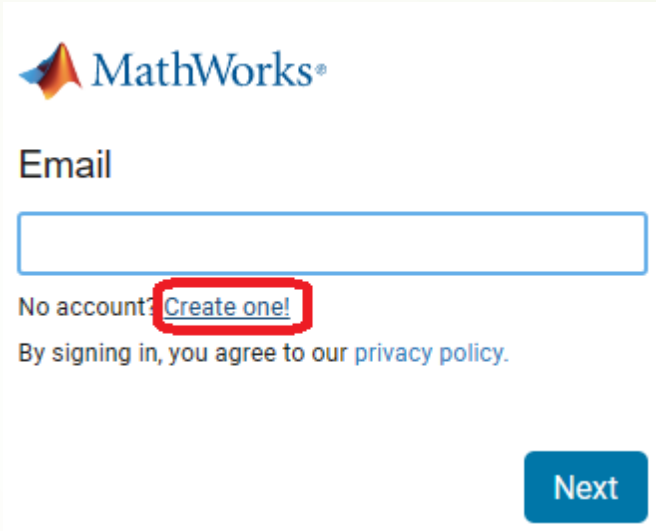
Thingspeak IoT platforma

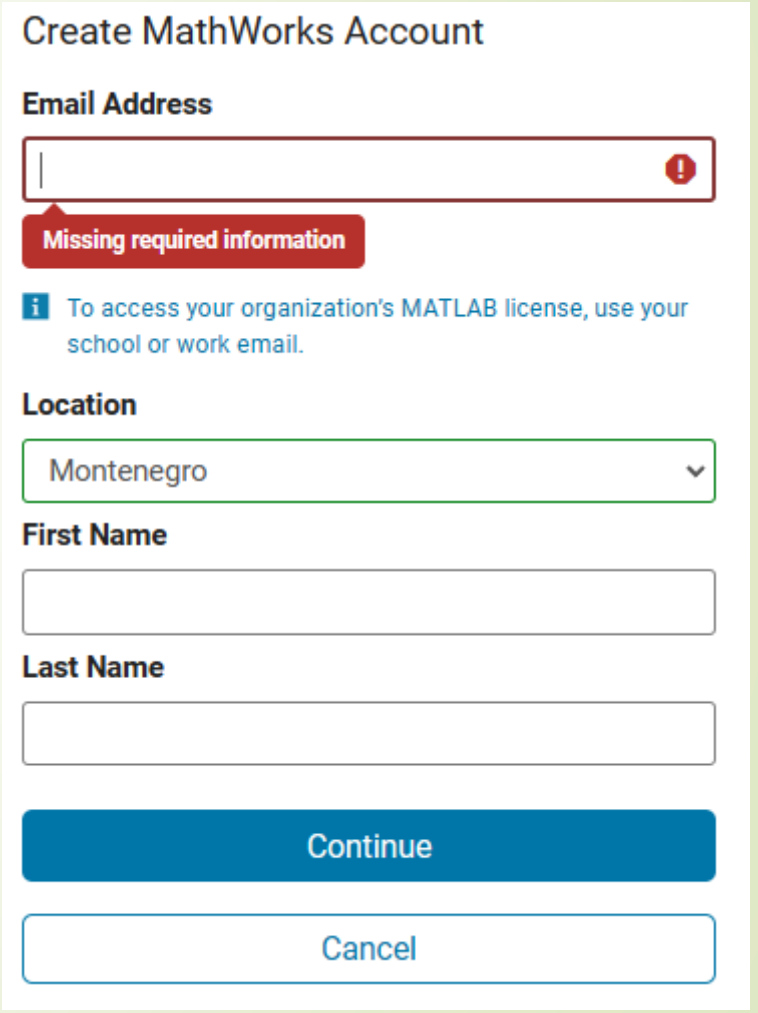
ThingSpeak

- IoT analitička platforma koja omogućuje prikupljanje, vizuelizaciju i analizu padataka uživo.
- ThingSpeak omogućuje inženjerima i naučnicima da naprave prototip i IoT sistem bez postavljanja servera i razvijanja web softvera.
- Otvorena IoT platforma sa MATLAB analitikom
- Nalazi se na adresi: [IoT Analytics - ThingSpeak Internet of Things]



ThingSpeak prijavljivanje





Kreiraj račun na ThingSpeak https://thingspeak.com/

HTTP – za komuniciranje sa ThingSpeak

- HTTP – Hypertext Transfer protokol
- Dizajniran da omogući komunikacju između servera i klijenta
- Protokol zahtjeva i odgovora
- Klijent šalje HTTP zahjev serveru – server klijentu uzvraća odgovor
- Odgovor sadrži status izvršenja zahtjeva, a može sadržati i dodatne podatke.

