

SPELLING & GRAMMAR CHECKER

Demo of App:

https://tmccentral-my.sharepoint.com/:v/g/personal/habib_urrehman_tmcltd_com/Eb_DJqokq4pFjhT4YkjrAQYBwUdBD7LUx8Jo5E2KHe6qbw?e=x8ABHX

We have integrated the free version of the “LanguageTool” API into our application. Below is the rewritten code for the view and controller components.

View Code:

```
<mvc:View controllerName="scgc.project1.controller.View1"
    xmlns:mvc="sap.ui.core.mvc" displayBlock="true"
    xmlns="sap.m">
    <Page id="page" title="Spelling and Grammar Checker">
        <content>
            <VBox>
                <Input id="inputText" placeholder="Enter text to check"
liveChange="onLiveChange" />
                <Text id="resultText" text=""/>
            </VBox>
        </content>
    </Page>
</mvc:View>
```

Controller Code:

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/m/MessageToast",
    "sap/ui/model/json/JSONModel"
], function (Controller, MessageToast, JSONModel) {
    "use strict";

    return Controller.extend("scgc.project1.controller.View1", {
        onInit: function () {
            // Initialize the model to store the grammar check results
            var oModel = new JSONModel();
            this.getView().setModel(oModel, "grammarModel");
        }
    });
});
```

```

        // Initialize debounce function
        this._debounceGrammarCheck = this._debounce(this._checkGrammar, 500);
    },

    onLiveChange: function (oEvent) {
        var oInput = oEvent.getSource();
        var sText = oInput.getValue();
        if (sText.endsWith(" ")) {
            this._debounceGrammarCheck(sText.trim(), oInput);
        }
    },

    _checkGrammar: function (sText, oInput) {
        var that = this;

        // Call the LanguageTool API
        fetch("https://api.languagetool.org/v2/check", {
            method: "POST",
            headers: {
                "Content-Type": "application/x-www-form-urlencoded"
            },
            body: `text=${encodeURIComponent(sText)}&language=en-US&context=sentence`
        })
        .then(response => response.json())
        .then(response => that._handleGrammarCheckResponse(oInput, response))
        .catch(error => {
            MessageToast.show("Error checking grammar and spelling.");
            console.error("Error checking grammar and spelling:", error);
        });
    },

    _handleGrammarCheckResponse: function (oInput, response) {
        var oModel = this.getView().getModel("grammarModel");
        oModel.setData(response);

        var aMatches = response.matches;
        var sText = oInput.getValue();
        var sCorrectedText = sText;
        var aCorrections = [];

        if (aMatches.length > 0) {
            // Gather all corrections
            aMatches.forEach(match => {
                if (match.replacements && match.replacements.length > 0) {

```

```

        var bestReplacement = match.replacements.reduce((best,
current) => {
        return (current.value.length > best.value.length)?
current : best;
        });
        aCorrections.push({
            offset: match.offset,
            length: match.length,
            replacement: bestReplacement.value
        });
    }
});

    // Sort corrections by offset in descending order to avoid
overlapping issues
    aCorrections.sort((a, b) => b.offset - a.offset);

    // Apply corrections
    aCorrections.forEach(correction => {
        sCorrectedText = sCorrectedText.substring(0,
correction.offset) +
        correction.replacement +
        sCorrectedText.substring(correction.offset +
correction.length);
    });

