

WEBAPP勉強会 第5回目 解説

- WEBAPP勉強会 第5回目 解説
 - 新規登録ボタンを配置
 - 新規登録ダイアログを開く処理
 - 新規登録本体の実装
 - isUpdate変数定義
 - handleEditにisUpdate設定を追加
 - handleNewにisUpdate設定とidのダミーを追加
 - doExecuteに処理分岐を追加
 - 新規登録処理(registerItem)のダミーを実装します
 - computedで表示を制御

新規登録ボタンを配置

参考URL : <https://element.eleme.io/#/en-US/component/button#icon-button>

上記を参考にして、新規登録ダイアログを表示するボタンを絞り込みの右側に配置します。

```
<!-- 検索入力 -->
<el-input v-model="search" size="small" placeholder="検索文字列を入力">
  <template slot="prepend">内容で絞り込む</template>
</el-input>
```

検索入力の<el-input> ~ </el-input>の部分に新規入力のボタンを追加します。

普通に追加するだけだと、レイアウトが崩れてしまうので、el-rowとel-colを使ってレイアウトを整えます。

以下、簡単に構造を書きます。

```
<el-row>
  <el-col :span="20">
    <el-input> ~ </el-input>    ... 元の検索入力の部分
  </el-col>
  <el-col :span="4">
    <el-button> ~ </el-button> ... 新規登録ボタン
  </el-col>
</el-row>
```

上記の構造になるように検索入力部分を以下のように変更します。

```
<!-- 検索入力 -->
<el-row>
  <el-col :span="20">
    <el-input v-model="search" size="small" placeholder="検索文字列を入力">
```

```
        <template slot="prepend">内容で絞り込む</template>
      </el-input>
    </el-col>
    <el-col :span="4" style="text-align: center">
      <el-button type="success" icon="el-icon-circle-plus-outline" size="small"
@click="handleNew()">新規登録</el-button>
    </el-col>
  </el-row>
```