- U radu sa ThingSpeak platformom Arduino uređaj će imati ulogu klijenta a ThingSpeak platforma ulogu servera.

HTTP zahtjev

HTTP zahtjev generiše klijent, prema imanovanom host-u, lociranom na serveru.

Cilj zahtjeva je pristup resursu na serveru.

Korektno sastavljen HTTP zahtjev sadrži sljedeće elemente:

- Liniju zahtjeva;
- HTTP zaglavlja;
- Tijelo poruke, ako je potrebno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon poslednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP zahtjev – Linija zahtjeva

Linija zaglavlja je prva linija u poruci zahtjeva. Sastoji se iz tri dijela:

- Metod. Metod je jedno-rječna komanda koja govori serveru što da radi sa resursom. Na primjer, server može biti upitan da pošelje resurs klijentu.
- Komponenta staze URL-a za zahtjev. Staza identifikuje resurs na serveru.
- Broj HTTP verzije, ukazuje na HTTP specifikaciju s kojom je klijent pokušao uskladiti poruku.

Primjer linije zahtjeva:

GET /software/htp/cic/indeks.html HTTP/1.1

Linja zahtjeva može sadržati i dodatne podatke.

HTTP zahtjev – Zaglavlje (Header)

- Pruža prijemnoj strani informacije o poruci, pošiljaocu i načinu na koji pošiljaoc želi da komunicira sa primaocem.
- Svako HTTP zaglavlje se sastoji od imena i vrijednosti.
- HTTP protokol definiše standarsni set HTTP zaglavlja i opisuje kako ih koristiti korektno.
- HTTP zaglavlje zahtjeva klijenta sadrži informacije koje server može upotrijebiti u odlučivanju kako da odgovori na zahjev. To može biti da klijent čita zahtijevani dokunet na francuskom ili njemačkom jeziku i da dokument treba biti poslat jedino ako je mijenjan od naznačenog datuma.

```
Accept-Language: fr, de
If-Modified-Since: Fri, 10 Dec 2004 11:22:13 GMT
```

HTTP zahtjev – Tijelo poruke

- Može se nazvati i tijelom zahtjeva
- Aktuelni sadržaj poruke.
- Tijelo poruke može biti u originalnom obliku ili može biti kodirano.
- Može se nazvati i tijelom zahtjeva
- Prikladno je za neke metode zahtjeva, dok za druge nije.
- Na primjer, zahtjev sa POST metodom, koji šalje ulazne podatke serveru, ima tijelo poruke, koje sadrži te podatke.
- Zahtjev sa GET metodom, koji od servera traži da pošalje resurs, ne sadrži tijelo poruke.

HTTP odgovor

- HTTP odgovor generiše server i šalje klijentu.
- Cilj odgovora je da obezbijedi klijentu treženi resurs ili da ga informiše o izvršenju zahtejva ili da dojavi da je došlo do greške.
- HTTP odgovor se sastoji iz:
  - Statusne linije;
  - Zaglavlja;
  - Tijela poruke, koje je obično neophodno.

Nakon svakog HTTP zaglavlja slijedi znak za povratak na početak reda (carriage return) i znak za prelazak u novi red (line feed) (CR-LF). Nakon poslednjeg zaglavlja dodatni CR-LF je dodat (za dobijanje prazne linije), nakon kojeg počinje tijelo poruke.

HTTP odgovor - Statusna linija

- Statusna linija je prva linija u odgovoru. Sasloji se iz tri segmenta:
  - Broj HTTP verzije, koji ukazuje na HTTP specifikaciju po kojoj je server pokušao da usladi odgovor.
  - Statusni kod, koji je trocifarski broj i ukazuje na rezultat izvršenja zahtjeva.
  - Fraza razloga, poznata i kao tekst statusa, koji je čitljiv čovjeku i sažima značenje statusnog koda.

