Drawer 抽屉

有些时候，`Dialog` 组件并不满足我们的需求，比如你的表单很长，亦或是你需要临时展示一些文档，`Drawer` 拥有和 `Dialog` 几乎相同的 API，在 UI 上带来不一样的体验。

基本用法

呼出一个临时的侧边栏，可以从多个方向呼出

:::demo 需要设置 `visible` 属性，它的**类型**是 `boolean`，当为 **true** 时显示 Drawer。Drawer 分为两个部分：`title` 和 `body`， `title` 需要具名为 **title** 的 `slot`，也可以通过 `title` 属性来定义，默认值为空。需要注意的是，Drawer 默认是从右往左打开，当然可以设置对应的 `direction`，详细请参考 `direction` 用法 最后，本例还展示了 `before-close` 的用法

```
<el-radio-group v-model="direction">
  <el-radio label="ltr">从左往右开</el-radio>
  <el-radio label="rtl">从右往左开</el-radio>
  <el-radio label="ttb">从上往下开</el-radio>
  <el-radio label="btt">从下往上开</el-radio>
</el-radio-group>

<el-button @click="drawer = true" type="primary" style="margin-left: 16px;">
  点我打开
</el-button>

<el-drawer
  title="我是标题"
  :visible.sync="drawer"
  :direction="direction"
  :before-close="handleClose">
  <span>我来啦!</span>
</el-drawer>

<script>
  export default {
    data() {
      return {
        drawer: false,
        direction: 'rtl',
      };
    },
    methods: {
      handleClose(done) {
        this.$confirm('确认关闭? ')
          .then(_ => {
            done();
          })
          .catch(_ => {});
      }
    }
  };
</script>
```

:::

不添加 Title

当你不需要标题到时候, 你还可以去掉标题

:::demo 当遇到不需要 title 的场景时, 可以通过 `withHeader` 这个属性来关闭掉 title 的显示, 这样可以留出更大的空间给到用户, 为了用户的可访问性, 请务必设定 `title` 的值

```
<el-button @click="drawer = true" type="primary" style="margin-left: 16px;">
  点我打开
</el-button>

<el-drawer
  title="我是标题"
  :visible.sync="drawer"
  :with-header="false">
  <span>我来啦!</span>
</el-drawer>

<script>
  export default {
    data() {
      return {
        drawer: false,
      };
    }
  };
</script>
```

:::

自定义内容

和 `Dialog` 组件一样, `Drawer` 同样可以在其内部嵌套各种丰富的操作

:::demo

```
<el-button type="text" @click="table = true">打开嵌套表格的 Drawer</el-button>
<el-button type="text" @click="dialog = true">打开嵌套 Form 的 Drawer</el-button>
<el-drawer
  title="我嵌套了表格!"
  :visible.sync="table"
  direction="rtl"
  size="50%">
   <el-table :data="gridData">
      <el-table-column property="date" label="日期" width="150"></el-table-column>
      <el-table-column property="name" label="姓名" width="200"></el-table-column>
      <el-table-column property="address" label="地址"></el-table-column>
    </el-table>
</el-drawer>
```

```
<el-drawer
  title="我嵌套了 Form !"
  :before-close="handleClose"
  :visible.sync="dialog"
  direction="ltr"
  custom-class="demo-drawer"
  ref="drawer"
  >
  <div class="demo-drawer__content">
    <el-form :model="form">
      <el-form-item label="活动名称" :label-width="formLabelWidth">
        <el-input v-model="form.name" autocomplete="off"></el-input>
      </el-form-item>
      <el-form-item label="活动区域" :label-width="formLabelWidth">
        <el-select v-model="form.region" placeholder="请选择活动区域">
          <el-option label="区域一" value="shanghai"></el-option>
          <el-option label="区域二" value="beijing"></el-option>
        </el-select>
      </el-form-item>
    </el-form>
    <div class="demo-drawer__footer">
      <el-button @click="cancelForm">取 消</el-button>
      <el-button type="primary" @click="$refs.drawer.closeDrawer()"
:loading="loading">{{ loading ? '提交中 ...' : '确 定' }}</el-button>
    </div>
  </div>
</el-drawer>

<script>
export default {
  data() {
    return {
      table: false,
      dialog: false,
      loading: false,
      gridData: [{
        date: '2016-05-02',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-04',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-01',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
      }, {
        date: '2016-05-03',
        name: '王小虎',
        address: '上海市普陀区金沙江路 1518 弄'
```

```
      }],
      form: {
        name: '',
        region: '',
        date1: '',
        date2: '',
        delivery: false,
        type: [],
        resource: '',
        desc: ''
      },
      formLabelWidth: '80px',
      timer: null,
    };
  },
  methods: {
    handleClose(done) {
      if (this.loading) {
        return;
      }
      this.$confirm('确定要提交表单吗？')
        .then(_ => {
          this.loading = true;
          this.timer = setTimeout(() => {
            done();
            // 动画关闭需要一定的时间
            setTimeout(() => {
              this.loading = false;
            }, 400);
          }, 2000);
        })
        .catch(_ => {});
    },
    cancelForm() {
      this.loading = false;
      this.dialog = false;
      clearTimeout(this.timer);
    }
  }
}
</script>
```

