

# T-map: projektrapport

Twitter Trends map with english translation

Linn Borgström, Eric Simonsson, Bertilla Fabris, Thiptida Luengprakarn

<b>Om tjänsten</b>	<b>3</b>
Syfte	3
<b>Den tekniska lösningen</b>	<b>3</b>
Flask	3
Bootstrap	3
Ajax	3
Swagger	4
<b>Beskrivning av de datakällor/externa API:er</b>	<b>4</b>
GoogleMaps API	4
Twitter API	4
Google Cloud Translation API	5
<b>Hur data används</b>	<b>5</b>
<b>Installationsinstruktioner</b>	<b>5</b>
<b>Körinstruktioner</b>	<b>6</b>
Windows	6
Mac/Linux	6
<b>API-Design</b>	<b>6</b>
Varför denna API-design?	6

## Om tjänsten

T-map visar en karta över hela världen, där en användare kan klicka på ett land för att se en tweet från Twitter på vad som trendar i det valda landet. Till exempel, om ordet *K-pop* trenda i Ryssland, kommer det populäraste tweetet för just *K-pop* att visas för användaren samt möjligheten att komma till den aktuella tweeten. Det finns även möjligheten för användaren att klicka på en knapp för att översätta det specifika tweetet till engelska.

### Syfte

Syftet med T-map är att få reda på vad som trendar i ett specifikt land!

Ett exempel på användningsområde för detta skulle vara för journalister som vill veta vad som pågår i ett annat land. Ett annat exempel skulle vara en globetrotter som vill veta vad den aktuella trenden är i det land hen ska till näst.

Om man klickar på ett land där modersmålet kommer fram kan man enkelt få texten översatt till engelska för att förstå kontexten.

## Den tekniska lösningen

### Flask

Flask är en enkel ramverk för att bygga webbapplikationer med Python. Trots dess huvudsyfte kan även enklare APIer byggas med ramverket.

### Bootstrap

Bootstrap är ett populärt bibliotek som används till frontend bygge, det innehåller färdigskriven HTML, CSS och JavaScript-kod. Med Bootstrap slipper exempelvis webbutvecklaren manipulera färg och storlek i CSS-koden, utan vänder sig direkt till Bootstrap och får en funktionell och färdig knapp som följer standarder och har passande färgkombination för både synlighet och stil.

### Ajax

Ajax förbättrar interaktivitet av webbapplikationer. Detta görs med asynkrona anrop som inte påverkar utseende av sidan om inte den valda beteende uppstår, till exempel om användare klickar på en viss del av sidan kan en del text visas som inte var synlig innan.

## Swagger

Swagger ger möjligheten att beskriva strukturen av ett API så att maskiner kan läsa och förstå den. Strukturen kan skrivas i JSON eller YAML. Swagger följer OpenAPI Specification som har i syfte att beskriva hur RESTful API:er bör beskrivas och dokumenteras.

## Beskrivning av de datakällor/externa API:er

### GoogleMaps API

Maps JavaScript API och Geocoding API är delar av de olika tjänster som erbjuds av Google Maps Platform. Med Javascript API kan kartan visas på en webbsida och manipuleras genom programmeringsspråket JavaScript. Till exempel kan utseendet på kartan förändras till satellitvy.

Med Geocoding APIs reverse geocoding kan adresser som "Malmö, Entré, Sverige" konverteras till geografiska koordinater (latitud 37.328489 och longitud -122.390428). Koordinaterna kan sedan användas för att placera en markör eller positionera om kartan. Geocoding API returnerar data i formatet JSON.

*Maps API* används för att kunna visa en google karta på vår webbsida, den låter användaren välja ett land genom att trycka på det. Sedan används *Geocoding API* för att omvandla latitud och longitud till adresser men framförallt till de fullständiga adresserna som innehåller ländernas namn, vilket sedan används för våra requests till de andra API:erna.

Samma API-nyckel används för båda API:er; hantering av de tillåtna API:er och nyckel-inställningar utfördes genom *Google Cloud Console*.

### Twitter API

Twitter API är ett API som tillåter tillgång till många funktionaliteter som berör Twitter. Till exempel kan tweets sorteras efter land och koordinater eller språk. Genom Twitter API kan även tweets och profiler skapas, samt skicka meddelanden till andra användare. Returdata från API:et är i formatet JSON som senare kan manipuleras utav andra programmeringsspråk, i vårt fall, genom Python3.

Twitter API kräver information om arbetet eller projektet som ska genomföras för att tillåta tillgång till API:et. Detta kan göra genom e-mail eller andra kommunikationsvägar. När tillgång har erbjudits erbjuds en token för att användas i samband med Twitter API:et.

## Google Cloud Translation API

Cloud Translation tar en text som ska översättas, och översätter den texten till ett valfritt språk. T-map översätter alltid texten till engelska då det är ett språk som många förstår. Det språk som texten är skriven i, upptäcker cloud translation automatiskt. Ett exempel är om texten är på japanska behöver utvecklaren eller användaren, inte specificera att det är en text på japanska, utan cloud translation upptäcker det automatiskt och översätter det sedan till engelska. Cloud Translation returnerar data i formatet JSON.