Primjer statusne linje:

```
HTTP/1.1 200 OK
```

HTTP odgovor – Zaglavlja (Headers)

- Sadrži informacije koje klijent koristi da pronađe više podataka o odgovoru, kao i da pronađe podatke o serveru koji je poslao poruku.
- Ove informacije mogu pomoći klijentu u prezentaciji odgovora korisniku.
- Na primjer, prikazana zaglavlja govore klijentu kada je odgovor poslat, od strane kojeg servera je poslat, kao i da je to JPEG slika.

```
Date: Thu, 09 Dec 2004 12:07:48 GMT
Server: IBM_CICS_Transaction_Server/3.1.0(zOS)
Content-type: image/jpg
```

HTTP odgovor – Tijelo poruke

- Naziva se i tijelom odgovora.
- Većina odgovora sadrže tijelo poruke. Izuzeci su kada server odgovara na zahtjev klijenta, koji je koristio HEAD metod (koji koristi zaglavlja ali ne i tijelo odgovora) i gdje server koristi određene statusne kodove.
- U odgovoru na uspješno izvršen zahtjev, tijelo poruke sadrži resurs koji je klijent zahtijevao ili neke informacije o statusu radnje koji je klijet zahtijevao.
- U odgovoru na neuspješno izvršen zahtjev, tijelo poruke može da pruži dodatne informacije o razlozima greške ili o nekoj radnji koju klijent treba da preduzima da bi se zahtjev uspješno izvršio.

HTTP zahtjev - metode

- HTTP definiše set metoda (načina) da indicira akciju koja će biti izvršena na datom resursu.
- Mada mogu biti i imenice, metode zahtjeva se često označavaju kao HTTP glagoli.
- Svaki metod koristi različitu semantiku.

HTTP zahtjev – vrste metoda

- **GET** Get metod se koristi za traženje podataka iz određenog resursa.
- **HEAD** HEAD metod zahtijeva zaglavlja resursa identificiranog danim URL-om, bez vraćanja samog resursa.
- **POST** POST metod se koristi za slanje podataka na obradu određenom resursu, često izazivajući promjene stanja ili druge efekte na njemu.
- **PUT** PUT se koristi za ažuriranje ili zamjenu postojećeg resursa na serveru sa obezbijeđenim podacima.
- **DELETE** DELETE metod briše specificirani resurs.
- **CONNECT** Uspostavlja vezu sa serverom, onosno, identificiranim ciljanim resursom.
- **OPTIONS** OPTIONS metod opisuje komunikacione opcije za ciljani resurs.
- **TRACE** TRACE metod vrši praćenje komunikacionog linka do ciljanog resursa.
- **PATCH** PATCH metod obavlja parcijalnu modifikaciju resursa

HTTP zahtjev – **GET** metod

- GET se koristi za traženje podataka iz specificiranog izvora
- Treba imati na umu da se upitni string (par ime/vrijednost) šalje u URL-u GET zahtjeva.

/test/demo_form.php?name1=value1&name2=value2

- Nekoliko napomena u vezi GET zahtjeva:
  - GET zahtjevi se mogu keširati (spremiti u predmemoriju)
  - GET zahtjevi ostaju u istoriji pregledača
  - GET zahtjevi se mogu objeležiti
  - GET zahtjevi se nikada ne bi trebali koristiti kada se radi o osjetljivim podacima
  - GET zahtevi imaju ograničenje dužine
  - GET zahtevi se koriste samo za traženje podataka (ne promjenu)

HTTP zahtjev – **POST** metod

- POST metod se koristi za slanje podataka serveru za kreiranje/ažuriranje resursa
- Podaci poslati serveru POST metodom smješteni su u tijelu HTTP zahtjeva.

```
POST /test/demo_form.php HTTP/1.1
Host: w3schools.com

name1=value1&name2=value2
```