    oInput.setValue(sCorrectedText);
}

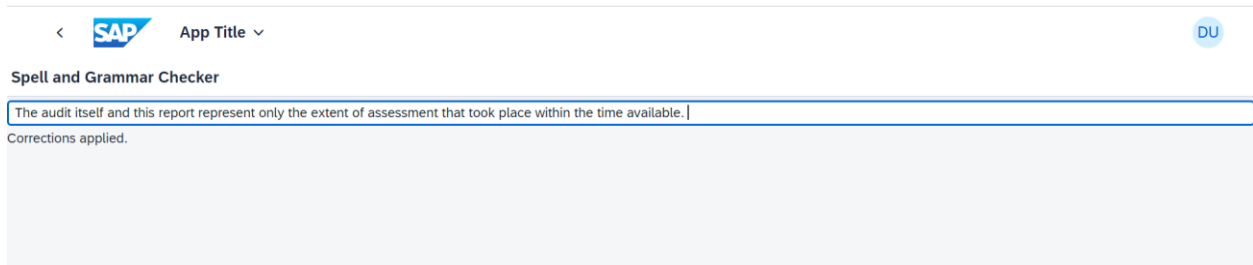
    var sResultText = aMatches.length > 0? "Corrections applied." : "No
issues found.";
    this.getView().byId("resultText").setText(sResultText);
},

    _debounce: function (func, wait) {
        var timeout;
        return function () {
            var context = this, args = arguments;
            clearTimeout(timeout);
            timeout = setTimeout(function () {
                timeout = null;
                func.apply(context, args);
            }, wait);
        };
    }
}

```

```
});  
});
```




Fiori App View:





LanguageTool & other API Plans:


NOTE: First You have to sign up for RAPID API Key.


1- JSPELL Checker:

 **JSpell Checker** Freemium   (2)


 9.5 Popularity

 100% Service Level

 933ms Latency

 100% Test

BASIC
\$0.00 / mo

PRO 
\$4.99 / mo

ULTRA
\$19.99 / mo

Note that this API is free to use but has a limit of 15 requests per minute.

Choose the plan that's right for You

RapidAPI partners directly with API providers to give you no-fuss, transparent pricing.

Basic	Pro ★ Recommended	Ultra
\$0.00 /mo	\$4.99 /mo	\$19.99 /mo
Requests ⓘ 300 / Month + \$0.01	Requests ⓘ 10,000 / Month + \$0.005	Requests ⓘ 100,000 / Month + \$0.001
Rate Limit 15 requests per minute	Rate Limit -	Rate Limit -
Start Free Plan	Choose This Plan	Choose This Plan

CODE:

