



## Дизајн и архитектура на софтвер

### Домашна работа 4

## Дизајн шаблон - Singleton

Singleton шаблонот се користи во **News Scraper** микросервисот, кој служи за превземање на извештаите на компаниите. Тој се користи за да се осигури дека само една инстанца на **Translator** ќе постои во целата апликација, која нејзина примарна цел е да ги преведува извештаите од англиски на македонски јазик. Ова е важно затоа што при самото scrape-ање, задачите се поделени на сите јадра од процесорот, и секое јадро испраќа асинхрони барања. Со singleton, се избегнува создавање нови инстанци секој пат, што заштедува ресурси и го олеснува споделувањето на истата инстанца меѓу различните задачи (асинхрони task-ови).

*helpers.py*

```
async def translate_mk_to_en(text):
    if text is not None:
        if isinstance(text, str) and text.strip():
            if "..." in text:
                return None

            normalized_text = normalize_cyrillic(text)
            try:
                translator = TranslatorSingleton().get_instance()
                max_length = 5000
                if len(normalized_text) > max_length:
                    parts = [normalized_text[i:i + max_length] for i in range(0, len(normalized_text), max_length)]
                    translated_parts = [translator.translate(part, src='mk', dest='en').text for part in parts]
                    return ' '.join(translated_parts)

                detected_lang = translator.detect(normalized_text).lang
                if detected_lang == 'mk':
                    return translator.translate(normalized_text, src='mk', dest='en').text
                return normalized_text
            except Exception as e:
                return normalized_text
    else:
        return None
```



### *translator\_singleton.py*

```
import asyncio
from googletrans import Translator

3 usages  ▴ Daniel Ilievski *
class TranslatorSingleton:
    _instance = None
    _lock = asyncio.Lock()

    1 usage new *
    @classmethod
    def get_instance(cls):
        with cls._lock:
            if cls._instance is None:
                cls._instance = TranslatorSingleton()
            return cls._instance.translator

    ▴ Daniel Ilievski *
    def __init__(self):
        self.translator = Translator()
```