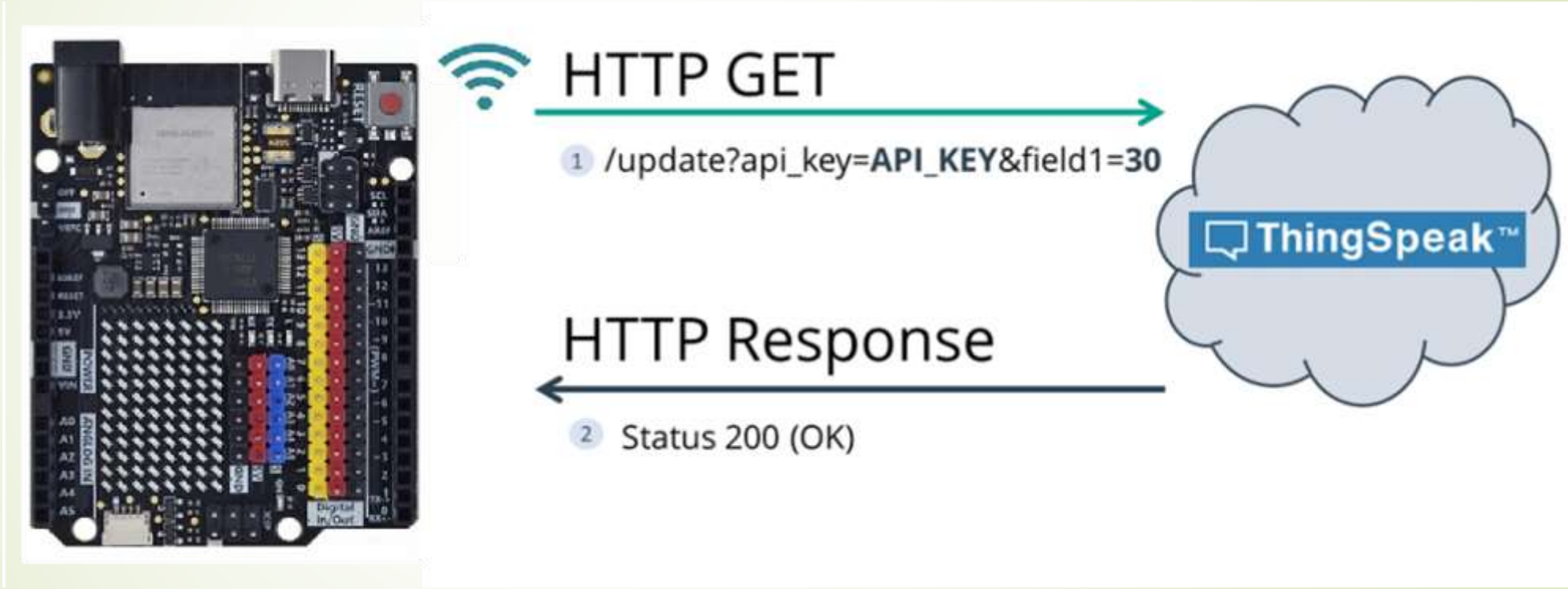
- Nekoliko napomena u vezi POST zahtjeva:
  - POST zahtevi se nikada ne kešuju
  - POST zahtevi ne ostaju u istoriji pregledača
  - POST zahtevi se ne mogu označiti
  - POST zahtevi nemaju ograničenja u pogledu dužine podataka

