**توضیحاتی در مورد JSDAL**

Version 1.0

|  |  |  |  |
| --- | --- | --- | --- |
| **تاريخ** | **نسخه** | **توضيحات** | **نويسنده** |
| 1/7/1388 | Version 1.0 | توضیحات کلی در مورد JSDAL | محمد صلواتی |

# JSDAL Class

## مسئولیت:

مسئولیتهای این کلاس را میتوان بصورت زیر لیست کرد:

* دریافت Data از DB بوسیله Web Service
* تبدیل اطلاعات دریافت شده از Server و تبدیل آنها از Text به object
* Cache کردن اطلاعات برای جلوگیری از roundtripهای اضافه به Server
* ارائه Service هایی جهت ارتباط با DB

## اجزا کلاس:

1. DataSource Lists

شامل تعدادی List(Array) است که جهت ذخیره اطلاعات دریافتی از DB استفاده میشود

این List ها میتوانند ساختار one-to-many را پشتیبانی کنند.بعنوان مثال لیست مربوط به Category ها دارای Property , RuleList است که یک آرایه از Rules میباشد.این لیست در طول گذرهای مختلف پروژه , بسته به نیازها اضافه میشود.

1. GetData Callback(s)

شامل تعدادی متغیر برای ذخیره آدرس متدهای که پس از ارائه یک سرویس باید به آنجا callback شود میباشد.

به عبارت دیگر , هر کلاس که میخواهد از سرویسهای JSDAL استفاده نماید در اکثر اوقات حداقل یکبار لازم است که متغیر Callback را Set کند. همچنین ما به ازای هر نوع استفاده از سرویس باید یک Callback نیز به JSDAL داده شود تا هنگام callback , client code بداند که چه عملی callback شده است.

به ازای هر کلاس که میخواهد از JSDAL استفاده کند باید یک callback و Callback Code داشته باشیم

1. Public Method(s)

تمامی سرویسهایی که توسط Client قابل استفاده است در این قسمت تعریف شده است

1. PrivateMethod(s)

تمامی متودهای خصوصی در اینجا نگهداری میشوند.

1. Web service Method(s)

تمامی متدهایی که با وب سرویسها در ارتباط هستند(Ajax Web service) در این قسمت نگهداری میشوند.

1. Web service Callback

Callback مربوط به web service ها در این قسمت handle می شود.

1. Exception Handler
2. برای دادن انواع پیامها اخطار استفاده میشود.

## توضیحاتی در مورد نحوه پیاده سازی چند Method:

### JSDAL.GetData(objectId,dataType)

#### نتیجه تابع:

در بین دادهای ذخیره شده جستجو میکند و داده خاصی را بازمیگرداند

#### پارامترها:

* ObjectId:هر گاه برای واکشی یک داده به یک ID نیاز باشد , از این پارامتر استفاده میشود , که معمولا یا PK یک object است یا FK یک جدول است.
* DataType:در هنگام SaveData و GetData برای هر نوع داده ای که به لحاظ واکشی دارای رابطه Business خاص خود است از یک کلید که بصورت string در نظر گرفته شده است استفاده میشود.

#### نحوه عملکرد:

بوسیله پارامتر DataType متوجه میشود که client به چه قسمتی نیاز دارد و قسمت مربوط را واکشی میکند.

\*همچنین قابل ذکر است که DataSourceListها تنها از طریق SaveData و GetData قابل دسترسی هستند.

### JSDAL.SaveData(dataList , dataType,dependency key)

#### نتیجه تابع:

عمل ذخیره داده ها را انجام میدهد

#### پارامترها:

* dataList:معمولا client یک لیستی از Data را که شاید حتی از محتوای آن نیز خبر نداشته باشد در اختیار دارد و میخاهد آنرا ذخیره کند
* dataType:در هنگام SaveData و GetData برای هر نوع داده ای که به لحاظ واکشی دارای رابطه Business خاص خود است از یک کلید که بصورت string در نظر گرفته شده است استفاده میشود.
* dependency Key:گاهی یک داده لازم است درمن داده دیگر ذخیره شود.در این شرایط , این تابع برای ذخیره سازی نیاز به ID والد دارد(one-to-many)

#### نحوه عملکرد:

با توجه به DataType عمل ذخیره سازی را انجام میدهد.

### Public Methods

از آنجایی که نحوه عملکرد توابع عمومی تا حد زیادی به یکدیگر شبیه هستند به توضیح یکی از آنها بسنده می نماییم.

### JSDAL.GetParentCategories (callbackCode)

#### نتیجه تابع:

Categoryهای ریشه را برمیگرداند

#### پارامترها:

* callback Code:Client در هنگام فراخوانی یک code به این تابع میدهد تا پس از callback , client بداند که callback انجام شده مربوط به کدام فراخوانی میباشد.

#### نحوه عملکرد:

در ابتدا تابع check میکند که آیا client قبل از فراخوانی callback را set کرده است یا خیر.در صورت valid بودن callback , تابع به دادهای ذخیره شده را بررسی مینماید تا در صورتی که قبلا این داده از server دریافت شده است آنرا بصورت catch شده بوسیله callback برگرداند , در غیر این صورت web service مربوطه را فراخوانی میکند.

### Ajax Web service Callback

در صورتی که یک web service با موفقیت کاری را به اتمام برساند success callback میکند و بوسیله User Context انتخابی عملیات مربوطه انجام میشود.

\*در سیستم ما بین client و server انتقال اطلاعات بصورت String میباشد.

لذا در هنگام callback , داده با فرمت دریافتی از string بصورت object درمی آیند.این داده ها با فرمت JSON منتقل شده اند.

سپس اقدام به Save کردن آنها مینماییم و در نهایت به client Code , callback میکنیم تا از دادها استفاده کند.