TIDOMINO DEVELOPER GUIDE

Date	19 December 2013
Release	1.0.1
Prepared By	John Jardin (john.jardin@ukuvuma.co.za)

CONTENTS

INTRODUCTION

TiDomino is a JavaScript module that allows Appcelerator Titanium developers to use IBM Domino programming patterns to structure local data sets and integrate with IBM Domino and XWork Applications.

MINIMUM REQUIREMENTS

- Appcelerator Titanium Studio 3.0.0 and above
- IBM Domino Server 8.5.3 and above or IBM XWork Server 8.5.3 and above.

RELEASE NOTES

RELEASE 1.0.1

TIDOMINO.JS

None

TIDOMINO KITCHEN SINK

- Modified parameters in app.js and TiDomino_spec.js to reference TiDomino Sample DB

TIDOMINO.NSF

None

TIDOMINO_SAMPLEDB.NSF

- The TiDomino Sample DB is a new Notes Database that contains 3 forms and sample data for testing the TiDomino Kitchen with

INSTALLATION

INSTALL NOTES DATABASES ON DOMINO OR XWORK SERVER

1. Parts of the TiDomino download from OpenNTF are 2 Notes Databases (**tidomino.nsf and tidomino_sample.nsf**). You will need to copy these databases to your IBM Domino or XWork Server's "**Data**" directory.
2. In the Domino Administrator Client, under the "**Files**" tab, navigate to the 2 TiDomino Notes Databases and do the following:
    a. Sign both Databases either with the Server ID or with whatever id you would prefer, as long as this ID has **Manager** access to both Notes Databases and has **Read/Write** access to applications you want your mobile app to integrate with.
    b. Tweak the Access Control List of both Notes Databases to fit the Company's policies. This is mainly for Admins and Servers. (**NB:** Ensure that "**Anonymous**" access has "**No Access**", which is the default setting for both Notes Databases)

NOTE: Default Access to the TiDomino Database is currently "**Manager**". Please change this to "**Reader**" with all options deselected.

NOTE: Default Access to the TiDomino Sample DB Database is currently "**Manager**". Please change this to "**Editor**" with Deletion Rights selected.

NOTE: If you need to place the Notes Databases (**tidomino.nsf and tidomino_sampledb.nsf**) in a different folder within the "**Data**" directory, this is fine. All you will need to do is:

- Modify the variable "**TI_DOMINO_PATH**" in the **TiDomino.js** file.
- Use "**/**" instead of "****" in your path string, when required.

INSTALL TIDOMINO.JS IN YOUR TITANIUM MOBILE APPLICATION

Depending on if your mobile project is based on Titanium's "Classic" or "Alloy" template, you will need to do the following:

1. For a Titanium Classic Project:
    a. Import the "TiDomino.js" file into the "**Resources**" folder.
    b. Whenever you want to reference or use the TiDomino module, do the following:
        i. `var Session = require('TiDomino');`
2. For a Titanium Alloy Project:
    a. If it doesn't exist yet, create a folder called "**lib**" inside the "**app**" folder in your project.
    b. Import the "TiDomino.js" file into this "**lib**" folder.
    c. Whenever you want to reference or use the TiDomino module, do the following:
        i. `var Session = require('TiDomino');`

IMPORT TIDOMINO KITCHEN SINK INTO TITANIUM STUDIO

1. Extract the TiDomino Kitchen Sink zip file
2. In Titanium Studio, do the following:
    a. Click on “File\Import”
    b. Expand “General” and select “Existing Projects into Workspace”.
    c. Click Next.
    d. For “Select Root Directory”, click “Browse” and navigate to the extracted “TiDomino Kitchen Sink” folder.
    e. Select the “TiDomino Kitchen Sink” folder and click on “Open”.
    f. Check the “TiDomino Kitchen Sink” project.
    g. Check “Copy Projects To Workspace”.
    h. **OPTIONAL:** Add the Project to your working sets.
    i. Click “Finish”.
3. In the Kitchen Sink’s “**app.js**”, modify the parameters at the top of the file. These parameters are used by each of the functions in the Kitchen Sink Project, so as to test the TiDomino functions against your IBM Domino or XWork environment.
4. To perform quick tests, run the TiDomino Kitchen Sink Project using the iPhone or iPad Simulator.

TIDOMINO FUNCTIONS

DECLARE A NEW SESSION

Before you are able to call TiDomino's functions, you have to instantiate a new "Session" object and assign it to a variable in your Titanium Mobile Project.

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
```

"**ss**" is your variable that you will now use to initiate TiDomino Functions.

CREATE A SERVER CONNECTION

DESCRIPTION

Creates a Server Connection Object holding details about the Domino Server you will be connecting to.

SYNTAX

```
createServerConnection(serverKey, dominoServer, serverHostName, isHTTPS,
requireInternetConnection);
```

PARAMETERS

serverKey

String. Any key to uniquely identify the Server Connection. (You can have more than1)

dominoServer

String. The name of your Domino or XWork Server.

serverHostName

String. The Hostname of your Domino or XWork Server

isHTTPS

Boolean. Will the Server Connection be connecting using HTTPS?

requireInternetConnection

Boolean. Does this Server Connection require an internet connection, or will it connect to a local Domino Server on the network or on the current machine?

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);
```

AUTHENTICATE WITH IBM DOMINO OR XWORK SERVER

DESCRIPTION

Authenticates your mobile session with an IBM Domino or XWork Server.