HTTP zahtjev – **GET** vs. **POST** metod

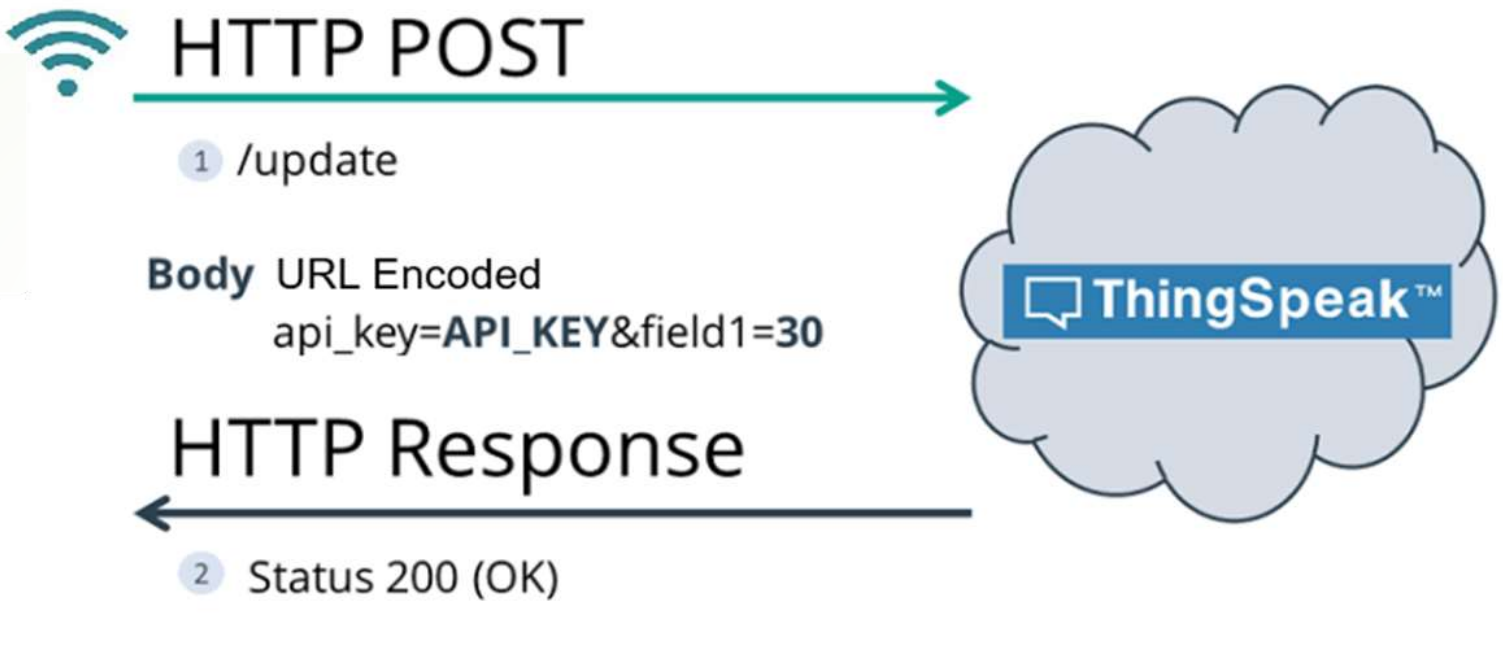
	GET	POST
BACK dugme/Ponovo učitaj	Neškodljivo	Podaci že biti ponovo poslati
Označenost	Može se označiti	Ne može se označiti
Keširanje	Može se keširati	Ne može se keširati
Tip kodiranja	application/x-www-form-urlencoded	application/x-www-form-urlencoded ili multipart/form-data. Upotreba višedjelnog kodiranja za binarne podake
Istorija	Parametri ostaju u istoriji pregledača	Parametri ne ostaju u istoriji pregledača
Ograničenja u dužini podataka	Da, kod slanja podataka, GET method dodaje podatke na URL; dužina URL-a je ograničena (maximalna URL dužina je 2048 karaktera)	Bez ograničenja
Ograničenja u tipu podataka	Samo ASCII karakteri dozvoljeni	Bez restrikcija. Binarni podaci su takođe dozvoljeni
Bezbjednost	GET manje siguran u poređenju s POST, jer su podaci dio URL-a Ne koristiti GET kada se šalje ložinka ili driuge osjetljive informacije	POST je malo sigurniji od GET jer parametri nijesu smješteni u istoriji pregledača ili u web server logu
Vidljivost	Podaci su vidljivi svima u URL-u	Podaci nijesu prikazani u URL-u

Arduino UNO+ESP8266 – GET preko URL-a

URL (Uniform Resource Identifier)je u stvari web adresa oblika http ili https.

