[1.根类 1](#_Toc20682)

[2.表单的容器 1](#_Toc7755)

[3.数据交互和加载 1](#_Toc23468)

[4.字段的控制 2](#_Toc30222)

[5.其中夹杂布局的类 2](#_Toc8901)

[Validations 内置校验 4](#_Toc3767)

[Ext.data.validations 4](#_Toc910)

[Ext.form.action.Submit 5](#_Toc3639)

## 1.根类

Ext.form.Basic

提供了,表单组件,字段管理,数据验证,表单提交,数据加载的功能

## 2.表单的容器

Ext.form.Panel

容器自动关联 Ext.form.Basic 的实例对象更方便的进行字段的配置

重要属性

defaultType:"" 设置默认子项 的xtype

## 3.数据交互和加载

Ext.form.action.Action

Ext.form.action.Submit ajax方式提交

Ext.form.action.StandardSubmit 原始表单提交方法

Ext.form.action.DirectLoad

Ext.form.action.DirectSubmit 指令式的模式

## 4.字段的控制

Ext.form.field.Base 是跟类

混入了类 [Ext.form.field.Field]得到表单值的处理功能

混入了类[Ext.form.Labelable]得到表单标签错误信息提示的功能

Ext.form.field.Text   文本框方式的

Ext.form.field.Trigger 触发器

Ext.form.field.Time  带有时间下拉框 和自动验证的input表单。

Ext.form.field.Date 带有日期选择器下拉框并会自动进行

日期验证的日期输入表单

Ext.form.field.Number   数值型的文本表单，对非数组值行的

按键进行自动过滤，并且限定一系列 有效范围。

Ext.form.field.File 文件上传的

Ext.form.field.ComboBox 选择框

Ext.form.field.Checkbox 选择框方式的

Ext.form.field.Radio 单选框

Ext.form.field.Hidden 特殊的-隐藏字段  提交表单时传递到后台。

Ext.form.field.HtmlEditor 特殊的-文本编辑框

## 5.其中夹杂布局的类

Ext.form.FieldContainer    文本域容器

Ext.form.CheckboxGroup  表单项容器。

Ext.form.RadioGroup    一个[field container](http://extjs-doc-cn.github.io/ext4api/#!/api/Ext.form.FieldContainer)其中有一个专门

把[Ext.form.field.Radio](http://extjs-doc-cn.github.io/ext4api/#!/api/Ext.form.field.Radio) 控件分类按列分布的布局安排

Ext.form.Label

创建一个独立的 <label /> 元素,此元素可以加入到 form

 之中,也可以通过 [forId](http://extjs-doc-cn.github.io/ext4api/#!/api/Ext.form.Label-cfg-forId) 与该form中的表单域 field 关联

Ext.form.Labelable

一个混合类，允许组件被配置且装饰有标签和错误消息，

作为表单字段的通用组件。

Ext.form.FieldSet

控件组,输入域组合容器. 包装一组输入域的容器,

Ext.form.FieldContainer  文本域容器

Ext.form.Panel

重要的配置项

title:'',             标题头

bodyStyle:'',      自定义到css 样式

frame : ,            以什么方式提交提画面

height: ,            高

width :, 宽

renderTo:'',    指定某个ｉｄ　，这元素将被渲染。

defaultType:'',   当前容器中创建子组件时使用的默认 [xtype](http://extjs-doc-cn.github.io/ext4api/#!/api/Ext.Component)

defaults:{}

由于混入了Ext.form.Labelable

可以配置

labelSeparator 字段名字和值的分割符号

labelWidth 标签宽度

重要的方法

Ext.form.field.Text  文本框(xtype: textfield)

重要的配置项

width : 150,

allowBlank: false, //不能是空

labelAlign :'left',

msgTarget:'side'//在字段的右面展示数据

重要的方法

Ext.form.field.ComboBox

控件支持自动完成、远程加载、和许多其他特性。

## Validations 内置校验

### Ext.data.validations

本单例包含一个验证函数集合, 用以验证任何类型的数据。通常在 [Model](http://extjs-doc-cn.github.io/ext4api/" \l "!/api/Ext.data.Model)中使 用它们, 它们是自动生成和执行的。

可以通过msgTarget改变错误信息的显示位置，通过invalidText改变错误

信息的内容，每个字段都有自己的invalidText实现方式，错误信息中有

许多可替换的标记

例如，在Date Field的invalidText中，任何’{0}’ 都会被替换成这个字段的值，’ {1}’会被替换成这个字段的format，下面的代码展示了如何使用这个特

性自定义错误信息

自定义校验：

用Text Field的regex配置应用一个校验规则，和使用maskRe配置限制可输入的字符，这有一个使用TextField校验输入时间的例子

  {

     fieldLabel: 'Last Login Time',

     name: 'loginTime',

     regex: /^([1-9]|1[0-9]):([0-5][0-9])(\s[a|p]m)$/i,

     maskRe: /[\d\s:amp]/i,

     invalidText: 'Not a valid time.  Must be in the format "12:34 PM".'}

### Ext.form.action.Submit

这类用来处理用Form 提交数据，并可以处理应答的返回值。

这个类的实例只在Form 提交的时候创建。

buttons: [ {

text: 'Submit',

handler: function() { var form = this.up('form').getForm();

if (form.isValid()) {

form.submit({ success: function(form, action) {

Ext.Msg.alert('Success', action.result.msg); },

failure: function(form, action) {

Ext.Msg.alert('Failed', action.result.msg); } });

} else {

Ext.Msg.alert('Invalid Data', 'Please correct form errors.')

} }

} ]

  1.首先找到对这个类的的引用.

2. 提交之前调用了isValid方法确保每个表单字段都已经填写正确

3.最后调用submit方法，并传递了两个回调函数success和failure，在这两个回调函数的参数中，action.result可以引用到服务器端返回JSON的解析后的对象

数据可以通过loadRecord方法直接从Model加载进入Form Panel：

Ext.ModelMgr.getModel('User').load(1, {

success: function(user) {

// 当用户加载成功，加载数据到表单 userForm.loadRecord(user); } });

代替submit方法，可以使用BasicForm的updateRecord方法更新form绑定的model，然后用Model的save方法保存数据

buttons: [ { text: 'Submit', handler: function() {

var form = this.up('form').getForm(),

//获取到这个from record = form.getRecord();

// 得到底层的模型实例 if (form.isValid()) {

//提交前确保表单含有有效数据 form.updateRecord(record);

// 更新的记录表单数据 record.save({

// 将记录保存到服务器 success: function(user) { Ext.Msg.alert('Success', 'User saved successfully.') },

failure: function(user) { Ext.Msg.alert('Failure', 'Failed to save user.') } }); } else {

// 显示错误警报,如果数据是无效的 Ext.Msg.alert('Invalid Data', 'Please correct form errors.') } }

} ]