新規登録ダイアログを開く処理

新規登録ボタンクリックでダイアログを表示する処理を実装します。

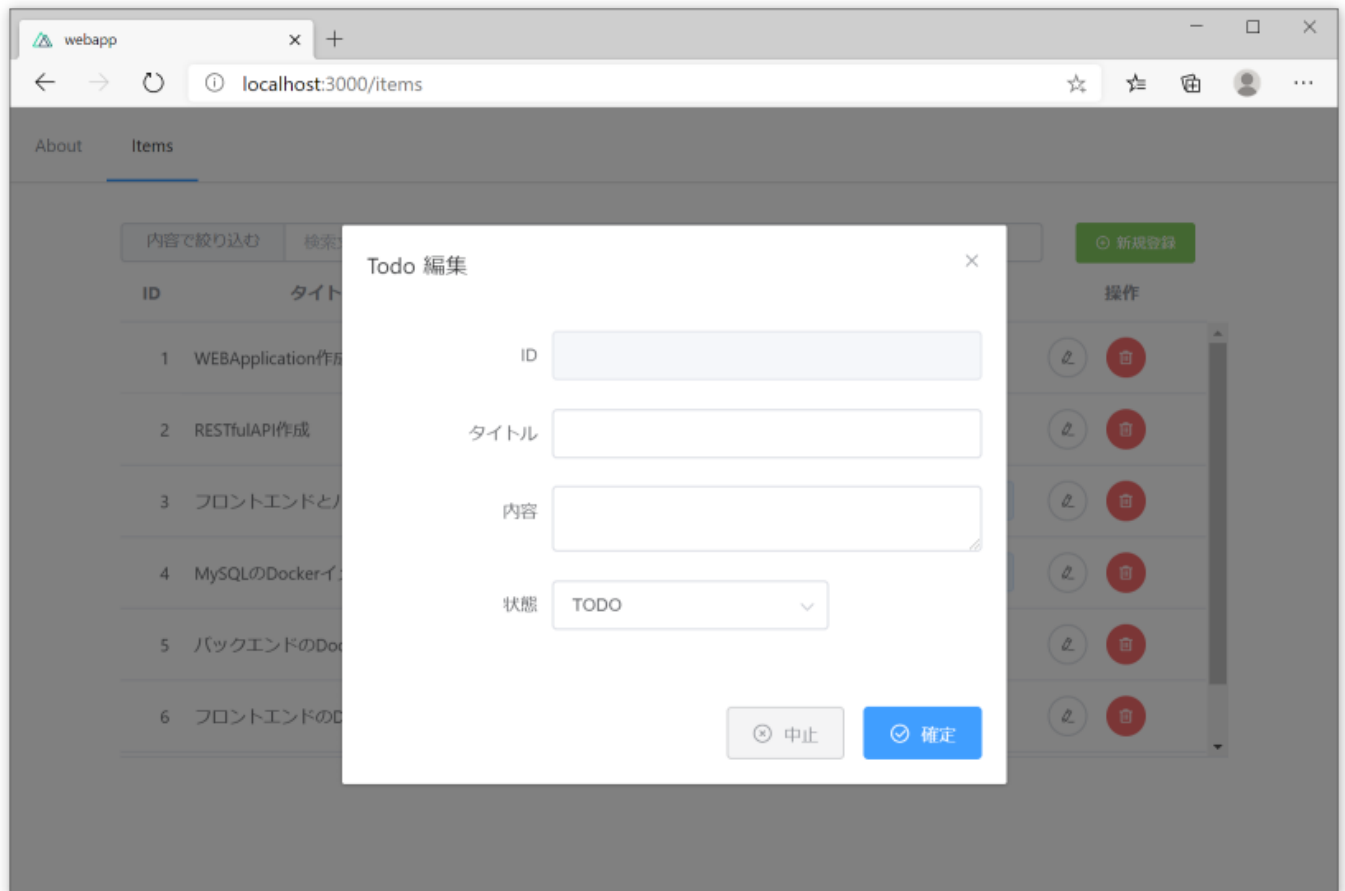
編集で作成したダイアログを利用します。

`methods: {}` 内の `handleDelete()` の下に `handleNew()` を追加します。

その際にはformを初期化します。

```
handleNew() {
  this.form.id = null
  this.form.title = ''
  this.form.content = ''
  this.form.status = 'TODO'
  this.dialogFormVisible = true
},
```

これで「新規作成」ボタンをクリックしたら、新規登録用のダイアログが表示されるようになります。



新規登録本体の実装

ダイアログで入力された内容を登録する処理を実装します。

新規登録のダイアログは、編集と共用となりますので、「確定」ボタンをクリックした際の動作を、編集と新規登録で分ける必要があります。この機能を実現するために、`isUpdate` という変数を導入します。

編集時は`isUpdate`を`true`とし、新規登録時は`isUpdate`を`false`とします。これにより、`doExecute()`内で処理を分岐できるようします。

1. 編集ボタンクリック時
 1. `handleEdit`内で`isUpdate`を`true`にする
 2. ダイアログを表示する
2. 新規作成ボタンクリック時
 1. `handleNew`内で`isUpdate`を`false`にする
 2. ダイアログを表示する
3. ダイアログの「確定」ボタンクリック
4. `doExecute()`がコールされる
 1. `isUpdate`が`true`の場合
 1. `updateItem`をコール
 2. `isUpdate`が`false`の場合
 1. `insetItem`をコール

`isUpdate`変数定義

`data() {...}` 内の`dialogFormVisible`の下に`isUpdate`を追加します。

また、REST APIを繋ぐまでダミーとしてクライアント側処理を行うため`nextId`も追加します。

```
    :
    formLabelWidth: '150px',
    dialogFormVisible: false,
    isUpdate: true,
    nextId: 8,
    rowNumber: '',
    :
```

`handleEdit`に`isUpdate`設定を追加

```
handleEdit(index, row) {
  this.isUpdate = true
  this.rowNumber = index
  this.fetchKey(row.id)
  this.form = { ...this.item }
  this.dialogFormVisible = true
},
```

`handleNew`に`isUpdate`設定と`id`のダミーを追加

```
handleNew() {
  this.isUpdate = false
  this.form.id = null
  this.form.title = ''
  this.form.content = ''
  this.form.status = 'TODO'
  this.dialogFormVisible = true
  // 以下はREST APIと接続するまでのダミー処理
  // idはREST APIでは自動採番の予定
  this.form.id = this.nextId
},
```

`doExecute`に処理分岐を追加

```
doExecute() {
  this.dialogFormVisible = false
  // 編集モード(isUpdate=true)の場合は更新処理
  if (this.isUpdate) {
    this.updateItem(this.form)
  } else {
    this.registerItem(this.form)
  }
}
```

```
  }  
},
```

新規登録処理(`registerItem`)のダミーを実装します

```
registerItem(param) {  
  // ToDo: REST API登録処理呼び出し  
  // Dummy insert  
  const item = { ...param }  
  this.tableData.push(item)  
  this.nextId++  
  
  const target = `${param.id}:${param.title}`  
  this.$message({  
    type: 'success',  
    message: `${target} : 登録が成功しました。`,  
    showClose: true,  
    duration: 5000  
  })  
},
```

本来はREST APIにparamを渡して登録処理を行う部分になります。

idはREST APIでは自動採番になるので設定不要ですが、今はクライアントで設定必要なのでnextIdで管理します。

いかがでしょうか？登録できたでしょうか？

computedで表示を制御

最後に、新規登録と編集により、ダイアログタイトルを変える方法を紹介します。

参考URL : <https://jp.vuejs.org/v2/guide/computed.html> 算出プロパティ

```
<el-dialog title="Todo 編集" :visible.sync="dialogFormVisible">
```

このダイアログタイトルを算出プロパティ`dialogTitle`に変更します。

```
<el-dialog :title="dialogTitle" :visible.sync="dialogFormVisible">
```

`data()` `{..}`の下に`computed: {..}`のブロックを追加します。

```
data() {  
  
  },
```

```
computed: {  
  dialogTitle() {  
    return this.isUpdate ? 'ToDo 編集' : 'ToDo 新規登録'  
  }  
},
```

`isUpdate`が`true`の場合は編集、それ以外の場合は新規登録と表示します。