SYNTAX

```
loginUser(serverConnection, username, password, functionToRun, functionParameters);
```

PARAMETERS

serverConnection

ServerConnection Object. A ServerConnection Object for the target Domino or XWork Server.

Username

String. A username used to authenticate with your Domino or XWork Server.

Password

String. A password used to authenticate with your Domino or XWork Server.

functionToRun

JavaScript Function. The JavaScript function that you want to execute once the “loginUser” function is complete. (You provide a reference to the JavaScript Object you want to run. This is not a String value)

functionParameters

JSON Object. The parameters you would like to pass to the “functionToRun” JavaScript Function. This must be passed as a JSON Object. (**NOTE:** You can specify null if you don’t want to pass parameters)

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);
ss.loginUser(con, "John Doe", "password123", myFunction, {processType:"1"});
```

GET ALL DOCUMENTS BY KEY

DESCRIPTION

Returns a collection of NotesDocument Objects from a Target Notes Database View using a key.

SYNTAX

getAllDocumentsByKey(key, exactMatch, fieldsToReturn, functionToRun, functionParameters);

PARAMETERS

key

String. A key to match the first sorted column in your Notes View.

exactMatch

Boolean. True if you want to find an exact match, or False for a partial match.

fieldsToReturn

String. The Field Names in your Notes Form that you want returned. Separate each fieldname with a comma.

functionToRun

JavaScript Function. The JavaScript function that you want to execute once the "loginUser" function is complete. (You provide a reference to the JavaScript Object you want to run. This is not a String value)

functionParameters

JSON Object. The parameters you would like to pass to the "functionToRun" JavaScript Function. This must be passed as a JSON Object. (**NOTE:** You can specify null if you don't want to pass parameters)

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);

var db = ss.getDatabase(con, "myDb.nsf");
var view = db.getView("ViewAliasName");
view.getAllDocumentsByKey("ColumnKey", true, "Date,Employee,TimeAllocation",
myFunction, null);
```

SUBMIT A NEW RECORD OR UPDATE AN EXISTING RECORD

DESCRIPTION

Submits a new record or updates an existing record in a Notes Database on the target Domino or XWork Server.

SYNTAX

doc.save(functionToRun, functionParameters);

PARAMETERS

functionToRun

JavaScript Function. The JavaScript function that you want to execute once the "loginUser" function is complete. (You provide a reference to the JavaScript Object you want to run. This is not a String value)

functionParameters

JSON Object. The parameters you would like to pass to the "functionToRun" JavaScript Function. This must be passed as a JSON Object. (**NOTE:** You can specify null if you don't want to pass parameters)

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);

var db = ss.getDatabase(con, "myDb.nsf");
var doc = db.createDocument({
        Form:'TitaniumDoc',
        Status:'New'
});
doc.Title = "Test Title";

doc.save(myFunction, null);
```

RUN XAGENT

DESCRIPTION

Executes an XAgent on the Domino or XWork Server.

SYNTAX

db.runXAgent(xpage, parameters, functionToRun, functionParameters);

PARAMETERS

xpage

String. The name of the XPage you want to execute.

parameters

JSON Object. Any parameters that you want to pass to the XAgent. (**NOTE:** This needs to be passed as a JSON Object, but will be converted into a String by the time your XAgent gets a handle on it)

functionToRun

JavaScript Function. The JavaScript function that you want to execute once the "loginUser" function is complete. (You provide a reference to the JavaScript Object you want to run. This is not a String value)

functionParameters

JSON Object. The parameters you would like to pass to the "functionToRun" JavaScript Function. This must be passed as a JSON Object. (**NOTE:** You can specify null if you don't want to pass parameters)

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);

var db = ss.getDatabase(con, "myDb.nsf");
db.runXAgent("mypage.xsp", {processType:"1"}, myFunction, null);
```

LOGOUT OF DOMINO OR XWORK

DESCRIPTION

Logs you out of the current session with the IBM Domino or XWork Server

SYNTAX

ss.logoutUser(serverConnection, functionToRun, functionParameters);

PARAMETERS

serverConnection

ServerConnection Object. A ServerConnection Object for the target Domino or XWork Server.

functionToRun

JavaScript Function. The JavaScript function that you want to execute once the “loginUser” function is complete. (You provide a reference to the JavaScript Object you want to run. This is not a String value)

functionParameters

JSON Object. The parameters you would like to pass to the “functionToRun” JavaScript Function. This must be passed as a JSON Object. (**NOTE:** You can specify null if you don’t want to pass parameters)

EXAMPLE

```
var Session = require('TiDomino');
var ss = new Session();
var con = ss.createServerConnection("acme", "ACME/Server", "www.acme.com", false,
true);

ss.logoutUser(con, myFunction, null);
```