帮助文档

Markdown 和快捷键全覆盖

> 💡 Tips：语雀支持全功能 markdown 语法，可以点击文档编辑页右下角小键盘查看全部支持的语法和快捷键。

- 支持导入导出 `markdown` 文件。
- 支持自动识别粘贴的 `markdown` 格式内容转换为富文本。

行内代码

> 💡 Tips：可通过 markdown 语法（`` ` `` + `code` + `` ` `` + 空格）或者快捷键 `ctrl/cmd` + `E` 快速插入行内代码。

在文本中使用 `行内代码`，可以顺畅地显示代码变量名。

代码块

> 💡 Tips：输入 `/代码块` 或点击上方工具栏点击上方工具栏 ，选择「代码块」、插入代码卡片。

代码块同时支持多种颜色主题：

JavaScript

```javascript
export default class QuickSort extends Sort {
  sort(originalArray) {
    const array = [...originalArray];

    if (array.length <= 1) {
      return array;
    }

    // Init left and right arrays.
    const leftArray = [];
    const rightArray = [];

    // Take the first element of array as a pivot.
    const pivotElement = array.shift();
    const centerArray = [pivotElement];

    // Split all array elements between left, center and right arrays.
    while (array.length) {
      const currentElement = array.shift();

      // Call visiting callback.
      this.callbacks.visitingCallback(currentElement);

      if (this.comparator.equal(currentElement, pivotElement)) {
        centerArray.push(currentElement);
      } else if (this.comparator.lessThan(currentElement, pivotElement)) {
        leftArray.push(currentElement);
      } else {
        rightArray.push(currentElement);
      }
    }
    // Sort left and right arrays.
    const leftArraySorted = this.sort(leftArray);
    const rightArraySorted = this.sort(rightArray);

    return leftArraySorted.concat(centerArray, rightArraySorted);
  }
}
```

JavaScript

```javascript
export default class QuickSort extends Sort {
  sort(originalArray) {
    const array = [...originalArray];

    if (array.length <= 1) {
      return array;
    }

    // Init left and right arrays.
    const leftArray = [];
    const rightArray = [];

    // Take the first element of array as a pivot.
    const pivotElement = array.shift();
    const centerArray = [pivotElement];

    // Split all array elements between left, center and right arrays.
    while (array.length) {
      const currentElement = array.shift();

      // Call visiting callback.
      this.callbacks.visitingCallback(currentElement);

      if (this.comparator.equal(currentElement, pivotElement)) {
        centerArray.push(currentElement);
      } else if (this.comparator.lessThan(currentElement, pivotElement)) {
        leftArray.push(currentElement);
      } else {
        rightArray.push(currentElement);
      }
    }
    // Sort left and right arrays.
    const leftArraySorted = this.sort(leftArray);
    const rightArraySorted = this.sort(rightArray);

    return leftArraySorted.concat(centerArray, rightArraySorted);
  }
}
```

```javascript
export default class QuickSort extends Sort {
  sort(originalArray) {
    const array = [...originalArray];

    if (array.length <= 1) {
      return array;
    }

    // Init left and right arrays.
    const leftArray = [];
    const rightArray = [];

    // Take the first element of array as a pivot.
    const pivotElement = array.shift();
    const centerArray = [pivotElement];

    // Split all array elements between left, center and right arrays.
    while (array.length) {
      const currentElement = array.shift();

      // Call visiting callback.
      this.callbacks.visitingCallback(currentElement);

      if (this.comparator.equal(currentElement, pivotElement)) {
        centerArray.push(currentElement);
      } else if (this.comparator.lessThan(currentElement, pivotElement)) {
        leftArray.push(currentElement);
      } else {
        rightArray.push(currentElement);
      }
    }
    // Sort left and right arrays.
    const leftArraySorted = this.sort(leftArray);
    const rightArraySorted = this.sort(rightArray);

    return leftArraySorted.concat(centerArray, rightArraySorted);
  }
}
```

