Backend Project

Korean-English Translator

All for One
Choi Minjoo

Choi Minjoo (Judy Choi)

- Careers
  - 2018 **Working Holiday** in France
  - 2020 ~ **M.S** in Kangwon Univ
    - Intelligence Software Lab
      - **NLP (Machine Translation)**
  - 2021 ~ 2022 **Bering Lab** (NLP Researcher & Engineer)
- SNS
  - https://www.facebook.com/minjoo.choi.562/
  - https://github.com/Judy-Choi
  - https://www.linkedin.com/in/judy-choi/

In last class…

We did

- Trained machine translation model
- But is just ‘model’

- Let’s develop web translator service using this model!

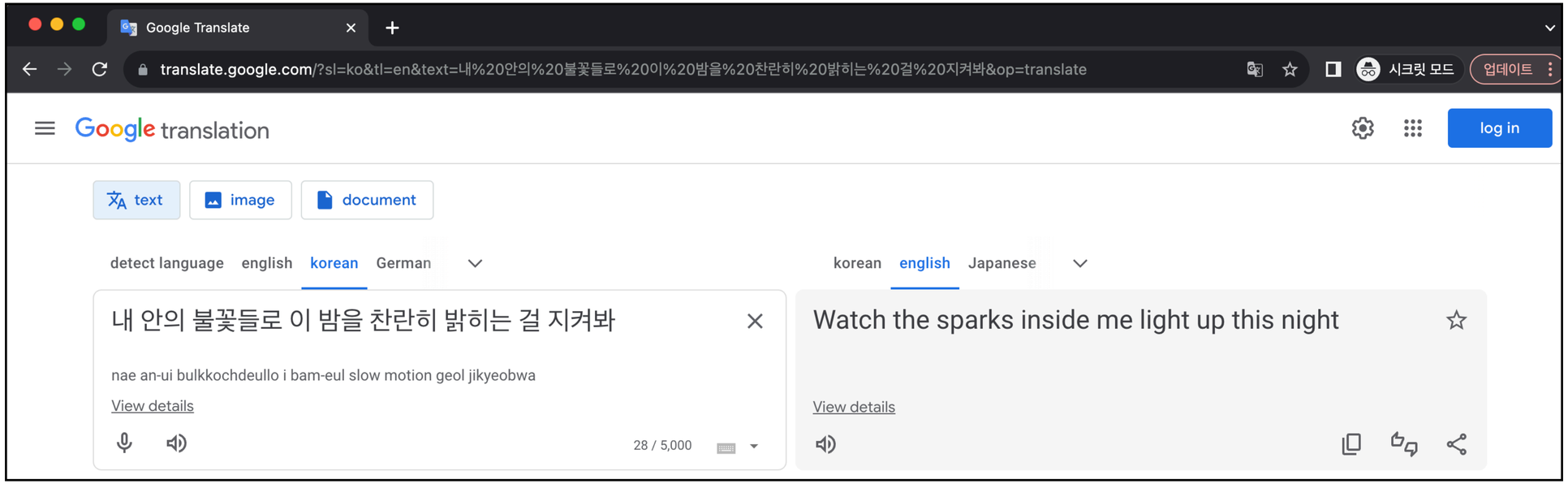
Contents

- Introduction
- Backend?
- Code Review

Introduction

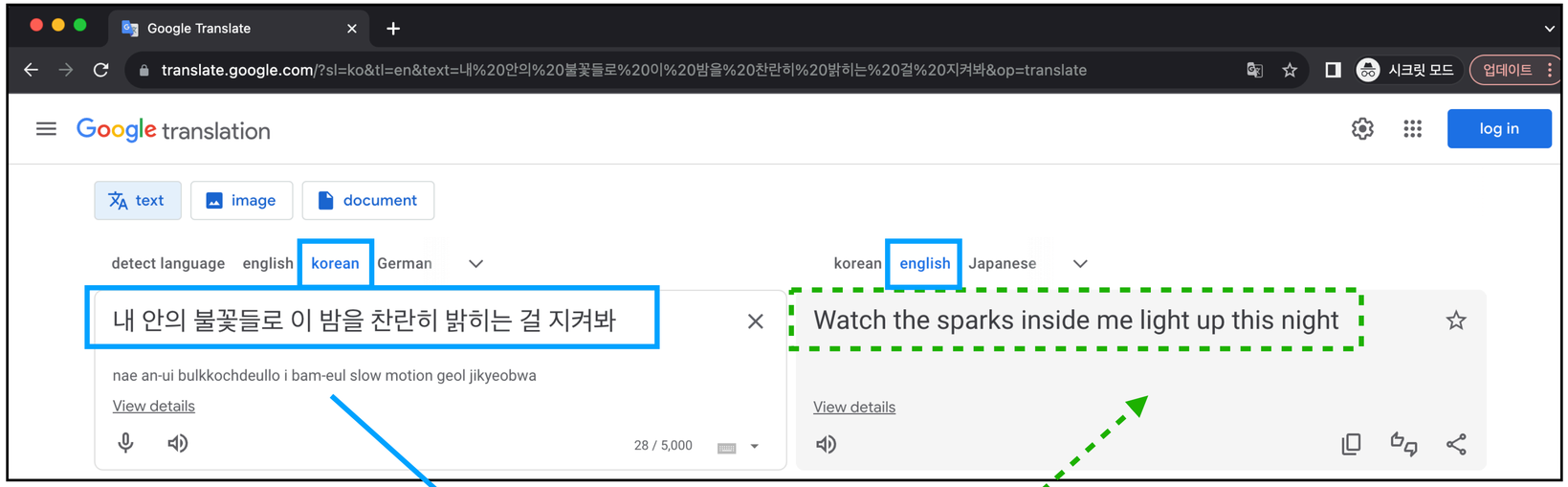
Web Translation Service

ex: Google translate



Web Translation Service

ex: Google translate



translate [Text] from Korean into English

return [Translated Text]

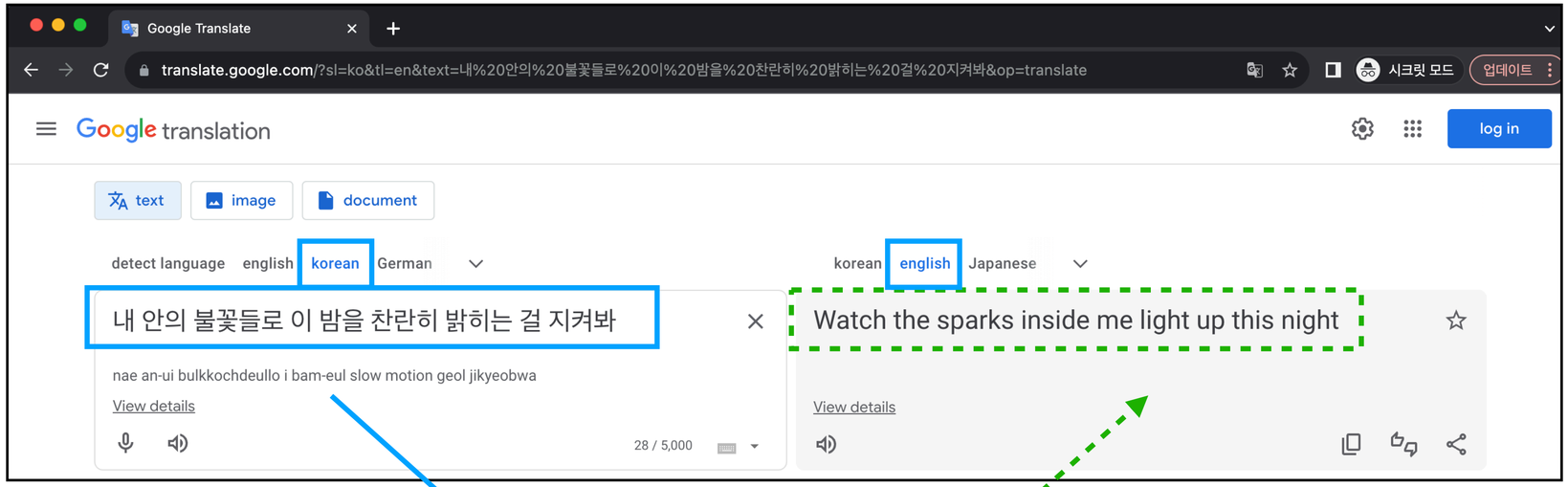
Google

Backend?

