Titanium App Development

DALAPI

* Uses json as database
* Persists json responses in db according to a caching configuration file
* Consider this json response from Categories/GetAll
  + var allCategories =[{ Id: 1, Name: “First Category”}, { Id: 2, Name: “Second Category” }];
  + var secondCategory = categoriesDAL.getById(2);

The generated DAL provides standard GetAll, GetByIndex, GetByPK, GetByIndex, Create, Delete, Update.

**Get By Primary Key**

GetByPK works by first getting the index of the PK value from an array of PK values. It uses that index to get the item from the json array.

**Get By Index**

GetByIndex is similar to GetByPK, except that it must use two arrays to lookup a value instead of one. The first array

// web service response

var allCategories = [{ Id: 1, Name: 'First Category' }, { Id: 2, Name: 'Second Category' }];

// creates a dal for an array

function DAL(array)

{

/\* Get By ID (this is generated because the property "Id" was set for the primary key setting for dalapi)

---------------------------- \*/

// create a local variable to store the unique ids from the array

var \_ids = [];

array.map(function (item) { \_ids.push(item.Id); });

// gets an item by id

this.getById = function (id) {

var index = \_ids.indexOf(id);

return array[index];

};

/\* Get By Name (this is generated because the property "Name" was added to the indexes array setting for dalapi)

---------------------------- \*/

// this first part is to create the index

var \_namesLookup = [];

var \_namesValues = [];

array.map(function (item) {

// add the item's name to the \_names array if it doesn't already exist

if(\_namesLookup.indexOf(item.Name) == -1)

{

\_namesLookup.push(item.Name);

\_namesValues.push([]);

}

// get the index of the name

var index = \_namesLookup.indexOf(item.Name);

// add the item to the values array

\_namesValues[index].push(item);

});

// returns an array of items that match the name, if any

this.getByName = function (name) {

var index = \_namesLookup.indexOf(name);

return \_namesValues[index];

}

}

// create a new DAL

var CategoriesDAL = new DAL(allCategories);

// lookup an item by id

var numberTwo = CategoriesDAL.getById(2);

alert("the name for getById(2) = " + numberTwo.Name);

// lookup an item by name

var arr = CategoriesDAL.getByName("Second Category");

alert("getByName('Second Category') found " + arr.length + " results, the first has an id of " + arr[0].Id);