

link: null  
title: 珠峰架构师成长计划  
description: 我们在form组件中直接将当前实例暴露出去  
keywords: null  
author: null  
date: null  
publisher: 珠峰架构师成长计划  
stats: paragraph=46 sentences=51, words=452

实现表单组件 #

- 组件结构目录

```
6#x251C;6#x2500;6#x2500; App.vue
6#x251C;6#x2500;6#x2500; components
6#x2502;      6#x251C;6#x2500;6#x2500; Form.vue
6#x2502;      6#x251C;6#x2500;6#x2500; FormItem.vue
6#x2502;      6#x2514;6#x2500;6#x2500; Input.vue
6#x2514;6#x2500;6#x2500; package.json
```

表单组件的使用 #

- Form 提供组件的数据
- FormItem 实现表单校验功能
- Input 输入内容时让FormItem发生校验

使用组件的方式 #

```
<template>
  <el-form :model="form" :rules="rules">
    <el-form-item label="6#x7528;6#x6237;6#x540D;" prop="username">
      <el-input v-model="form.username"></el-input>
    </el-form-item>
    <el-form-item label="6#x5BC6;6#x7801;" prop="password">
      <el-input v-model="form.password"></el-input>
    </el-form-item>
    <el-form-item>
    <el-form-item>
      <button>6#x786E;6#x8BA4;6#x63D0;6#x4EA4;</button>
    </el-form-item>
  </el-form-item></el-form>
</template>
<script>
import Form from './components/Form';
import FormItem from './components/FormItem';
import Input from './components/Input.vue';
export default {
  components:{
    'el-form':Form,
    'el-form-item':FormItem,
    'el-input':Input
  },
  data() {
    return {
      form:{
        username:'',
        password:''
      },
      rules:{
        username:[
          {required:true,message:'请输入用户名'}
        ],
        password:[
          {required:true,message:'请输入密码'}
        ]
      }
    }
  }
}
</script>
```

Form组件的基本结构 #

```
<template>
  <form onsubmit="return false">
    <slot></slot>
  </form>
</template>
<script>
export default {
  props:{
    model:{
      type:Object
    },
    rules:{
      type:Object
    }
  }
}
</script>
```

FormItem组件结构 #

```
<template>
  <div>
    <label v-if="label">{{label}}</label>
    <slot></slot>
    <div>{{x6821;{{x9A8C;{{x6587;{{x5B57;</div>
  </div>
</template>
<script>
  export default {
    props: {
      label: String,
      prop: String
    }
  }
</script>
```

## Input组件结构 #

```
<template>
  <input type="text" :value="inputValue">
</template>
<script>
export default {
  props: {
    value: String
  },
  data () {
    return {inputValue: this.value}
  }
}
</script>
```

## 实现组件间的数据传递 #

- 为了在formItem中可以拿到form组件中的数据

```
provide () {
  return {
    form: this
  }
},
```

我们在form组件中直接将当前实例暴露出去

- 为了能够实现input组件和formItem间的通信

```
import Vue from 'vue';
Vue.prototype.$bus = new Vue();
export default {
  props: {
    label: String,
    prop: String
  },
  mounted () {
    this.$bus.$on('input', (value) => {
      console.log(value)
    });
  }
}
```

我们在formItem组件中，通过eventsBus订阅input事件

- 在input组件中监听输入事件进行数据的发射

```
<input type="text" :value="inputValue" @input="handleInput">
handleInput (e) {
  // {{x66F4;{{x65B0;{{x6570;{{x636E;
  this.inputValue = e.target.value;
  this.$bus.$emit('input', {
    id: this.$parent.$parent._uid, // {{x4E3A;{{x4E86;{{x6807;{{x8BC6;{{x662F;{{x54EA;{{x4E2A;{{x8F93;{{x5165;{{x6846;
    value: this.inputValue
  }); // {{x53D1;{{x5C04;{{x8F93;{{x5165;{{x4E8B;{{x4EF6;
}
```

- 在formItem中进行数据校验

```

<template>
  <div>
    <label v-if="label">{{label}}</label>
    <slot></slot>
    <!-- 有错误 显示错误提示信息 -->
    <div v-if="validateStatus === 'error'">
      {{validateContent}}
    </div>
  </div>
</template>
<script>
import { Vue, mixins } from 'vue';
const { props, mixins } = mixins;
export default {
  props: {
    label: String,
    prop: String
  },
  data() {
    return {
      // 注入父级的实例
      status: 'error'
    }
  },
  methods: {
    validate(value) {
      let rules = this.$rules[this.prop]; // 获取当前字段的规则
      let rulesList = rules || [];
      // 如果必填 并且有值，那就去跑
      if (rulesList.length > 0) {
        this.$validate(value);
        this.$validate(value);
      }
    }
  }
};
</script>

```

到目前为止基本的校验功能已经实现啦

### 当点击按钮时校验当前表单是否验证成功 #

```

<el-form :model="form" :rules="rules" ref="form">
  <button @click="validate">validate</button>
  validate() { // form.validate()
    this.$refs.form.validate((valid) => {
      if (valid) {
        alert('validate success');
      } else {
        alert('validate failed');
      }
    });
  }

  // form.validate()
  validate(cb) {
    cb(this.$children.every(child => child.validateStatus !== 'error'))
  }
</el-form>

```