Web Translation Service

ex: Google translate

Client (Frontend)



translate [Text] from Korean into English

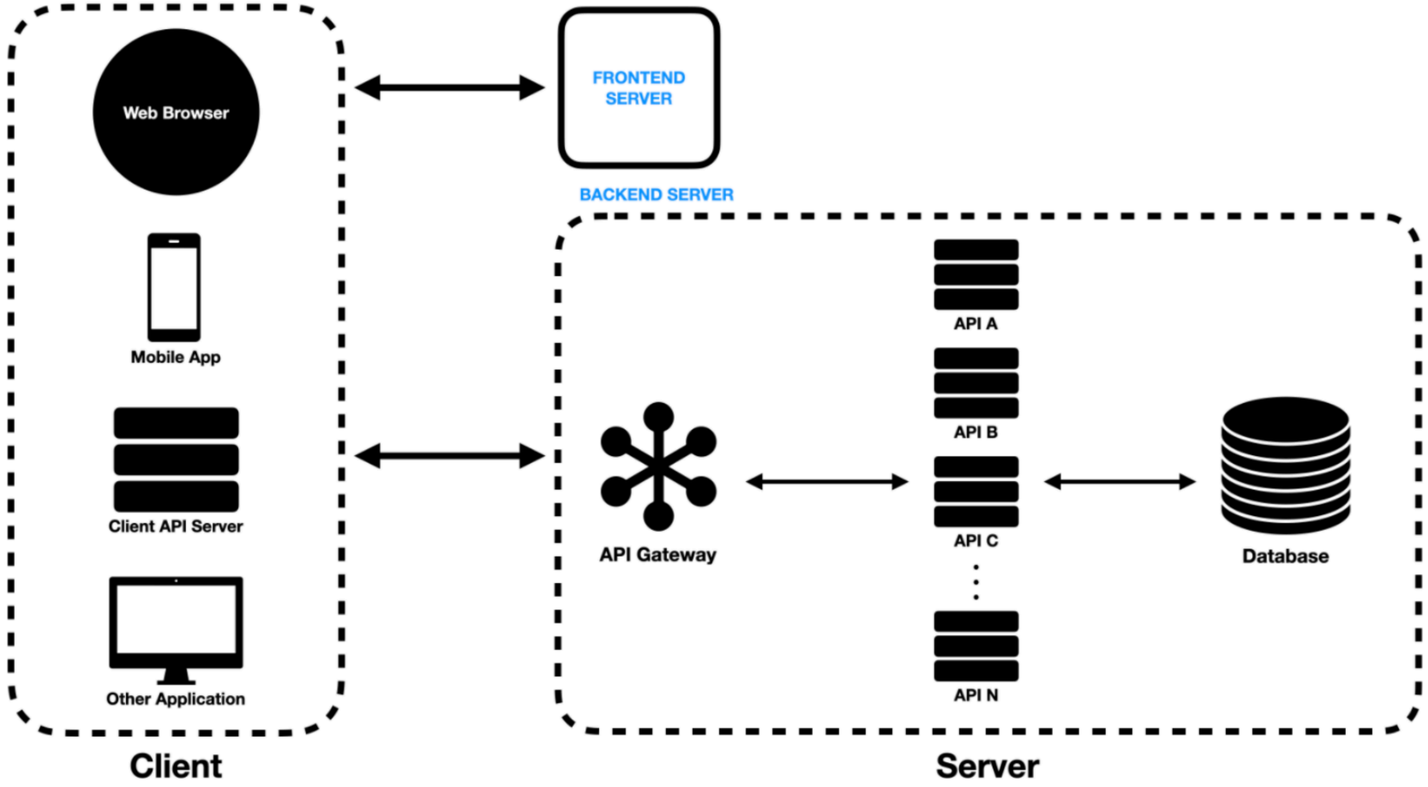
return [Translated Text]

Server (Backend)

