

API-Dokumentation Detectora

Kurze Dokumentation: API zur Textbewertung

Beschreibung:

Die API ermöglicht es, Texte zu analysieren und Wahrscheinlichkeiten für ihre Authentizität (echt/fake) zurückzugeben. Eingaben und Ausgaben erfolgen im JSON-Format.

Anfrage an die API

1. **URL:** /watson
2. **Methode:** POST
3. **Headers:**
 - a. APIKey: Erforderlich für die Authentifizierung.
 - b. Content-Type: application/json.
4. **JSON-Body:**
 - a. query: Der zu analysierende Text.
 - i. **Beispiel:** "Dein Text"

Antwort der API

- **Erfolgreiche Anfrage (200):**

```
json
Code kopieren
{
  "all_tokens": 150,
  "used_tokens": 100,
  "real_probability": 0.85,
  "fake_probability": 0.15
}
```

- all_tokens: Anzahl aller Tokens im Text.
 - used_tokens: Verarbeitete Tokens (je nach Token-Limit des Modells).
 - real_probability: Wahrscheinlichkeit, dass der Text authentisch ist.
 - fake_probability: Wahrscheinlichkeit, dass der Text generiert ist.
- **Fehlerhafte Anfrage (400):**

json

Code kopieren

```
{
  "error": "query parameter is required"
}
```

○ Ursache: Der query-Parameter fehlt.

- **Serverfehler (500):**

json

Code kopieren

```
{
  "error": "Fehlerbeschreibung"
}
```

○ Ursache: Interner Fehler während der Verarbeitung.

Python-Beispiel für die API-Anfrage

```
import requests
```

```
api_url = "https://backendkidetektor-apim.azure-api.net/watson"
```

```
headers = {
  'APIKey': "DeinAPIKey, # Authentifizierungsschlüssel
  'Content-Type': 'application/json'
}
```

```
data = {
  'query': "Dein Text."
}
```

```
try:
```

```
    response = requests.post(api_url, headers=headers, json=data)
```

```
    if response.status_code == 200:
```

```
        print("Erfolg:")
```

```
        print(response.json())
```

```
    elif response.status_code == 400:
```

```
        print("Fehlerhafte Anfrage:", response.json())
```

```
    else:
```

```
        print(f"Fehler {response.status_code}:", response.json())
```

```
except Exception as e:  
    print("Fehler beim Verbinden mit der API:", str(e))
```