Tableau Web Data Connector Development Guide

Important Links:

Tableau’s documentation of how to setup a web data connector. -1st place to start

<http://tableau.github.io/webdataconnector/docs/wdc_tutorial>

Rally Documentation on how to make API calls

<https://rally1.rallydev.com/slm/doc/webservice/>

YouTube playlist created by tableau developers going over how to create and debug web data connectors

<https://www.youtube.com/watch?v=og6p8BI4-Cw&list=PL_qx68DwhYA8E80s9JOnQzzPkX7sWK-pl>

YouTube playlist for everything you would need to know to create a project in Node JS

<https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBMdkKFn3HasZnnAqVjzHn_>

Tips to get started

There are really two main things you need to know how to do when developing a web data connector.

* How to define tables to be created in tableau

This is a good link to examples tableau created of different web data connectors and there tables

<http://tableau.github.io/webdataconnector/docs/wdc_samples>

As well as community created web data connectors

<https://tableau.github.io/webdataconnector/community/>

* How to create API calls to Rally

This is a good link to look at if you are having trouble creating the proper syntax for API calls

<https://help.rallydev.com/grid-queries>

There aren’t a lot of examples about using the rally api but I did find a couple of articles on stackoverflow.com and quick google searches.

Moving Through the Project

**App.js** is called on startup <http://localhost:3000>

Sends the user to the **index.html page and runs the rallyWDC.js file**

User clicks the **connectionbutton** and this calls the **click** function in the **rallyWDC.js** file

The user then authenticates with rally and is redirected to <http://localhost:3000/redirect>

route in **app.js** file

The /redirect route then sends redirects the user to the **/getWorkspaces** route with a get request

The get request to **/getWorkspaces** sends an api request to rally get the users workspaces and then renders the **workspaceSelectoin** view

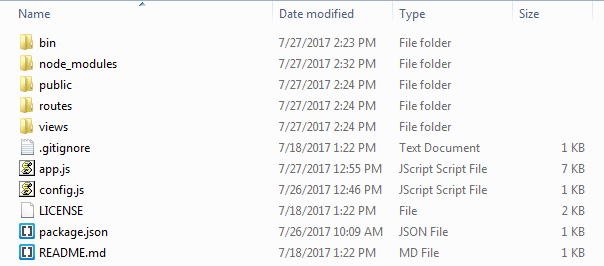
The **workspaceSelection.ejs** view displays a drop down with the available workspaces and when the user clicks the **workspacebuttun(ID name of the button)** a post request is sent to **/getProjects route**

The /**getProjects** route gets the workspace the user selected from the drop down and sends a api request to rally finding the projects under that workspace and the renders the **projectSelection.ejs** file

The **projectSelectoin.ejs** file displays a drop down with all the parent projects under the users workspace and once the user clicks the **sendData(ID name of the button)** button a post request to **/getTableauData** route

The **/getTableauData** route retrieves the project the user selected and saves that information for the client in a cookie. Then sends the user back to the main index route of the server **‘/’** this displays the **index.html** page again and loads the web data connector **rallyWDC.js** file

Now when the user clicks the

Project Model 

Config.js

This is where all your rally authentication information is stored this was explained the in installation file.

The variables defined were

HOSTPATH': 'http://localhost',

'PORT': 3000,

'CLIENT\_ID': 'f644fa154f85470f84f9edaf0c8bc59b',

'CLIENT\_SECRET': 'CKzeUErtTbmoR8wzBCCkr1E8y9mpPU10oz8d8wuqx9M',

'REDIRECT\_PATH': '/redirect'

Package.json

This is all the node modules that are being used in the project

"dependencies": {

"body-parser": "~1.17.1", used to retrieve information from body of post request

"cookie-parser": "~1.4.3", allow users to access stored data on server on the client side HTML

"ejs": "^2.5.6, this allows you to create .ejs files which are just like html files accept you can input dynamic content to display to the user

"express": "~4.15.2", server framework

"express-session": "^1.15.3", create a session in express to pass variables on the server side

"moment": "^2.18.1", easy dates

"nodemon": "^1.11.0", allows live updates to your project so you don’t have to reopen your application to see development changes

"request": "^2.81.0" easy http requests

},

"devDependencies": {

"nodemon": "^1.11.0"

}

**app.js**

This is the main server file that handles all the requests it is called whenever <http://localhost:3000> with any path after like / or /redirect /getProjects etc…

**Variables**

var getWorkspaces = require('./routes/getWorkspaces');

var getProjects = require('./routes/getProjects');

var getTableauData = require('./routes/getTableauData');

var config =require('./config.js');

var clientID = config.CLIENT\_ID;

var clientSecret = config.CLIENT\_SECRET;

var redirectURI = config.HOSTPATH + ":" + config.PORT + config.REDIRECT\_PATH;

**Route Info**

App.get(/)- This is called whenever a GET request to <http://localhost:3000> is called. This will be on startup and when the user finishes filtering data and redirects to the index page

App.use(/getWorkspaces, getWorkspaces)- This is called when any type of request (GET,POST) is sent to <http://localhost:3000/getWorkspaces> and will use the getWorkspaces route functions.