Google

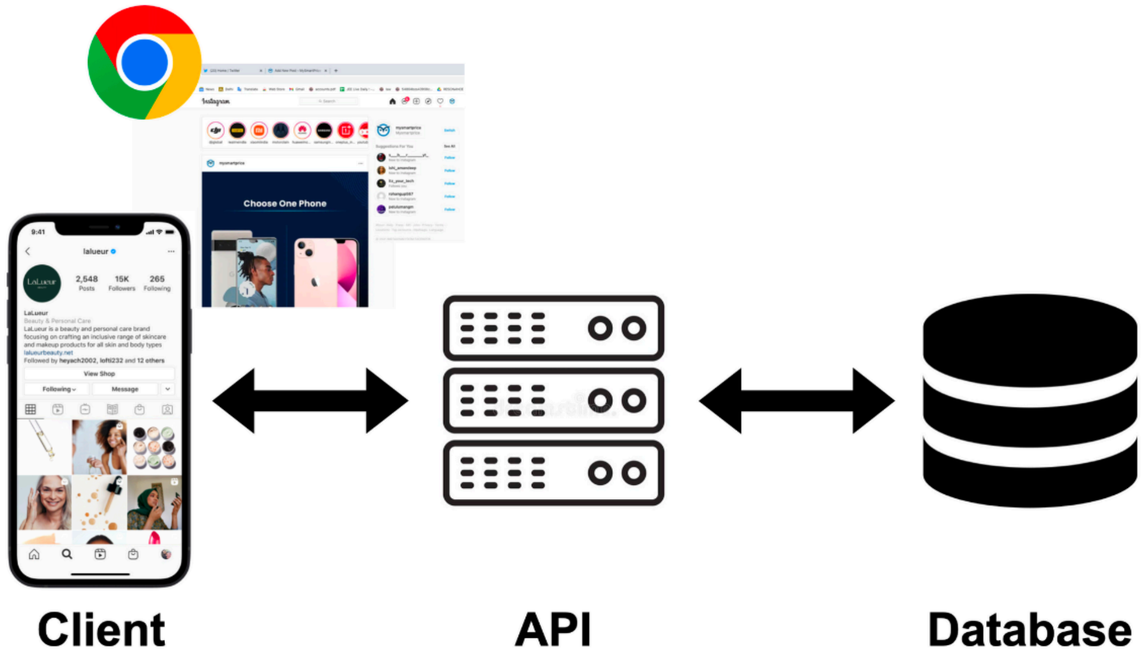
Web Development

Modern Web System Architecture



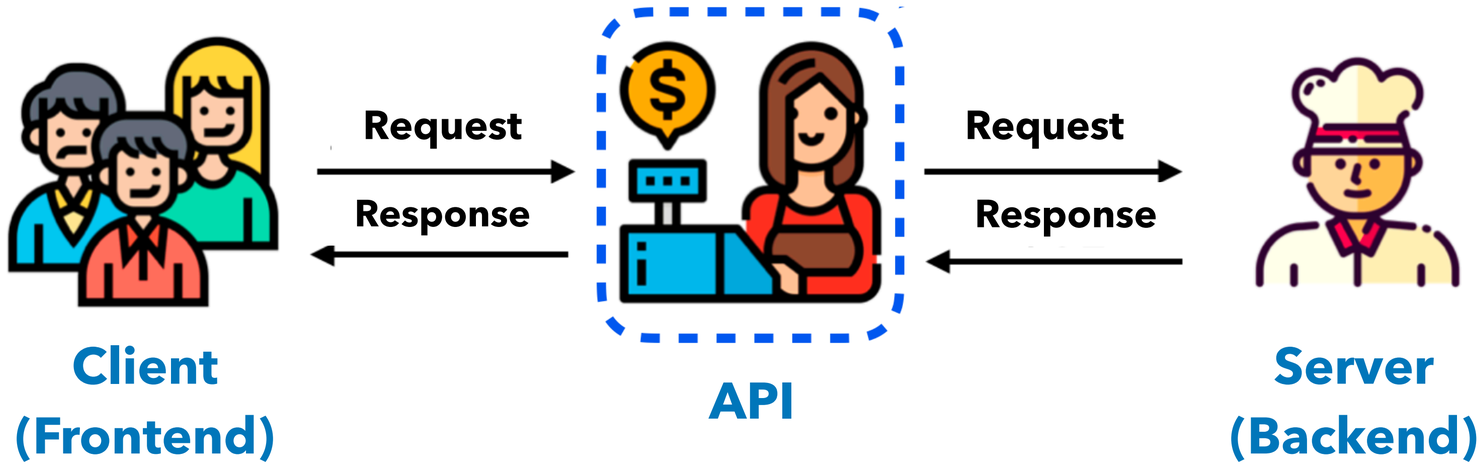
API

Application Programming Interface



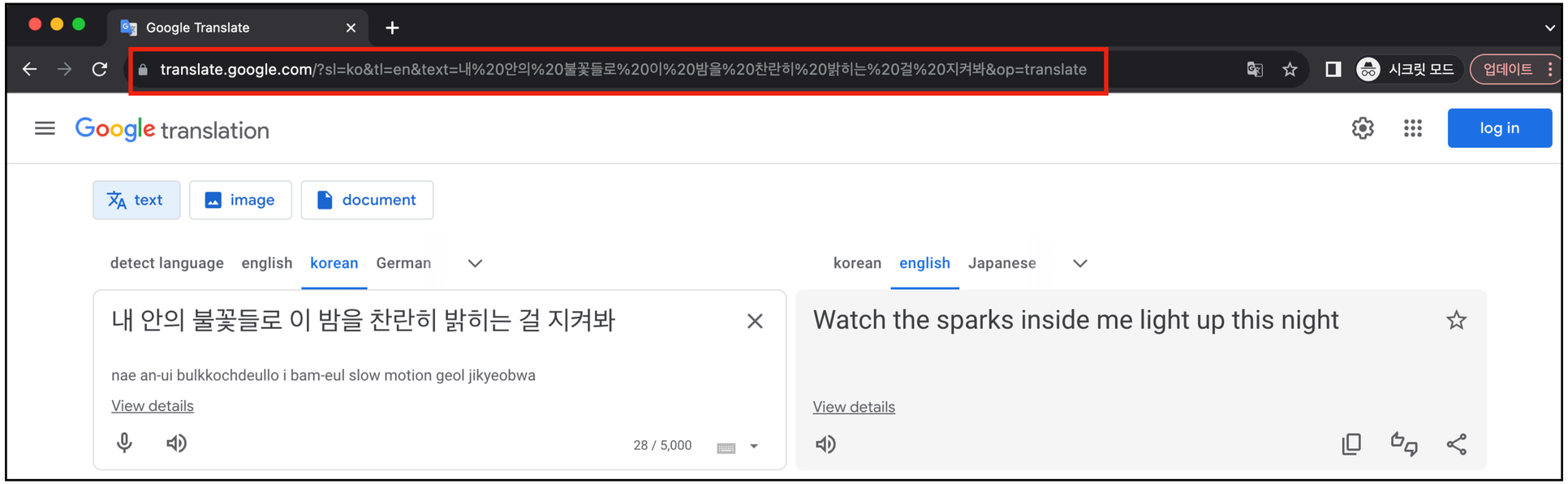
API