数学公式

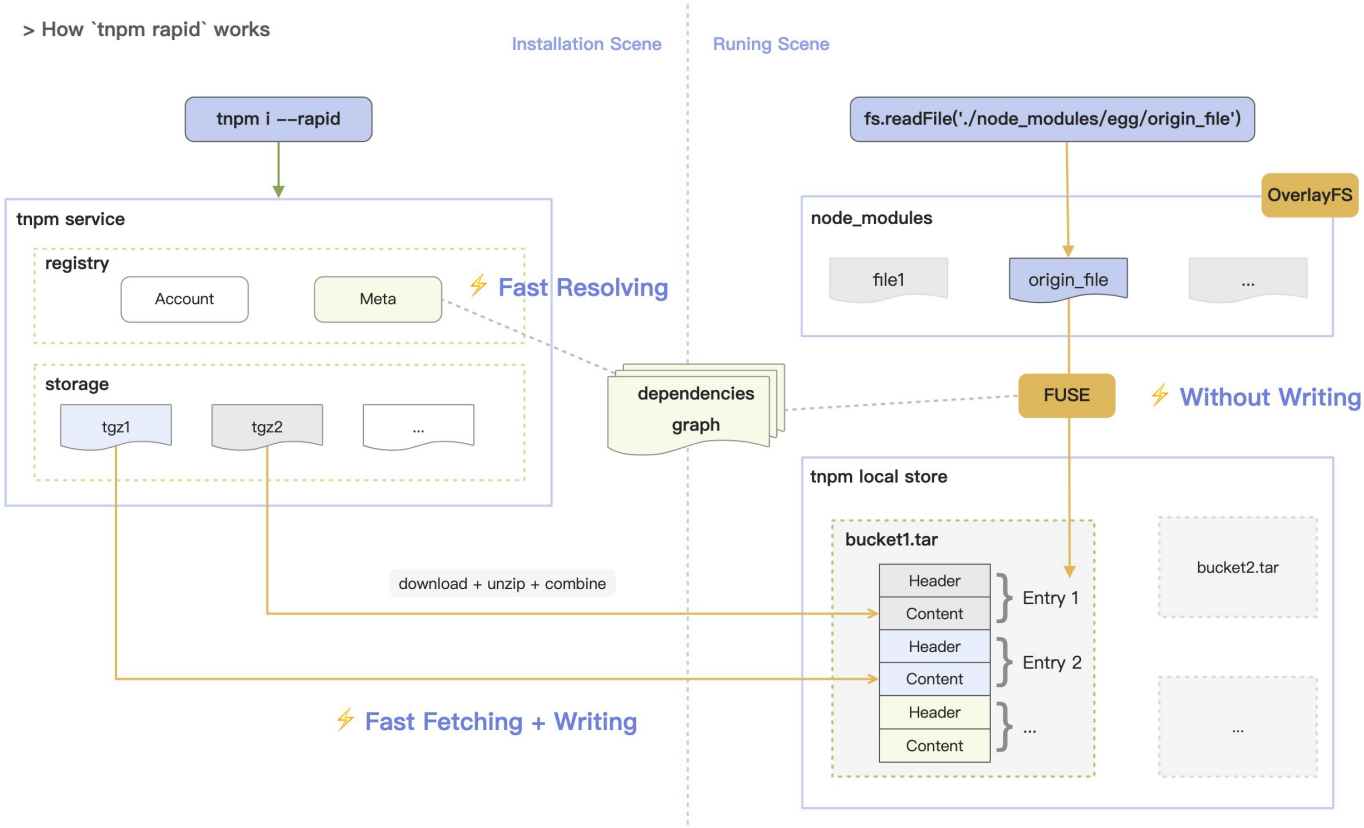
> Tips：输入 `/公式` 或点击上方工具栏点击上方工具栏，选择「公式」、插入公式卡片。

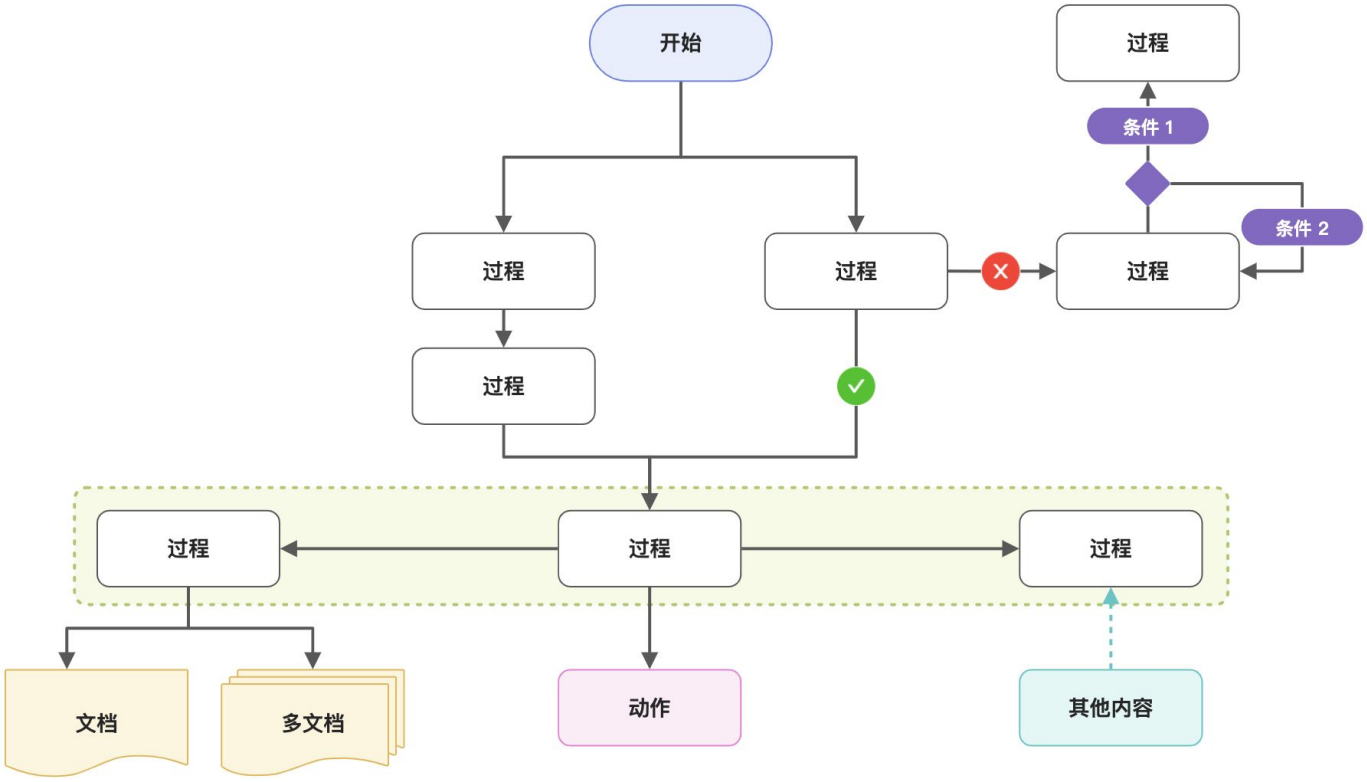
公式支持行内嵌套： $c = \pm\sqrt{a^2 + b^2}$ ，也支持块级嵌入。

