

Watson API Tone Analyzer

IBM API 사용 가이드를 먼저 읽고 와주세요!

Tone Analyzer란

문장의 내용을 통해 감정을 분석하는 기술입니다.

Tone Analyzer 분석 코드

```
import json
```

```
from ibm_watson import ToneAnalyzerV3
```

```
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

```
authenticator = IAMAuthenticator('발급받은 API키 입력')
```

```
tone_analyzer = ToneAnalyzerV3(
```

```
    version='2017-09-21',
```

```
    authenticator=authenticator
```

```
)
```

```
tone_analyzer.set_default_headers({'x-watson-learning-opt-out': "true"}) #수집 못하게 하기
```

```
tone_analyzer.set_service_url('발급받은 URL 입력')
```

```
text = 'I got paid yesterday. \
```

```
'So, I bought the game looking forward to. \
```

```
'But today, Playstartion 5 is broken. \
```

```
tone_analysis = tone_analyzer.tone(
```

```
    {'text': text},
```

```
    content_type='application/json',
```

```
    sentences = True,
```

```
    accept_language = "en"
```

```
).get_result()
```

```
print(json.dumps(tone_analysis, indent=2))
```

#문단이 끝나지 않았음을 \로 표시합니다.

##(☆중요) 문장 구분은 문장 마지막 띄워쓰기 한 칸으로 구분되어집니다.

#입력 형식이 무엇인가 -> text/plain, text/html도 가능합니다.

#문장별로 분석 True가 디폴트 False시 문단 전체의 Tone분석합니다.

tone name을 원하는 지역 언어로 변경 *근데 영어 아니면 깨집니다.. 미지정시 영어(en)

입력형태에 대해서...

```
import json
```

```
from ibm_watson import ToneAnalyzerV3
```

```
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

```
authenticator = IAMAuthenticator('발급받은 API키 입력')
```

```
tone_analyzer = ToneAnalyzerV3(
```

```
    version='2017-09-21',
```

```
    authenticator=authenticator
```

```
)
```

```
tone_analyzer.set_default_headers({'x-watson-learning-opt-out': "true"})
```

```
tone_analyzer.set_service_url('발급받은 URL 입력')
```

```
text = 'I got paid yesterday. \'
```

```
      'So, I bought the game looking forward to. \'
```

```
      'But today, Playstartion 5 is broken. \'
```

```
tone_analysis = tone_analyzer.tone(
```

```
    text,
```

```
    content_type='text/plain;charset=utf-8',    #입력 형식을 텍스트로 받은 것, text/plain의 경우 인코딩 방식을 지정해 줄 수도
```

```
    있습니다..
```

```
    sentences = True,
```

```
    accept_language = "en"
```

```
).get_result()
```

```
print(json.dumps(tone_analysis, indent=2))
```

Tone Analyzer 분석 결과

```
{
  "document_tone": {
    "tones": [
      {
        "score": 0.555029,
        "tone_id": "sadness",
        "tone_name": "Sadness"
      }
    ]
  },
  "sentences_tone": [
    {
      "sentence_id": 0,
      "text": "I got paid yesterday.",
      "tones": []
    },
    {
      "sentence_id": 1,
      "text": "So, I bought the game looking forward to.",
      "tones": [
        {
          "score": 0.590473,
          "tone_id": "joy",
          "tone_name": "Joy"
        }
      ]
    }
  ],
  {
    "sentence_id": 2,
    "text": "But today, Playstartion 5 is broken.",
    "tones": [
      {
        "score": 0.731701,
        "tone_id": "sadness",
        "tone_name": "Sadness"
      }
    ]
  }
]
```