ex: Restaurant



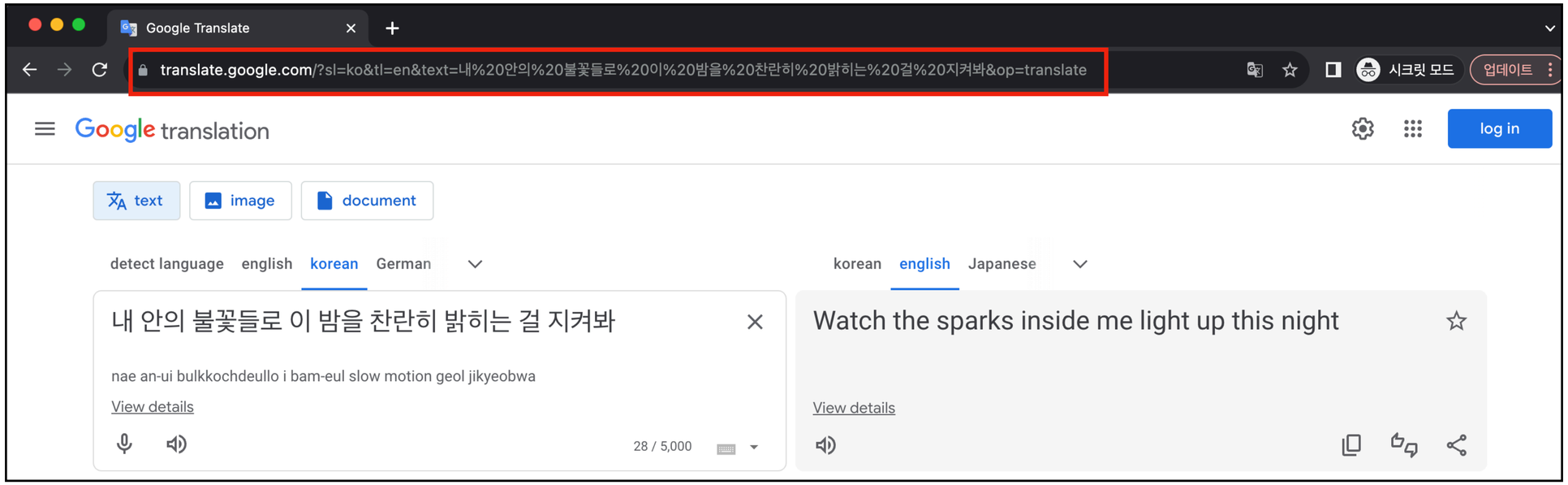
API

ex: Google translate



API

ex: Google translate



- sl (Source Language) = ko (Korean)
- tl (Target Language) = en (English)
- text = [Text Sentence]