This is the same for the getProjects and getTableauData app.use functions

**Routes Folder**

This is where all of our route functions are stored. So when a user is sent to a route these functions will be called.

**getWorkspaces**

This route create a HTTP get request and sends an api call to rally to get the users available workspaces

**getProjects**

This route gets the users selected workspace and then uses that info to build another rally api get request and returns all the available projects under that workspace

**getTableauData**

This route takes in the users selected project and then stores that data for the web data connector (rallyWDC.js) file on the client side to be able to access this info to create its query

**Views**

This is where all the filtering webpages are shown that are dynamically displayed with information the user inputted to us as well as the returned information from our rally API calls to get workspaces and projects

**Public**

**Index.html**

This is our index html page that loads our web data connector code any time it is loaded

**JS folder**

**rallyWDC.js**

This is our actual web data connector file where we communicate with tableau.

I would become vary familiar with this tutorial that tableau put together as they are way better at explaining what’s going on in this file than me

[**http://tableau.github.io/webdataconnector/docs/wdc\_tutorial**](http://tableau.github.io/webdataconnector/docs/wdc_tutorial)

And a very good video that goes over the authentication process

<https://www.youtube.com/watch?v=l4zX8zm_UWQ>

Already completed version

<https://k504866430.github.io/rally-tableau/>

Examples

So I created a bunch of other tables which I feel would be good reference to look at when/if you need to create new tables

var project\_cols=[

//{ id : "ID", alias : "AutoID", dataType : tableau.dataTypeEnum.float },

{ id : "Name", alias : "Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "ParentName", alias : "Parent Name", dataType : tableau.dataTypeEnum.string },

{ id : "ParentID", alias : "Parent ID", dataType : tableau.dataTypeEnum.string },

{ id : "WorkspaceName", alias : "Workspace Name", dataType : tableau.dataTypeEnum.string },

{ id : "WorkspaceID", alias : "Workspace ID", dataType : tableau.dataTypeEnum.string },

];

var projectTable ={

id : "Project",

alias : "Project Data",

columns : project\_cols

};

var iteration\_cols=[

{ id : "Name", alias : "Name", dataType : tableau.dataTypeEnum.string },

{ id : "StartDate", alias : "Start Date", dataType : tableau.dataTypeEnum.date },

{ id : "EndDate", alias : "End Date", dataType : tableau.dataTypeEnum.date },

{ id : "ProjectName", alias : "Project Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "PlanEstimate", alias : "Plan Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "PlannedVelocity", alias : "Planned Velocity", dataType : tableau.dataTypeEnum.float },

{ id : "ObjectID", alias : "Iteration ID", dataType : tableau.dataTypeEnum.string },

{ id : "State", alias : "State", dataType : tableau.dataTypeEnum.string },

{ id : "TaskActualTotal", alias : "Task Actual Total", dataType : tableau.dataTypeEnum.float },

{ id : "TaskEstimateTotal", alias : "Task Estimate Total", dataType : tableau.dataTypeEnum.float },

{ id : "TaskRemainingTotal", alias : "Task Remaining Total", dataType : tableau.dataTypeEnum.float },

];

var iterationTable ={

id : "Iteration",

alias : "Iteration Data",

columns : iteration\_cols

};

var release\_cols=[

//{ id : "ID", alias : "AutoID", dataType : tableau.dataTypeEnum.float },

{ id : "Name", alias : "Name", dataType : tableau.dataTypeEnum.string },

{ id : "ObjectID", alias : "Release ID", dataType : tableau.dataTypeEnum.string },

{ id : "Accepted", alias : "Accepted", dataType : tableau.dataTypeEnum.float },

{ id : "PlanEstimate", alias : "Plan Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "PlannedVelocity", alias : "Planned Velocity", dataType : tableau.dataTypeEnum.float },

{ id : "ProjectName", alias : "Project Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "EndDate", alias : "End Date", dataType : tableau.dataTypeEnum.date },

{ id : "StartDate", alias : "Start Date", dataType : tableau.dataTypeEnum.date },

{ id : "State", alias : "State", dataType : tableau.dataTypeEnum.string },

{ id : "TaskActualTotal", alias : "Task Actual Total", dataType : tableau.dataTypeEnum.float },

{ id : "TaskEstimateTotal", alias : "Task Estimate Total", dataType : tableau.dataTypeEnum.float },

{ id : "TaskRemainingTotal", alias : "Task Remaining Total", dataType : tableau.dataTypeEnum.float },

//{ id : "LastUpdated", alias : "Last Update", dataType : tableau.dataTypeEnum.date },

];

var releaseTable ={

id : "Release",

alias : "Release Data",

columns : release\_cols

};

var defect\_cols = [

//{ id : "ID", alias : "AutoID", dataType : tableau.dataTypeEnum.float },

{ id : "FormattedID", alias : "FormattedID", dataType : tableau.dataTypeEnum.string },

{ id : "ClosedDate", alias : "Closed Date", dataType : tableau.dataTypeEnum.date },