$$f(x) = \int_{-\infty}^{\infty} \hat{f}\xi\, e^{2\pi i \xi x}\, d\xi$$

画板

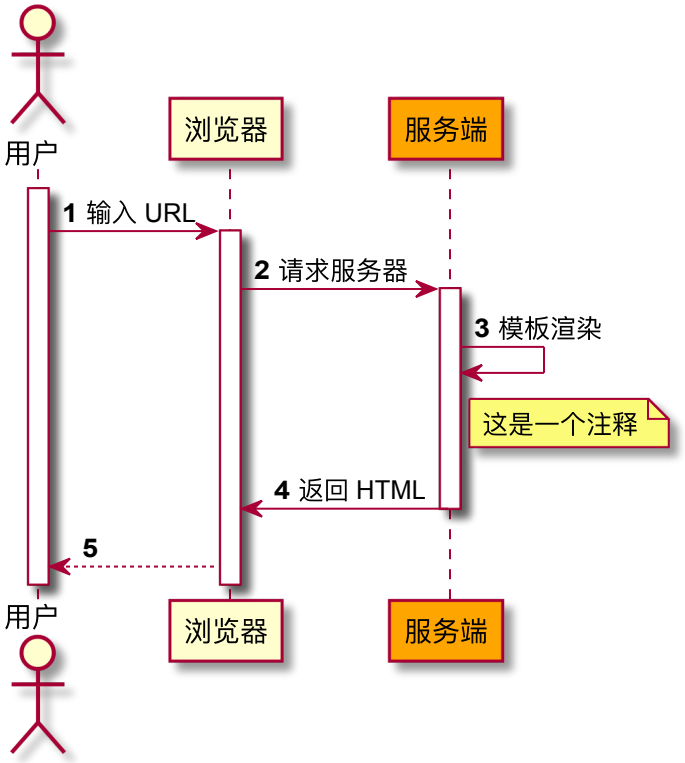
> 💡 Tips：输入 `/画板` 或点击上方工具栏 ➕，选择「画板」、绘制流程图、架构图等各种图形。

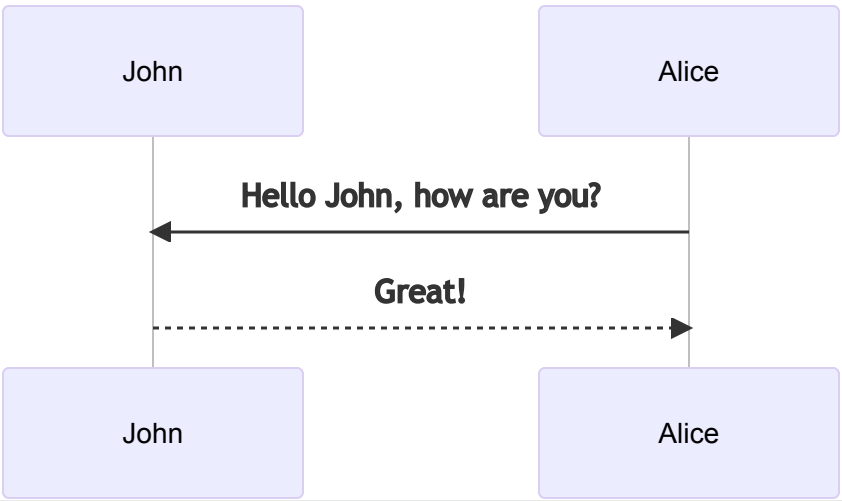




文本绘图

> 💡 Tips：输入 `/文本绘图` 点击上方工具栏 ⊕，选择「文本绘图」、插入文本绘图卡片。
> 支持 plantuml、mermaid 等多种格式，点击 `预览` 可看到图形。具体代码样式见说明文档。


用户
浏览器
服务端
1 输入 URL
2 请求服务器
3 模板渲染
这是一个注释
4 返回 HTML
5
用户
浏览器
服务端


John
Alice
Hello John, how are you?
Great!
John
Alice