springMCV 同时接收 bean 及 List

> 存在这样的业务需要，在一个表单详情页，比如说我们最常见的订单确认页，一般我们会在这个页面提交表单数据，以及商品的列表数据，那么如何做，才能是提交的过程最简单，最大程度上减少数据格式的转化。

以表单提交的方式处理

前端代码

```js
            let detail = new URLSearchParams();
            detail.append('factid', '01');
            detail.append('factname', '测试1');
            detail.append('factshort', '1');
            detail.append('linkman', '测试联系人');
            let list = new URLSearchParams();
            detail.append('faclist[0].factid', '02');
            detail.append('faclist[0].factname', '测试1');
            detail.append('faclist[0].factshort', '1');
            detail.append('faclist[0].linkman', '测试联系人');

this.$axios.post('/pestiot.web/smo/savedetailandlist.do',detail)
                .then(function (res) {
                    console.dir(res);
                })
```

后端代码

```java
@Controller
@RequestMapping("smo")
public class SmoparkController extends BaseController {

    public static class BasFactoryList implements Serializable {
        private List<BasFactory> faclist;

        public List<BasFactory> getFaclist() {
            return faclist;
        }
        public void setFaclist(List<BasFactory> faclist) {
            this.faclist = faclist;
        }
```

```
        public BasFactoryList() {
            super();
        }
    }

    @RequestMapping(value = "/savedetailandlist", method = RequestMethod.POST)
    @ResponseBody
    public ReturnValue SaveDetailAndList (BasFactory detail, BasFactoryList list) {
        ReturnValue rtv = new ReturnValue();
        System.out.println(detail.getFactid());
        System.out.println(list.getFaclist().get(0).getFactid());
        return rtv;
    };

}
```

以 pojo 的方式处理

前端代码

```js
            let detail = new URLSearchParams();
            detail.append('smo.factid', '01');
            detail.append('smo.factname', '测试 1');
            detail.append('smo.factshort', '1');
            detail.append('smo.linkman', '测试联系人');
            let list = new URLSearchParams();
            detail.append('smolist.faclist[0].factid', '02');
            detail.append('smolist.faclist[0].factname', '测试 1');
            detail.append('smolist.faclist[0].factshort', '1');
            detail.append('smolist.faclist[0].linkman', '测试联系人');

this.$axios.post('/pestiot.web/smo/savedetailandlist.do',detail)
                .then(function (res) {
                    console.dir(res);
                })
```

后端代码

```java
@Controller
@RequestMapping("smo")
public class SmoparkController extends BaseController {

    public static class BasFactoryList implements Serializable {
        private List<BasFactory> faclist;

        public List<BasFactory> getFaclist() {
            return faclist;
        }
        public void setFaclist(List<BasFactory> faclist) {
            this.faclist = faclist;
        }

        public BasFactoryList() {
            super();
        }
    }

    @RequestMapping(value = "/savedetailandlist", method = RequestMethod.POST)
    @ResponseBody
    public ReturnValue SaveDetailAndList (@ModelAttribute("smo") BasFactory detail, @ModelAttribute("smolist") BasFactoryList list) {
        ReturnValue rtv = new ReturnValue();
        System.out.println(detail.getFactid());
        System.out.println(list.getFaclist().get(0).getFactid());
        return rtv;
    };

    @InitBinder("smo")
    public void InitBinderA(WebDataBinder binder) {
        binder.setFieldDefaultPrefix("smo.");
    }

    @InitBinder("smolist")
    public void InitBinderB(WebDataBinder binder) {
        binder.setFieldDefaultPrefix("smolist.");
    }
}
```

pojo 模式下前端的 json 转化函数

```js
utils.addPreParams = function (params, pre) {
  let result = new URLSearchParams()
  for (let key in params) {
    // TODO 判断对象属性是否为数组，如果是数组需要特殊处理
    if (_.isArray(params[key])) {
      let _tempList = params[key]
      for (let i = 0; i < _tempList.length; i++) {
        let item = _tempList[i]
        for (let k in item) {
          result.append(`${pre}.${key}[${i}].${k}`, item[k])
        }
      }
    } else {
      result.append(`${pre}.${key}`, params[key]);
    }
  }
  return result;
};
```