:::

多层嵌套

`Drawer` 组件也拥有多层嵌套的方法

:::demo 同样, 如果你需要嵌套多层 `Drawer` 请一定要设置 `append-to-body` 属性为 **true**

```
<el-button @click="drawer = true" type="primary" style="margin-left: 16px;">
```

```
  点我打开
</el-button>

<el-drawer
  title="我是外面的 Drawer"
  :visible.sync="drawer"
  size="50%">
  <div>
   <el-button @click="innerDrawer = true">打开里面的!</el-button>
   <el-drawer
     title="我是里面的"
     :append-to-body="true"
     :before-close="handleClose"
     :visible.sync="innerDrawer">
     <p>_(:з」∠)_</p>
   </el-drawer>
  </div>
</el-drawer>

<script>
  export default {
    data() {
      return {
        drawer: false,
        innerDrawer: false,
      };
    },
    methods: {
      handleClose(done) {
        this.$confirm('还有未保存的工作哦确定关闭吗？')
          .then(_ => {
            done();
          })
          .catch(_ => {});
      }
    }
  };
</script>
```

:::

:::tip

Drawer 的内容是懒渲染的，即在第一次被打开之前，传入的默认 slot 不会被渲染到 DOM 上。因此，如果需要执行 DOM 操作，或通过 `ref` 获取相应组件，请在 `open` 事件回调中进行。

:::

:::tip

Drawer 提供一个 `destroyOnClose` API, 用来在关闭 Drawer 时销毁子组件内容, 例如清理表单内的状态, 在必要时可以将该属性设置为 **true** 用来保证初始状态的一致性

:::

:::tip

如果 `visible` 属性绑定的变量位于 Vuex 的 store 内，那么 `.sync` 不会正常工作。此时需要去除 `.sync` 修饰符，同时监听 Drawer 的 `open` 和 `close` 事件，在事件回调中执行 Vuex 中对应的 mutation 更新 `visible` 属性绑定的变量的值。

:::

Drawer Attributes

参数	说明	类型	可选值	默认值
append-to-body	Drawer 自身是否插入至 body 元素上。嵌套的 Drawer 必须指定该属性并赋值为 true	boolean	—	false
before-close	关闭前的回调，会暂停 Drawer 的关闭	function(done)，done 用于关闭 Drawer	—	—
close-on-press-escape	是否可以通过按下 ESC 关闭 Drawer	boolean	—	true
custom-class	Drawer 的自定义类名	string	—	—
destroy-on-close	控制是否在关闭 Drawer 之后将子元素全部销毁	boolean	-	false
modal	是否需要遮罩层	boolean	—	true
modal-append-to-body	遮罩层是否插入至 body 元素上，若为 false，则遮罩层会插入至 Drawer 的父元素上	boolean	—	true
direction	Drawer 打开的方向	Direction	rtl / ltr / ttb / btt	rtl
show-close	是否显示关闭按钮	boolean	—	true
size	Drawer 窗体的大小, 当使用 `number` 类型时, 以像素为单位, 当使用 `string` 类型时, 请传入 'x%', 否则便会以 `number` 类型解释	number / string	-	'30%'
title	Drawer 的标题，也可通过具名 slot （见下表）传入	string	—	—
visible	是否显示 Drawer，支持 .sync 修饰符	boolean	—	false
wrapperClosable	点击遮罩层是否可以关闭 Drawer	boolean	-	true
withHeader	控制是否显示 header 栏, 默认为 true, 当此项为 false 时, title attribute 和 title slot 均不生效	boolean	-	true

Drawer Slot

name	说明
—	Drawer 的内容
title	Drawer 标题区的内容

Drawer Methods

name	说明
closeDrawer	用于关闭 Drawer, 该方法会调用传入的 `before-close` 方法

Drawer Events

事件名称	说明	回调参数
open	Drawer 打开的回调	—
opened	Drawer 打开动画结束时的回调	—
close	Drawer 关闭的回调	—
closed	Drawer 关闭动画结束时的回调	—