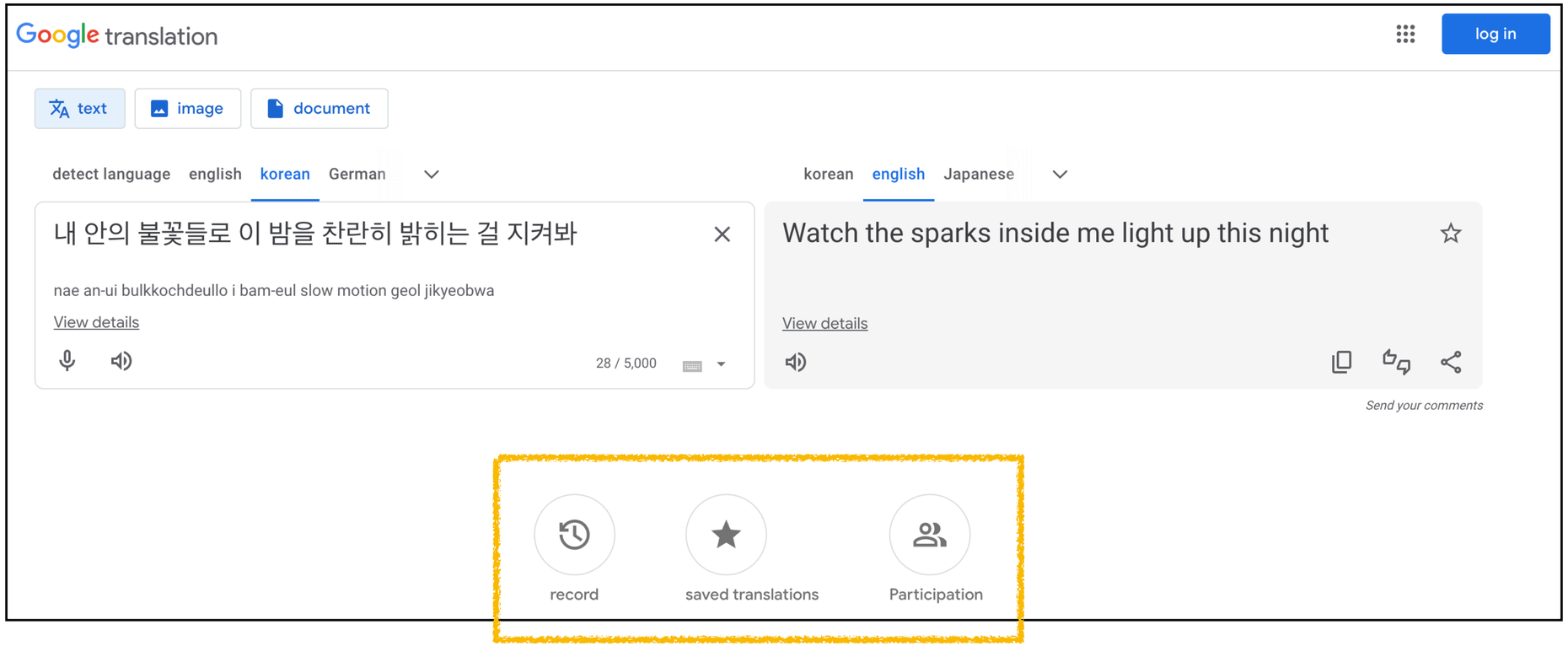
API

API Documentation

- ex: Swagger UI
  - https://cs93.site/docs

DataBase

ex: Google Translate



DataBase

Manage data

- RDBMS (Relational Database Management System)
  - MySQL, Maria, MongoDB, Postgre…