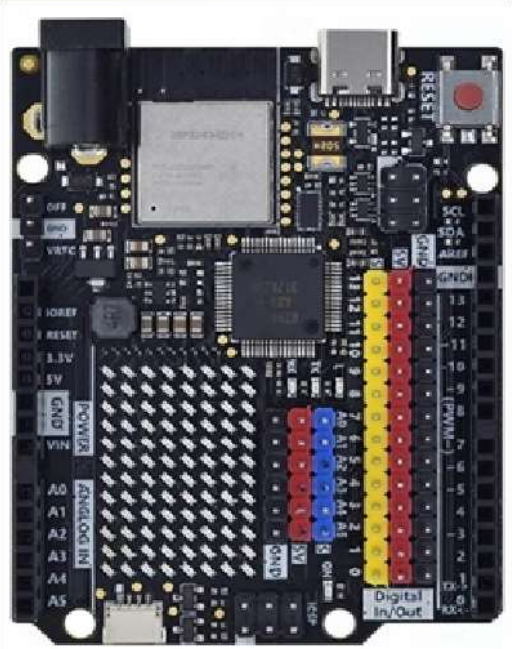


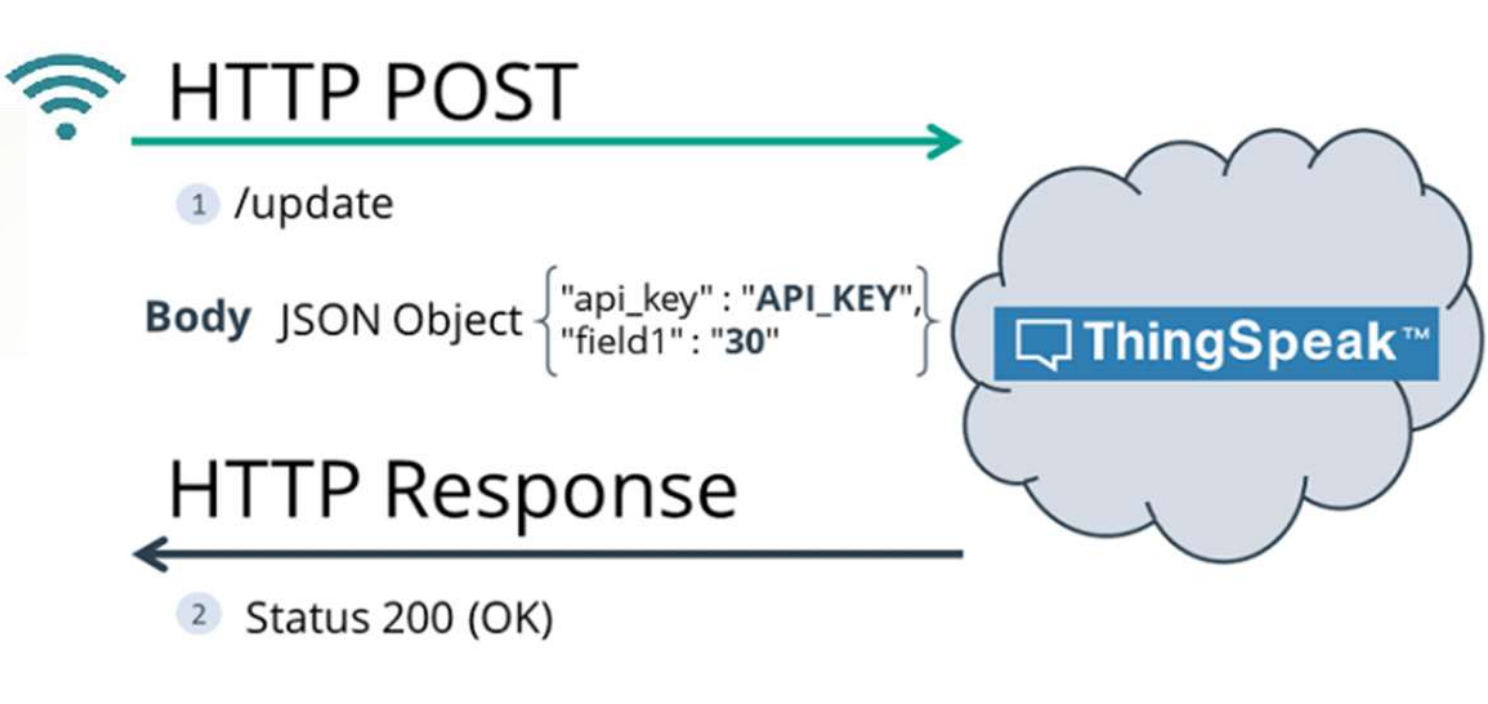
Arduino UNO+ESP8266 – HTTP POST - URL enkodiran



POST /update HTTP/1.1
Host: example.com
api_key=api&field1=value1
Content-Type: application/x-www-form-urlencoded