## Hur data används

När servern är igång så renderas kartan från Google Maps API. När användaren klickar på ett land hämtas information om longitud och latitud som sedan omvandlas till namn på stad och land via *Geocode*. Därefter hämtas endast namnet på de land användaren har klickat på. Det namnet skickas till vårt API:et som därefter gör kopplingen med vår backend som i sin tur gör en GET-metod via Flasks request-metod till Twitters API. Anropet till Twitters API görs i flera steg. Första steget är att söka på det namn på det land som efterfrågas för att få fram ett ID. Geocodes namn på länder och hur Twitter benämner länder stämmer inte alltid överens. Därför får stavningen kontrolleras efter hur Twitter vill och göra felhantering därefter. Till exempel får Sydkorea och Nordkorea omvandlas till *Korea*. Om användaren väljer att klicka på områden där det inte finns något land, exempelvis ett hav, eller där Twitter inte har data för trender i det specifika landet, visas upp ett meddelande av att adressen kunde inte hittas respektive att det inte finns några trender för det valda landet. Det ID som hämtats ut i tidigare steg används för att söka på vad för nyckelord som trender. Detta kan vara allt ifrån ett namn eller en hashtag. Det görs därefter en ny sökning genom Twitters API på nyckelordet för att få ut en specifik tweet. Därefter plockar sedan ut den tweeten som har mest *likes* och skickar tillbaka nyckelord, en url-länk samt texten på den utvalda tweet till vårt API som i sin tur skickar tillbaka det till vår frontend. Väl på frontend tas det hand om resultatet och visar det på hemsidan.

## Installationsinstruktioner

Öppna kod-filerna i Visual Studio Code eller övrigt textredigerare, klicka på “*file*” uppe i högra hörnet, klicka på “*Open Folder...*”, navigera till där du sparat filerna, klicka på mappen och sedan “*Välj mapp*”.

I projektmappen finns även en fil vid namnet *config.py* som innehåller alla API nycklar till de olika tjänster. Vänligen respektera att de flesta API:er fakturerar enligt antal anrop som görs. Denna fil kommer finnas tillgänglig vid lämnämningen av detta projekt.

## Körinstruktioner

### Windows

1. Säkerställa att `python --version` returnerar en version över 3.8.0
2. Aktivera den virtuella utvecklingsmiljö med `. myenv/Scripts/activate` OBS: Skulle du få felmeddelande i detta steget gör `python -m venv myenv` och sedan börja om på steget.
3. Se till att du står i rätt mapp(Tmap-main/T-map-main), annars navigera dig dit. Installera den rätta version av ramverket Flask och requests med `pip install -r requirements.txt`
4. Sätt igång Flask med `set FLASK_APP=src/app.py` eller om det görs i PowerShell `$env:FLASK_APP = "src/app.py"`
5. Kör servern genom att skriva `flask run`
6. Öppna webbadressen som returneras i konsolen, nu kan du explorera T-map.

### Mac/Linux

1. Säkerställa att `python --version` returnerar en version över 3.8.0
2. Skapa en virtuell miljö genom att skriva `python3 -m venv myenv`
3. Aktivera den virtuella utvecklingsmiljö med `. myenv/bin/activate`
4. Installera den rätta version av ramverket Flask och requests med `pip install -r requirements.txt`
5. Sätt igång Flask med `export FLASK_APP=app.py`. Om du får ett felmeddelande kan du behöva lägga till Tmap-main-mappen till din path. Detta görs genom att skriva `export PATH="$ (pwd):$PATH"`
6. Kör servern genom att skriva `flask run`
7. Öppna webbadressen som returneras i konsolen, nu kan du explorera T-map.

## API-Design

Vårt API har två endpoints:

- GET/tweet/<place>
  - För att få tweets från ett specifikt land.
- POST/translate/
  - För att översätta en text.

Vår frontend kommunicerar med vårt eget API för att göra två olika requests. Vårt API kommunicerar sedan med vårt backend. Backend i sin tur kommunicerar med de olika API:er som är kopplade till de webbtjänster som används. Den gör requests till de respektive webbtjänsterna med den data som kommit från vår frontend via vårt eget API. Därefter skickar backend tillbaka deras respons till vårt eget API som sedan frontend får.

## **Varför denna API-design?**

Alla i utvecklar-teamet hade en ganska bra idé på vad som skulle göras! Dock var tjänsten utmanande att bygga. Ingen i utvecklar-teamet visste exakt vad och hur det skulle göras, vilket gjorde det svårt att kommunicera det till de andra i teamet. Tillslut hade utvecklarna ett möte, samt pratade med Johan, och kom överens om designen ovan. Dock som med all planering var allt inte riktigt bestämt.

Anledningen till att utvecklingsteamet använder en GET för att få fram ord som trender samt hämta det mest populära tweetet på det som trender, är för att utvecklarna hämtar hem data från Twitter som inte kommer att ändras i detta skedet.

Att ett POST-anrop används för att översätta texten är för att data manipuleras från det som hämtats från Twitter-anropet till en översatt version av den hämtade texten.

## **API-dokumentation**

Vår API-dokumentation finns via vår webbtjänst. Då klickar användaren på docs-fliken som tar en vidare till vår dokumentation. Nedan finns även en länk till Swagger där vi gjort vår dokumentation:

<https://app.swaggerhub.com/apis/MiuMiuMiuMiuMiu/TweetMap/1.0.0>