Ext.js Données

Ext.js Données est utilisé pour le chargement et la sauvegarde de toutes les données dans l'application.

Le paquet de données a plusieurs nombre de classes, mais les classes les plus importantes sont:

1. Modèle
2. boutique
3. Procuration

## Modèle:

La classe de base pour modèle est Ext.data.Model. Il représente une entité dans une application. Il lie les données du magasin pour voir. Il a la cartographie d'objets de données d'arrière-plan à la vue dataIndex. Les données sont extraites à l'aide de magasin.

### La création d'un modèle:

Pour créer un modèle que nous devons étendre la classe d’Ext.data.Model et nous avons besoin de définir des champs leur nom et leur cartographie.

Ext.define('StudentDataModel', {

extend: 'Ext.data.Model',

fields: [

{name: 'name', mapping : 'name'},

{name: 'age', mapping : 'age'},

{name: 'marks', mapping : 'marks'}

]

});

Ici, le nom doit être identique au dataIndex que nous déclarons dans la vue et la cartographie doit correspondre aux données statiques ou dynamiques de base de données qui doit être récupérée en utilisant magasin.

## Boutique:

La classe de base pour le magasin est Ext.data.Store. Elle contient les données mises en cache localement, ce qui doit être rendu sur la vue à l'aide des objets de modèle. Magasin récupère les données en utilisant des procurations qui a le chemin défini pour les services pour récupérer les données d'arrière-plan.

Stocker les données peut être extraites à partir de deux façons statiques ou dynamiques.

### Magasin statique:

Pour magasin statique, nous aurons toutes les données présentes dans le magasin comme ci-dessous:

Ext.create('Ext.data.Store', {

model: 'StudentDataModel',

data: [

{ name : "Asha", age : "16", marks : "90" },

{ name : "Vinit", age : "18", marks : "95" },

{ name : "Anand", age : "20", marks : "68" },

{ name : "Niharika", age : "21", marks : "86" },

{ name : "Manali", age : "22", marks : "57" }

];

});

### Magasin dynamique:

Les données dynamiques peuvent être récupérées en utilisant un proxy. Nous pouvons avoir proxy peut les données extraites de l'Ajax, Repos et Json.

## Procuration:

La classe de base pour le proxy est Ext.data.proxy.Proxy. Proxy est utilisé par les modèles et les magasins pour gérer le chargement et la sauvegarde des données du modèle.

Il existe deux types de procurations:

1. Proxy Client
2. Proxy Server

### Proxy Client

proxies clients comprennent la mémoire et le stockage local en utilisant HTML5 stockage local.

### Proxy Server

proxies Server gère les données de serveur distant en utilisant les données Ajax, JSON et le service Rest.

#### Définition de procurations dans le serveur:

Ext.create('Ext.data.Store', {

model: 'StudentDataModel',

proxy : {

type : 'rest',

actionMethods : {

read : 'POST' // Get or Post type based on requirement

},

url : 'restUrlPathOrJsonFilePath', // here we have to include the rest URL path which fetches data from database or Json file path where the data is stored

reader: {

type : 'json', // the type of data which is fetched is of JSON type

root : 'data'

},

}

});