- API & Database should do 4 jobs with data:
  - ‘CRUD’
    - Create / Read / Update / Delete

To develop Backend…

Web Framework for API

- We use **FastAPI**
  - Support python
  - Fast
  - Latest Released
  - Support Swagger UI for API Documentation
  - Type Hunting (Check data type)

To develop Backend…

RDBMS

- We use **MySQL**
  - Most popular for SQL
  - Appropriate RDBMS for our tiny project
    - (ex: MongoDB is good for document..)
  - Also we can use **DBeaver** database tool

Code Review

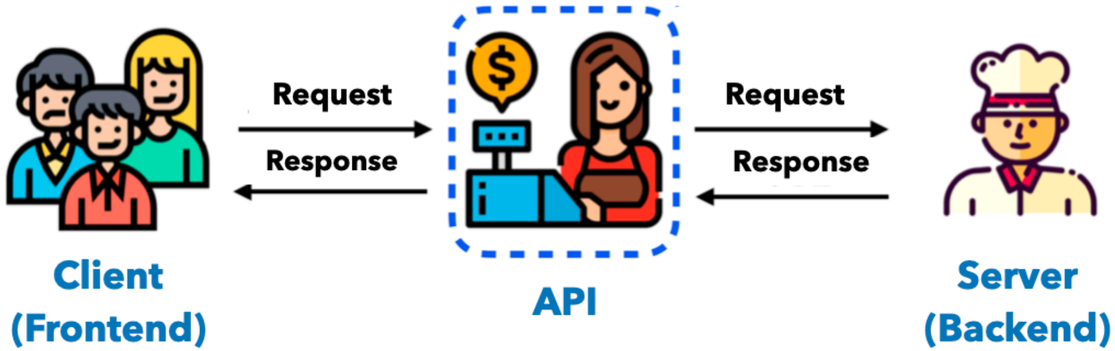
Backend

Requirements

Communicate with ‘Query Parameter’

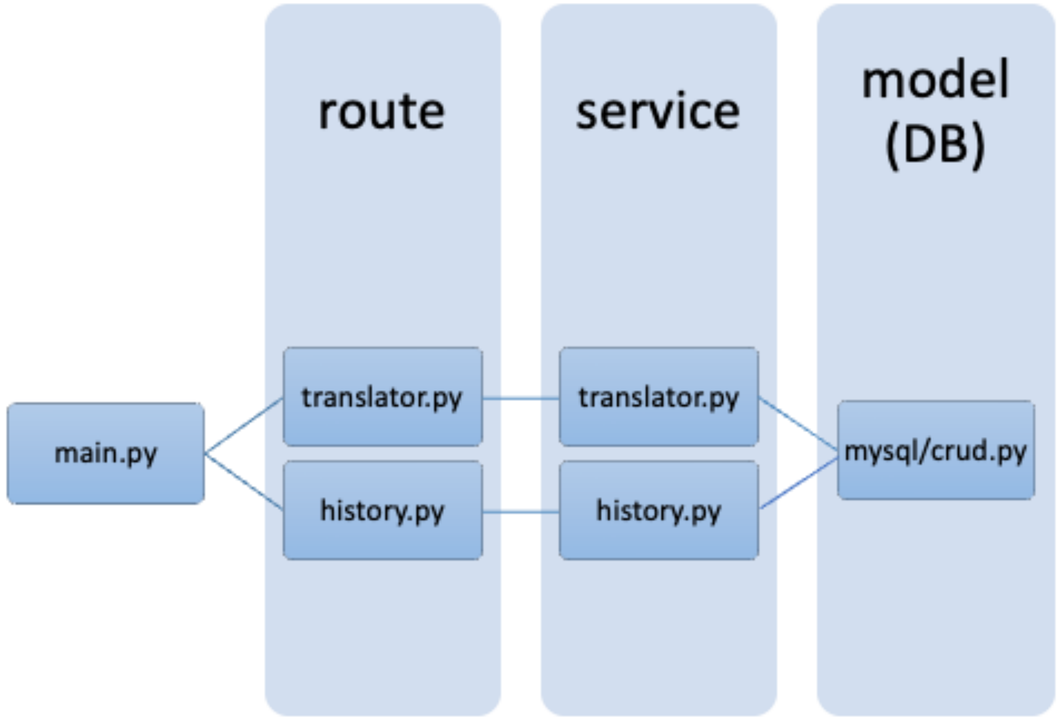
- https://cs93.site/translate?sl=ko&tl=en&text=집에가고싶다
  url route query parameter

- ex: Restaurant
  - url : cs93 restaurant
  - route : sit, order, pay
  - query parameter : detail option (ex: order/steak, pay/credit card…)



Architecture

FastAPI recommends...



