UI LIBRARY DESIGN

DICARBENE (WXJ)

1. ABSTRACT

1.1. Button.

1.1.1. *props.*

- content
- icon
- type
- size
- iconPlacement
- disabled
- loading

1.1.2. *button types.*

```
1:  function BINARY-SEARCH(A, n, v)
2:      ▷ Initialize the search range
3:      l ← 1
4:      r ← n
5:
6:      while l ≤ r do
7:          mid ← floor ((l+r)/2)
8:          if A[mid] < v then
9:              l ← m + 1
10:         else if A[mid] > v then
11:             r ← m − 1
12:         else
13:             return m
14:     return null
```

1.1.3. *source code.*

```
<template>
    <div class="gt-ui-button" :class="buttonClass">
        <template v-if="type === 'icon'">
                    <i  v-if="useIconfont"  class="gt-uicomponents-iconfont
icon" :class="icon"></i>
             <img v-else class="icon-url" :src="icon" alt="">
        </template>
        <span>{{ content }}</span>
    </div>
</template>

<script>
const TYPE_ENUM = ['primary', 'secondary', 'normal', 'outline', 'text'];
const SIZE_ENUM = ['tiny','small','medium','large'];
```

```
const ICON_PLACEMENT_ENUM = ['left','right'];
export default {
    name: 'GtUiButton',
    props: {
        content: {
            required: true,
            type: String,
            default: 'default'
        },
        type: {
            required: false,
            validator(value) {
                return Object.values(TYPE_ENUM).includes(value);
            },
            default: 'normal'
        },
        theme: {
            required: false,
            validator(value) {
                return Object.values(THEME_ENUM).includes(value);
            },
            default: 'default'
        },
        size: {
            required: false,
            validator(value) {
                return Object.values(SIZE_ENUM).includes(value);
            },
            type: String,
            default: 'medium'
        },
        iconPlacement: {
            required: false,
            validator(value) {
                return Object.values(SIZE_ENUM).includes(value);
            },
            type: String,
            default: 'left',
        },
        disabled: {
            required: false,
            type: Boolean,
            default: false
        },
        loading: {
            required: false,
            type: Boolean,
            default: false,
        }
    },
    computed: {
        buttonClass() {
            return `gt-ui-button_${this.type} gt-ui-button_${this.type}_$
{this.theme}`
```

```
        },
        useIconfont() {
            return this.icon?.startsWith('icon');
        }
    }
}
</script>
<style lang='less' scoped>
@import "./index.less";
</style>
```

References