```
import axios from 'axios';
```

```
const options = {
  method: 'POST',
  url: 'https://jspell-checker.p.rapidapi.com/check',
  headers: {
    'x-rapidapi-key': 'Sign Up for Key',
    'x-rapidapi-host': 'jspell-checker.p.rapidapi.com',
    'Content-Type': 'application/json'
  },
  data: {
    '0': '{',
    '1': '\n',
    '2': ' ',
    '3': '"',
    '4': 'I',
    '5': 'a',
    '6': 'n',
    '7': 'g',
    '8': 'u',
    '9': 'a',
    '10': 'g',
    '11': 'e',
    '12': '"',
```

'13: ':,
'14: '' ,
'15: ''',
'16: 'e',
'17: 'n',
'18: 'U',
'19: 'S',
'20: ''',
'21: ',,
'22: '\n',
'23: ' ',
'24: ''',
'25: 'f',
'26: 'i',
'27: 'e',
'28: 'l',
'29: 'd',
'30: 'v',
'31: 'a',
'32: 'l',
'33: 'u',
'34: 'e',
'35: 's',
'36: ''',
'37: ':',
'38: '' ,
'39: ''',
'40: 't',
'41: 'h',
'42: 'i',
'43: 's',
'44: 's',
'45: '' ,
'46: 'i',
'47: 's',
'48: '' ,
'49: 'i',
'50: 'n',
'51: 't',
'52: 'r',
'53: 'e',
'54: 's',
'55: 't',
'56: 'i',
'57: 'n',
'58: 'g',

'59: ''',
'60: ',',
'61: '\n',
'62: ' ',
'63: ''',
'64: 'c',
'65: 'o',
'66: 'n',
'67: 'f',
'68: 'i',
'69: 'g',
'70: ''',
'71: ':',
'72: ''',
'73: '{',
'74: '\n',
'75: ' ',
'76: ' ',
'77: ''',
'78: 'f',
'79: 'o',
'80: 'r',
'81: 'c',
'82: 'e',
'83: 'U',
'84: 'p',
'85: 'p',
'86: 'e',
'87: 'r',
'88: 'C',
'89: 'a',
'90: 's',
'91: 'e',
'92: ''',
'93: ':',
'94: ''',
'95: 'f',
'96: 'a',
'97: 'I',
'98: 's',
'99: 'e',
'100: ',',
'101: '\n',
'102: ' ',
'103: ' ',
'104: ''',

'105': 'i',
'106': 'g',
'107': 'n',
'108': 'o',
'109': 'r',
'110': 'e',
'111': 'T',
'112': 'r',
'113': 'r',
'114': 'e',
'115': 'g',
'116': 'u',
'117': 'T',
'118': 'a',
'119': 'r',
'120': 'C',
'121': 'a',
'122': 'p',
'123': 's',
'124': '"',
'125': ':',
'126': "'",
'127': 'f',
'128': 'a',
'129': 'T',
'130': 's',
'131': 'e',
'132': ',',
'133': '\n',
'134': ' ',
'135': ' ',
'136': '"',
'137': 'i',
'138': 'g',
'139': 'n',
'140': 'o',
'141': 'r',
'142': 'e',
'143': 'F',
'144': 'i',
'145': 'r',
'146': 's',
'147': 't',
'148': 'C',
'149': 'a',
'150': 'p',

'151': 's',
'152': '',
'153': ':',
'154': '',
'155': 't',
'156': 'r',
'157': 'u',
'158': 'e',
'159': ',',
'160': '\n',
'161': ' ',
'162': ' ',
'163': '',
'164': 'i',
'165': 'g',
'166': 'n',
'167': 'o',
'168': 'r',
'169': 'e',
'170': 'N',
'171': 'u',
'172': 'm',
'173': 'b',
'174': 'e',
'175': 'r',
'176': 's',
'177': '',
'178': ':',
'179': '',
'180': 't',
'181': 'r',
'182': 'u',
'183': 'e',
'184': ',',
'185': '\n',
'186': ' ',
'187': ' ',
'188': '',
'189': 'i',
'190': 'g',
'191': 'n',
'192': 'o',
'193': 'r',
'194': 'e',
'195': 'U',
'196': 'p',

'197': 'p',
'198': 'e',
'199': 'r',
'200': '',
'201': ':',
'202': '',
'203': 'f',
'204': 'a',
'205': 'l',
'206': 's',
'207': 'e',
'208': ',',
'209': '\\n',
'210': ' ',
'211': ' ',
'212': '',
'213': 'i',
'214': 'g',
'215': 'n',
'216': 'o',
'217': 'r',
'218': 'e',
'219': 'D',
'220': 'o',
'221': 'u',
'222': 'b',
'223': 'l',
'224': 'e',
'225': '',
'226': ':',
'227': '',
'228': 'f',
'229': 'a',
'230': 'l',
'231': 's',
'232': 'e',
'233': ',',
'234': '\\n',
'235': ' ',
'236': ' ',
'237': '',
'238': 'i',
'239': 'g',
'240': 'n',
'241': 'o',
'242': 'r',

```

    '243': 'e',
    '244': 'W',
    '245': 'o',
    '246': 'r',
    '247': 'd',
    '248': 's',
    '249': 'W',
    '250': 'i',
    '251': 't',
    '252': 'h',
    '253': 'N',
    '254': 'u',
    '255': 'm',
    '256': 'b',
    '257': 'e',
    '258': 'r',
    '259': 's',
    '260': '"',
    '261': ':',
    '262': "'",
    '263': 't',
    '264': 'r',
    '265': 'u',
    '266': 'e',
    '267': '\n',
    '268': ' ',
    '269': '}',
    '270': '\n',
    '271': '}'
  }
};

try {
  const response = await axios.request(options);
  console.log(response.data);
} catch (error) {
  console.error(error);
}

```

2- BING SPELL CHECK:

Choose the plan that's right for You

RapidAPI partners directly with API providers to give you no-fuss, transparent pricing.

Basic	Pro	Ultra	Mega
\$0.00 /mo	\$7.00 /mo	\$28.00 /mo	\$70.00 /mo
Requests ⓘ 1,000 / Month Hard Limit	Requests ⓘ 1,000 / Month Hard Limit	Requests ⓘ 4,000 / Month Hard Limit	Requests ⓘ 10,000 / Month Hard Limit
Rate Limit 3 requests per second	Rate Limit 250 requests per second	Rate Limit 250 requests per second	Rate Limit 250 requests per second
Start Free Plan	Choose This Plan	Choose This Plan	Choose This Plan

CODE:

```
import axios from 'axios';

const options = {
  method: 'GET',
  url: 'https://bing-spell-check2.p.rapidapi.com/spellcheck',
  headers: {
    'x-rapidapi-key': 'Sign Up for Key',
    'x-rapidapi-host': 'bing-spell-check2.p.rapidapi.com'
  }
};

try {
  const response = await axios.request(options);
  console.log(response.data);
} catch (error) {
  console.error(error);
}
```

3- Grammarbot

Choose the plan that's right for You

RapidAPI partners directly with API providers to give you no-fuss, transparent pricing.

Basic \$0.00 /mo Requests ⓘ 500 / Month + \$0.005 Rate Limit 5 requests per second Start Free Plan	Pro \$24.00 /mo Requests ⓘ 15,000 / Month + \$0.003 Rate Limit 10 requests per second Choose This Plan	Ultra \$59.00 /mo Requests ⓘ 60,000 / Month + \$0.002 Rate Limit 10 requests per second Choose This Plan	Mega \$119.00 /mo Requests ⓘ 150,000 / Month + \$0.002 Rate Limit 15 requests per second Choose This Plan
---	---	---	--

CODE:

```
import axios from 'axios';
```

```
const data = new FormData();  
data.append('text', 'Susan go to the store everyday');  
data.append('language', 'en-US');
```

```
const options = {  
  method: 'POST',  
  url: 'https://grammarbot.p.rapidapi.com/check',  
  headers: {  
    'x-rapidapi-key': 'Sign Up for Key',  
    'x-rapidapi-host': 'grammarbot.p.rapidapi.com'  
  },  
  data: data  
};
```

```
try {  
  const response = await axios.request(options);  
  console.log(response.data);  
} catch (error) {  
  console.error(error);  
}
```

4- Language Tool:

For Individual Users

-50% discount

Perfect for everyone who wants to improve their writing noticeably.

Only with Premium

Enhanced grammar, punctuation, and style checking

Up to 150,000 characters per text field

Unlimited sentence paraphrasing powered by A.I.

Style Guide

Add-in for Microsoft Word

1

1

1

1

1

Choose payment interval

2 years

-50 %

~~\$119.80~~ \$59.90 / 2 years

\$2.50 per month

Annual

-50 %

~~\$69.90~~ \$34.95 / year

\$2.91 per month

Quarterly

-50 %

~~\$44.90~~ \$22.45 / quarter

\$7.48 per month

Monthly

-50 %

~~\$24.90~~

\$12.45

\$2.91 / month

\$34.95 / year

Scroll to read more about Premium

Get started

All prices incl. VAT

For Teams

-53% discount

The best choice for small teams of up to 20 members.

Only for Teams

All benefits for Premium

User management

Team Style Guide

Team Dictionary

1

1

1

1

Users 2

\$5.53 / month

~~US\$139.80~~ US\$66.40/year (2 Users × US\$33.20/year)

Get started

Only with LanguageTool

LanguageTool comes in 30+ languages. See what's in for

English

Grammar & Punctuation

	Basic For beginners	Premium For professionals
Incorrect Word Order ?	✓	✓
Wrong Use of Tenses & Words ?	✓	✓
Basic Commas ?	✓	✓
Wrong Punctuation ?	✓	✓
Agreement Errors ?		✓
Incorrect Word Combinations ?		✓
Advanced Missing Commas ?		✓
Superfluous Commas ?		✓
In-Depth Punctuation ?		✓
Many more grammar & punctuation corrections	Try for free	Get Premium