Main

main.py

- Create FastAPI Instance
- Set Middleware
- Set ‘root’
- Add translate router to FastAPI instance

```python
# Allows CORS middleware
# example: "http://localhost"
# * : all
origins = ["*"]

# Create FastAPI Instance
app = FastAPI(title="NMT-API",
              description="Backend API for NMT")


# Setting Middleware
app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"]
)
```

```python
@app.get("/")
async def root():
    """
    root (health check)
    """
    return {"message": "Hello World"}

# Add translate router to FastAPI instance using 'include_router' method
# To access from outside
app.include_router(translate_router, prefix="/translate")
app.include_router(history_router, prefix="/history")
```

Route

route/translate.py

- Routing?
  1. Interpret the requested URL
  2. Run the appropriate function
  3. Return the result value

```python
@router.get("/")
async def translate_text(params: schemas.TranslateBase = Depends()) -> dict:
    """
    get machine translated text
    """
    mt_text = await translate.translate_text(params)
    return {"translated": mt_text}
```

- GET/POST
  1. Get query parameters in URL
     - https://cs93.site/translate/?sl=ko&tl=en&text=내 안의 불꽃들로 이 밤을 찬란히 밝히는 걸 지켜봐
  2. Run the function using query parameters
  3. Return the result value

Service

service/translate.py

- Translate Korean sentence into English sentence
- Store data in a database

```python
async def translate_text(params) -> str:
    """
    Translate a string & Save to DB
    """

    mt_text = await ctranslate(params)

    if isinstance(mt_text, str):
        await crud.create_translate(params, mt_text)
        return mt_text
```

Model

model/mysql/crud.py

- CRUD to MySQL DB
  - Create
    - store translation data in a Database

```python
from . import MYSQL_SESSION, models

db = MYSQL_SESSION()


async def create_translate(params, mt_text):
    """_summary_
    Create SQLAlchemy model instance and input data
    Args:
        params (_type_): _description_
        mt_text (_type_): _description_
    """
    query = models.Translate(src_lang=params.sl,
                             src_text=params.text,
                             tgt_lang=params.tl,
                             mt_text=mt_text
                             )
    try:
        # db 세션 지정
        db.add(query)
        db.commit()
        db.refresh(query)
    except IntegrityError as error:
        raise ValueError(
            "Error occurs when data that goes against the primary key enters the DB") from error
    # except IntegrityError:
    #     raise
    except OperationalError:
        db.rollback()
        raise
    finally:
        db.close()
```

Model Serving

Model Serving

MLOps

- **M**achine **L**earning **Op**eration**s**
  - Core function of Machine Learning for production-
    - engineering, streamlining, maintaining, monitoring

- MLOps Framework:
  - TorchServe, Triton, BentoML, Tensorflow Serving…

https://www.databricks.com/glossary/mlops

CTranslate
by OpenNMT

- C++ and Python library for efficient inference with Transformer models

```
translator = ctranslate2.Translator(
    f"nmt/model/bin/{params.sl}-{params.tl}", device="cpu")
# pylint: disable=too-many-function-args
sp_sl = spm.SentencePieceProcessor(
    f"nmt/model/sentencepiece/sp_model.{params.sl}")
sp_tl = spm.SentencePieceProcessor(
    f"nmt/model/sentencepiece/sp_model.{params.tl}")

input_tokens = sp_sl.Encode(params.text, out_type=str)
results = translator.translate_batch([input_tokens])

output_tokens = results[0].hypotheses[0]
output_text = sp_tl.Decode(output_tokens)

return output_text
```

Q&A

https://www.facebook.com/minjoo.choi.562/

https://github.com/Judy-Choi

https://www.linkedin.com/in/judy-choi/