응용) 파일에서 텍스트를 읽어와 분석

```
import json
from ibm_watson import ToneAnalyzerV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
```

```
authenticator = IAMAuthenticator('발급받은 API키 입력')
tone_analyzer = ToneAnalyzerV3(
    version='2017-09-21',
    authenticator=authenticator
)
tone_analyzer.set_default_headers({'x-watson-learning-opt-out': "true"}) #수집 못하게 하기
tone_analyzer.set_service_url('발급받은 URL입력')
```

```
text=""
f= open("./tone_analyzer_input.txt",'r'); #파일에서 한 문장씩 읽어서 처리하는 예제, 내용은 앞선 text와 동일합니다.
```

```
while True:
    line = f.readline()
    if not line: break
    text += line+' '
```

```
tone_analysis = tone_analyzer.tone(
    text,
    content_type='text/plain;charset=utf-8', #입력형식이 무엇인가 -> text/plain, text/html도 가능합니다.
    sentences = True, #문장별로 분석 True가 디폴트 False시 문단 전체의 Tone분석합니다.
    accept_language = "en" #tone name을 원하는 지역 언어로 변경 *근데 영어 아니면 깨져서 보입니다.. 미지정시 영어 (en)임
).get_result()
print(json.dumps(tone_analysis, indent=2))
```

파일에서 읽어와 분석한 결과 → 동일하다.

```
{
  "document_tone": {
    "tones": [
      {
        "score": 0.555029,
        "tone_id": "sadness",
        "tone_name": "Sadness"
      }
    ]
  },
  "sentences_tone": [
    {
      "sentence_id": 0,
      "text": "I got paid yesterday.",
      "tones": []
    },
    {
      "sentence_id": 1,
      "text": "So, I bought the game looking forward to.",
      "tones": [
        {
          "score": 0.590473,
          "tone_id": "joy",
          "tone_name": "Joy"
        }
      ]
    }
  ]
},
{
  "sentence_id": 2,
  "text": "But today, Playstartion 5 is broken.",
  "tones": [
    {
      "score": 0.731701,
      "tone_id": "sadness",
      "tone_name": "Sadness"
    }
  ]
}
]
```


응용) 텍스트/ json이 아닌 Utterance도 가능

먼저 **utterance** 객체란 페이스북 댓글에서 쓰고 있는 것으로,
GitHub Issue 기반으로 **comment**를 작성 할 수 있게 해줍니다.

SNS 댓글 크롤링 후 감정 분석시 사용 할 수 있습니다.

응용) Utterance 분석

```
import json
from ibm_watson import ToneAnalyzerV3
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
authenticator = IAMAuthenticator('발급받은 API키 입력')
tone_analyzer = ToneAnalyzerV3(
    version='2017-09-21',
    authenticator=authenticator
)
tone_analyzer.set_default_headers({'x-watson-learning-opt-out': "true"})
tone_analyzer.set_service_url('발급받은 URL 입력')
utterances = [
    {
        "text": "Hello, I'm having a problem with your product.",
        "user": "customer"
    },
    {
        "text": "OK, let me know what's going on, please.",
        "user": "agent"
    },
    {
        "text": "Well, nothing is working :((",
        "user": "customer"
    },
]
utterance_analyses = tone_analyzer.tone_chat(utterances).get_result()
print(json.dumps(utterance_analyses, indent=2))
```

#수집 못하게 하기

#크롤링한 utterances 데이터

응용) Utterance 분석 결과

```
{
  "utterances_tone": [
    {
      "utterance_id": 0,
      "utterance_text": "Hello, I'm having a problem with your product.",
      "tones": [
        {
          "score": 0.686361,
          "tone_id": "polite",
          "tone_name": "Polite"
        }
      ]
    },
    {
      "utterance_id": 1,
      "utterance_text": "OK, let me know what's going on, please.",
      "tones": [
        {
          "score": 0.92724,
          "tone_id": "polite",
          "tone_name": "Polite"
        }
      ]
    },
  ],
}
```

```
{
  "utterance_id": 2,
  "utterance_text": "Well, nothing is working :((",
  "tones": [
    {
      "score": 0.997795,
      "tone_id": "sad",
      "tone_name": "Sad"
    }
  ]
}
```

문장별 분위기(tone) 분석되어 나옴
score는 신뢰도 입니다.