

Button组件

一.定义组件所需Props

components/button/src/button.ts

```
import { ExtractPropTypes, PropType } from 'vue'

export type Size = 'tiny' | 'small' | 'medium' | 'large'
export type Type =
  | 'primary'
  | 'success'
  | 'warning'
  | 'danger'
  | 'info'
  | 'default'
  | ''

export const buttonProps = {
  size: String as PropType<Size>, // 按钮大小
  type: {
    type: String as PropType<Type>, // 按钮格式
    validator: (val: string) => {
      return [
```

```
        'default',
        'primary',
        'success',
        'warning',
        'danger',
        'info',
        ''

    ].includes(val)
  },
  default: ''
},
iconPlacement: {
  // 图标位置
  type: String as PropType<'left' | 'right'>,
  default: 'left'
},
nativeType: {
  // 按钮类型
  type: String as PropType<'button' | 'submit'
  | 'reset'>,
  default: 'button'
},
// 状态 加载、禁用、圆角
loading: Boolean,
disabled: Boolean,
round: Boolean
}
```

```
export const buttonEmits = {
  click: (e: MouseEvent) => e instanceof
MouseEvent
}
export type ButtonProps =
ExtractPropTypes<typeof buttonProps>
export type buttonEmits = typeof buttonEmits
```

二.button.vue 结构实现

```
<template>
  <button
    :class="[
      bem.b(),
      bem.m(type),
      bem.m(size),
      bem.is('loading', loading),
      bem.is('disabled', disabled),
      bem.is('round', round)
    ]"
    :disabled="disabled || loading"
    :type="nativeType"
  >
    <slot></slot>
  </button>
</template>
```

```
<script lang="ts" setup>
import { createNamespace } from '@zishui/utils/create'
import { buttonEmits, buttonProps } from
'./button'

defineOptions({
  name: 'ZButton'
})
// 获取属性,及事件
const props = defineProps(buttonProps)
const emit = defineEmits(buttonEmits)

const bem = createNamespace('button')
</script>
```

三.Button入口编写

```
import { withInstall } from '@zishui/utils/withInstall'
import _Button from './src/button.vue'

const Button = withInstall(_Button)

export default Button // 导出Icon组件
export type { buttonProps } from './src/button'

declare module 'vue' {
  export interface GlobalComponents {
    ZButton: typeof Button
  }
}
```

四.Button组件试用

```
<ZButton
  :disabled="true"
  :loading="true"
  :round="true"
  type="primary"
  size="tiny"
>爱你😘</ZButton>
```

五.组件功能实现

1).样式编写

先准备点公共样式，放到 `theme-chalk/src/common/var.scss`

```
$color-primary: #409eff;  
$color-white: #ffffff;  
$color-black: #000000;  
$color-success: #67c23a;  
$color-warning: #e6a23c;  
$color-danger: #f56c6c;  
$color-info: #909399;
```

```
@use 'mixins/mixins' as *;  
@use 'common/var' as *;  
  
@keyframes rotate {  
  from {  
    transform: rotate(0deg);  
  }  
  to {  
    transform: rotate(360deg);  
  }  
}  
  
@include b(button) {
```

// BEM规范

```
align-items: center;
display: inline-flex;
cursor: pointer;
outline: none;
border: #fafafa;
border-radius: 5px;
user-select: none;
min-height: 40px;
line-height: 1;
vertical-align: middle;
padding: 0 20px;
```

```
@include m(tiny) {
    padding: 0 15px;
}
```

```
@include m(medium) {
    padding: 0 20px;
}
```

```
@include m(small) {
    padding: 0 18px;
}
```

```
@include m(large) {
    padding: 0 25px;
}
```

```
@include when(disabled) {
    // 针对不同类型处理
```

```
&,
&:hover,
&:focus {
    cursor: not-allowed;
}
}
@include when(loading) {
    // 针对不同类型处理
    svg.loading {
        animation: rotate 1s linear infinite;
    }
}
@include when(round) {
    border-radius: 20px;
}
@include m(primary) {
    //渲染不同类型的button
    @include button-variant($color-white,
$color-primary, $color-primary);
}
@include m(success) {
    @include button-variant($color-white,
$color-success, $color-success);
}
@include m(warning) {
    @include button-variant($color-white,
$color-warning, $color-warning);
}
```



```

}
@include m(danger) {
    @include button-variant($color-white,
$color-danger, $color-danger);
}
@include m(info) {
    @include button-variant($color-white,
$color-info, $color-info);
}
}

```

提供scss的辅助方法，方便后续使用

```

@mixin button-variant($color, $background-color,
$border-color) {
    color: $color;
    background: $background-color;
    border-color: $border-color;
}