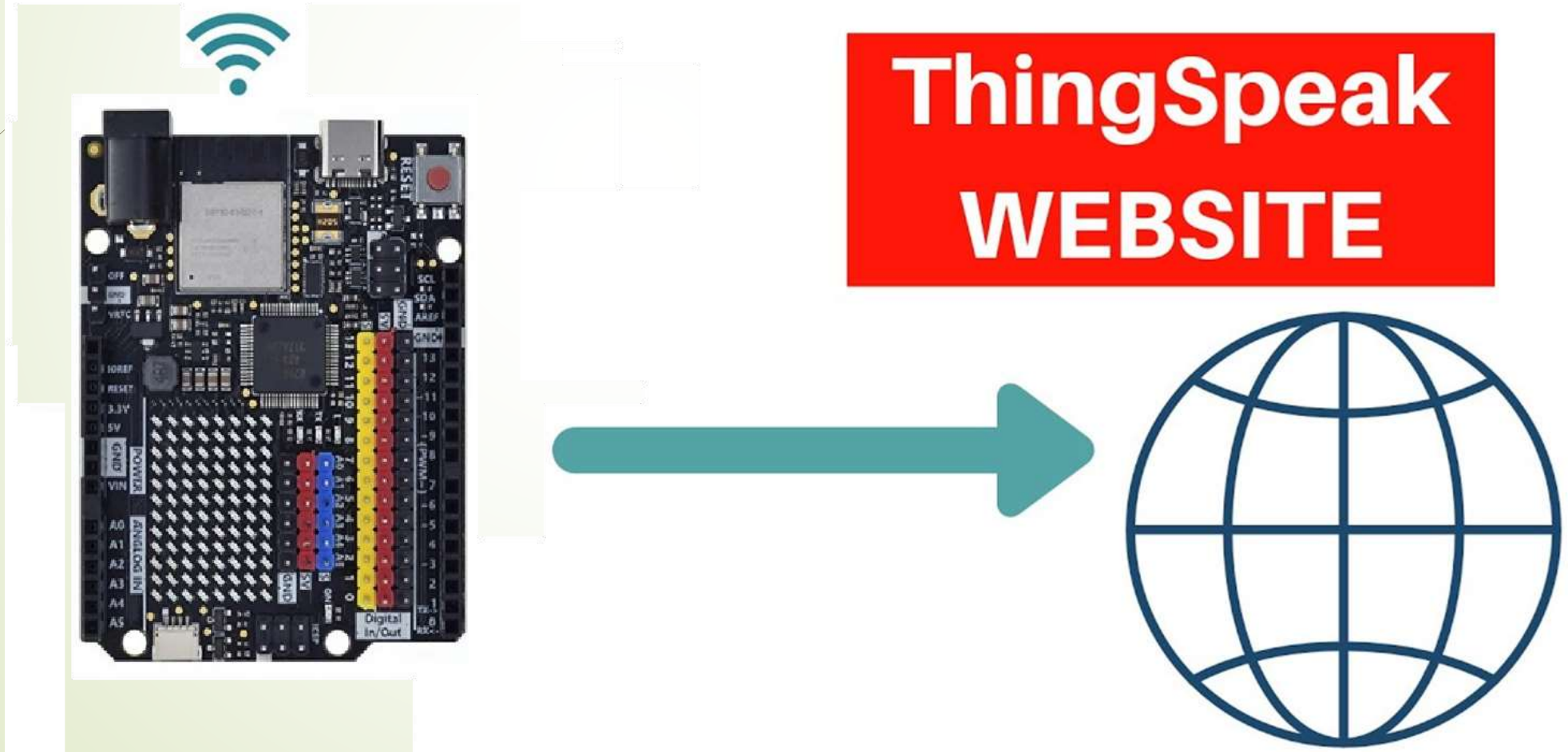
Arduino UNO+ESP8266 – HTTP POST - JSON object





POST /update HTTP/1.1
Host: example.com
{api_key: "api", field1: value1}
Content-Type: application/json

Upisivanje podataka u ThingSpeak kanal



Za vježbu

1. Upotrijebiti sensor za temperaturu i vlagu. Vrijednosti dobijene sa senzora slati na ThingsSpeak i prikazivati u dva odvojena dijagrama istog kanala.

(2-1)

2. Otvaranjam i zatvaranjem prekidača na uređaju 1, mijenjati smjer okretanja koračnog motora na uređaju 2. Pritisnut taster na uređaju 1, zaustavlja okretanje koračnog motora na uređaju 2. Otpušten taster na uređaju 1, omogućuje nastavak okretanja koračnog motora na uređaju 2, na isti način kako se okretao prije pritiskanja tastera na uređaju 1.

(4-3-2)