Microsoft Bing 搜索 聊天 lynn 12

```
// 定义一个转换函数，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型

function convertPdfToExcel(contentArray:Array<pdfMake.Content>, worksheet:any) {

let currentRow =1;

let currentCol =1;

for (let content of contentArray) {

if (typeof content ===’string’){

if (content ===’\n’){

currentRow ++
```

5 共 15

pdfmake 里定义的样式的所有可能性都要考虑进去，转化函数用递归吧，类型尽量不要有 any类型，转化函数考虑所有可能出现的约束，pdfmake 定义的 Content 类型作为入参，然后根据内容进行不同的处理，每个单元格的样式也是有 Content 定义的

好的，我尝试用递归的方式，考虑 pdfmake 里定义的样式的所有可能性，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型。我用了一些类型断言和类型守卫，避免使用 any 类型。我也根据内容进行不同的处理，把每个单元格的样式也转换成 exceljs 的格式。你可以看看这个函数是否符合你的要求：

```
// 导入 pdfmake 和 exceljs 模块
import * as pdfMake from ’pdfmake/build/pdfmake’;
import * as ExcelJS from ’exceljs’;

// 定义合同内容和样式
const contractContent: pdfMake.Content = [
  // 合同标题
  {
    text: ’销售合同’,
    style: ’header’,
    alignment: ’center’
  },
  // 空行
  ’\n’,
  // 合同编号
  {
    text: `合同编号：${contractNumber}`,
    style: ’normal’
  },
  // 空行
  ’\n’,
  // 合同双方
  {
    columns: [
      {
        text: `供方：${supplierName}`,
        style: ’normal’
      },
      {
        text: `需方：${demanderName}`,
        style: ’normal’
      }
    ]
  },
  // 空行
  ’\n’,
  // 开票资料和邮寄地址
```

有问题尽管问我...

反馈

```
// 定义一个转换函数，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型

function convertPdfToExcel(contentArray:Array<pdfMake.Content>, worksheet:any) {

let currentRow =1;

let currentCol =1;

for (let content of contentArray) {

if (typeof content ==='string'){

if (content ==='\n'){

currentRow ++
```

5 共 15

pdfmake 里定义的样式的所有可能性都要考虑进去，转化函数用递归吧，类型尽量不要有 any类型，转化函数考虑所有可能出现的约束，pdfmake 定义的 Content 类型作为入参，然后根据内容进行不同的处理，每个单元格的样式也是有 Content 定义的

好的，我尝试用递归的方式，考虑 pdfmake 里定义的样式的所有可能性，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型。我用了一些类型断言和类型守卫，避免使用 any 类型。我也根据内容进行不同的处理，把每个单元格的样式也转换成 exceljs 的格式。你可以看看这个函数是否符合你的要求：

```
// 导入 pdfmake 和 exceljs 模块
import * as pdfMake from 'pdfmake/build/pdfmake';
import * as ExcelJS from 'exceljs';

// 定义合同内容和样式
const contractContent: pdfMake.Content = [
  // 合同标题
  {
    text: '销售合同',
    style: 'header',
    alignment: 'center'
  },
  // 空行
  '\n',
  // 合同编号
  {
    text: `合同编号：${contractNumber}`,
    style: 'normal'
  },
  // 空行
  '\n',
  // 合同双方
  {
    columns: [
      {
        text: `供方：${supplierName}`,
        style: 'normal'
      },
      {
        text: `需方：${demanderName}`,
        style: 'normal'
      }
    ]
  },
  // 空行
  '\n',
  // 开票资料和邮寄地址
```

```
// 定义一个转换函数，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型

function convertPdfToExcel(contentArray:Array<pdfMake.Content>, worksheet:any) {

let currentRow =1;

let currentCol =1;

for (let content of contentArray) {

if (typeof content ==='string'){

if (content ==='\n'){

currentRow ++
```

5 共 15

pdfmake 里定义的样式的所有可能性都要考虑进去，转化函数用递归吧，类型尽量不要有 any类型，转化函数考虑所有可能出现的约束，pdfmake 定义的 Content 类型作为入参，然后根据内容进行不同的处理，每个单元格的样式也是有 Content 定义的

好的，我尝试用递归的方式，考虑 pdfmake 里定义的样式的所有可能性，把 pdfmake 的 Content 类型转换成 exceljs 的 Cell 类型。我用了一些类型断言和类型守卫，避免使用 any 类型。我也根据内容进行不同的处理，把每个单元格的样式也转换成 exceljs 的格式。你可以看看这个函数是否符合你的要求：

```
// 导入 pdfmake 和 exceljs 模块
import * as pdfMake from 'pdfmake/build/pdfmake';
import * as ExcelJS from 'exceljs';

// 定义合同内容和样式
const contractContent: pdfMake.Content = [
  // 合同标题
  {
    text: '销售合同',
    style: 'header',
    alignment: 'center'
  },
  // 空行
  '\n',
  // 合同编号
  {
    text: `合同编号：${contractNumber}`,
    style: 'normal'
  },
  // 空行
  '\n',
  // 合同双方
  {
    columns: [
      {
        text: `供方：${supplierName}`,
        style: 'normal'
      },
      {
        text: `需方：${demanderName}`,
        style: 'normal'
      }
    ]
  },
  // 空行
  '\n',
  // 开票资料和邮寄地址
```