```

2).插槽显示处理

编写个loading组件用于显示loading、loadingIcon.vue

```

<template>
  <ZIcon>
    <svg

```

```
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
viewBox="0 0 1024 1024"
class="loading"
>
<path
    d="M512 1024c-69.1 0-136.2-13.5-199.3-
40.2C251.7 958 197 921 150 874c-47-47-84-101.7-
109.8-162.7C13.5 648.2 0 581.1 0 512c0-19.9
16.1-36 36-36s36 16.1 36 36c0 59.4 11.6 117 34.6
171.3c22.2 52.4 53.9 99.5 94.3 139.9c40.4 40.4
87.5 72.2 139.9 94.3C395 940.4 452.6 952 512
952c59.4 0 117-11.6 171.3-34.6c52.4-22.2 99.5-
53.9 139.9-94.3c40.4-40.4 72.2-87.5 94.3-
139.9C940.4 629 952 571.4 952 512c0-59.4-11.6-
117-34.6-171.3a440.45 440.45 0 0 0-94.3-
139.9a437.71 437.71 0 0 0-139.9-94.3C629 83.6
571.4 72 512 72c-19.9 0-36-16.1-36-36s16.1-36
36-36c69.1 0 136.2 13.5 199.3 40.2C772.3 66 827
103 874 150c47 47 83.9 101.8 109.7 162.7c26.7
63.1 40.2 130.2 40.2 199.3s-13.5 136.2-40.2
199.3C958 772.3 921 827 874 874c-47 47-101.8
83.9-162.7 109.7c-63.1 26.8-130.2 40.3-199.3
40.3z"
    fill="currentColor"
></path>
</svg>
```

```
    </ZIcon>
  </template>

  <script lang="ts" setup>
    import ZIcon from '@zi-shui/components/icon'
  </script>
```

3).实现功能

```
<template>
  <button
    :class="[
      bem.b(),
      bem.m(type),
      bem.m(size),
      bem.is('loading', loading),
      bem.is('disabled', disabled),
      bem.is('round', round)
    ]"
    :disabled="disabled || loading"
    :type="nativeType"
    @click="handleClick"
  >
    <template v-if="iconPlacement == 'left'">
      <LoadingComponent v-if="loading">
    </LoadingComponent>
    <template v-else-if="slots.icon">
```

```
        <Component :is="slots.icon"></Component>
      </template>
    </template>
    <slot></slot>
    <template v-if="iconPlacement == 'right'">
      <LoadingComponent v-if="loading">
</LoadingComponent>
      <template v-else-if="slots.icon">
        <Component :is="slots.icon"></Component>
      </template>
    </template>
  </button>
</template>
```

```
<script lang="ts" setup>
import { createNamespace } from '@zishui/utils/create'
import LoadingComponent from './loadingIcon.vue'
import { useSlots } from 'vue'
import { buttonEmits, buttonProps } from
'./button'

defineOptions({
  name: 'ZButton'
})
// 获取属性,及事件
defineProps(buttonProps)
```

```
const emit = defineEmits(buttonEmits)
const bem = createNamespace('button')
const slots = useSlots()
const handleClick = (e: MouseEvent) => {
  emit('click', e)
}
</script>
```

六.文档编写

按钮 Button

按钮用来触发一些操作。

基础用法

```
<script setup lang="ts">
import { BatteryDead, BatteryFullSharp, Heart }
from '@vicons/ionicons5'
const handleClick = () =>{
  alert('😊')
}
</script>

<ZButton>Default</ZButton>
<ZButton type="primary">Primary</ZButton>
```

```
<ZButton type="success">Success</ZButton>
<ZButton type="warning">Warning</ZButton>
<ZButton type="danger">Danger</ZButton>
<ZButton type="info">Info</ZButton>
```

```vue

```
<template>
 <ZButton>Default</ZButton>
 <ZButton type="primary">Primary</ZButton>
 <ZButton type="success">Success</ZButton>
 <ZButton type="warning">Warning</ZButton>
 <ZButton type="danger">Danger</ZButton>
 <ZButton type="info">Info</ZButton>
</template>
```

```

禁用

```
<ZButton type="primary"
disabled>Primary</ZButton>
```

```html

```
<ZButton type="primary"
disabled>Primary</ZButton>
```

```

图标

```
<ZButton :round="true" size="large"
@click="handleClick">
<template #icon>
<ZIcon size="20">
<Heart />
</ZIcon>
</template>
我爱你
</ZButton>
```

```
<ZButton :round="true" :loading="true">
<template #icon>
<ZIcon size="20">
<Heart />
</ZIcon>
</template>
我爱你
</ZButton>
```

```
<ZButton :round="true" icon-placement="right">
<template #icon>
<ZIcon size="20">
<Heart />
</ZIcon>
</template>
我爱你
```

```
</ZButton>
```

```
` ``html
```

```
<ZButton :round="true" size="large"
```

```
@click="handleClick">
```

```
  <template #icon>
```

```
    <ZIcon size="20">
```

```
      <Heart />
```

```
    </ZIcon>
```

```
  </template>
```

```
  我爱你
```

```
</ZButton>
```

```
<ZButton :round="true" :loading="true">
```

```
  <template #icon>
```

```
    <ZIcon size="20">
```

```
      <Heart />
```

```
    </ZIcon>
```

```
  </template>
```

```
  我爱你
```

```
</ZButton>
```

```
<ZButton :round="true" icon-placement="right">
```

```
  <template #icon>
```

```
    <ZIcon size="20">
```

```
      <Heart />
```

```
    </ZIcon>
```



```
</template>
我爱你
</ZButton>
` ``
```

事件

```
<ZButton @click="handleClick">
点我啊
</ZButton>
```

```
` ``html
<ZButton @click="handleClick"> 点我啊 </ZButton>
` ``
```

API

Button Props

名称	类型	默认值	说明
-----	-----		
-----		-----	-----

size	tiny \	small \	medium \	
large		medium		按钮大小
type	primary \	success \	warning \	
\danger \	info\	default	default	按钮颜色
icon-placement	left \	right		
		left		icon 位置
nativeType	button \	reset \	submit	
		button		按钮类型
disabled	boolean			
		false		按钮是否禁用
loading	boolean			
		false		按钮是否显示
加载状态				
round	boolean			
		false		按钮是否显示
圆角				

Button Slots

名称	默认值	说明	
-----	-----	-----	
default	undefined	按钮的内容	

icon	undefined	按钮的图标	
------	-----------	-------	--