{ id : "CreationDate", alias : "Creation Date", dataType : tableau.dataTypeEnum.date },

{ id : "AcceptedDate", alias : "Accepted Date", dataType : tableau.dataTypeEnum.date },

{ id : "PlanEstimate", alias : "Plan Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "Rank", alias : "Rank", dataType : tableau.dataTypeEnum.float },

{ id : "ScheduleState", alias : "Schedule State", dataType : tableau.dataTypeEnum.string },

{ id : "State", alias : "State", dataType : tableau.dataTypeEnum.string },

{ id : "Enviroment", alias : "Enviroment", dataType : tableau.dataTypeEnum.string },

{ id : "Priority", alias : "Priority", dataType : tableau.dataTypeEnum.string },

{ id : "FoundBy", alias : "Found By", dataType : tableau.dataTypeEnum.string },

{ id : "IterationName", alias : "Iteration Name", dataType : tableau.dataTypeEnum.string },

{ id : "IterationID", alias : "Iteration ID", dataType : tableau.dataTypeEnum.string },

{ id : "Owner", alias : "Owner", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectName", alias : "Project Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "ReleaseName", alias : "Release Name", dataType : tableau.dataTypeEnum.string },

{ id : "ReleaseID", alias : "Release ID", dataType : tableau.dataTypeEnum.string },

{ id : "Tags", alias : "Tag", dataType : tableau.dataTypeEnum.string },

{ id : "ObjectID", alias : "Defect ID", dataType : tableau.dataTypeEnum.float },

{ id : "Requirement\_FormattedID", alias : "Requirement FormattedID", dataType : tableau.dataTypeEnum.string },

{ id : "Requirement\_Iteration", alias : "Requirement Itearation Name", dataType : tableau.dataTypeEnum.string },

{ id : "Requirement\_IterationID", alias : "Requirement Iteration ID", dataType : tableau.dataTypeEnum.string },

{ id : "RunDate", alias : "Run Date", dataType : tableau.dataTypeEnum.date },

//{ id : "RunProject", alias : "Run Project", dataType : tableau.dataTypeEnum.string },

{ id : "Severity", alias : "Severity", dataType : tableau.dataTypeEnum.string },

];

var defectTabel = {

id : "Defect",

alias : "Defect Data",

columns : defect\_cols

};

var task\_cols=[

// { id : "ID", alias : "AutoID", dataType : tableau.dataTypeEnum.float },

{ id : "FormattedID", alias : "FormattedID", dataType : tableau.dataTypeEnum.string },

{ id : "IterationName", alias : "Iteration Name", dataType : tableau.dataTypeEnum.string },

{ id : "IterationID", alias : "Iteation ID", dataType : tableau.dataTypeEnum.string },

{ id : "Actuals", alias : "Actuals", dataType : tableau.dataTypeEnum.float },

{ id : "Estimate", alias : "Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "ToDo", alias : "To Do", dataType : tableau.dataTypeEnum.float },

{ id : "TimeSpent", alias : "Time Spent", dataType : tableau.dataTypeEnum.float },

{ id : "State", alias : "State", dataType : tableau.dataTypeEnum.float },

{ id : "Tags", alias : "Tags", dataType : tableau.dataTypeEnum.string },

{ id : "TaskType", alias : "Task Type", dataType : tableau.dataTypeEnum.string },

{ id : "Owner", alias : "Owner", dataType : tableau.dataTypeEnum.string },

{ id : "Project", alias : "Project Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "Release", alias : "Release Name", dataType : tableau.dataTypeEnum.string },

{ id : "ReleaseID", alias : "Release ID", dataType : tableau.dataTypeEnum.string },

{ id : "WorkProduct\_FormattedID", alias : "Work Product ID", dataType : tableau.dataTypeEnum.string },

{ id : "LastUpdateDate", alias : "Last Update", dataType : tableau.dataTypeEnum.date },

{ id : "CreationDate", alias : "Creation Date", dataType : tableau.dataTypeEnum.date },

{ id : "ObjectID", alias : "Task ID", dataType : tableau.dataTypeEnum.string },

{ id : "RunDate", alias : "Run Date", dataType : tableau.dataTypeEnum.date },

//{ id : "RunProject", alias : "Run", dataType : tableau.dataTypeEnum.string },

];

var taskTabel = {

id : "Task",

alias : "Task Data",

columns : task\_cols

};

var portfolioItem\_cols=[

// { id : "ID", alias : "AutoID", dataType : tableau.dataTypeEnum.float },

{ id : "FormattedID", alias : "FormattedID", dataType : tableau.dataTypeEnum.string },

{ id : "ObjectID", alias : "Portfolio Item ID", dataType : tableau.dataTypeEnum.string },

{ id : "PortfolioItemName", alias : "Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectName", alias : "Project Name", dataType : tableau.dataTypeEnum.string },

{ id : "ProjectID", alias : "Project ID", dataType : tableau.dataTypeEnum.string },

{ id : "ParentName", alias : "Parent Name", dataType : tableau.dataTypeEnum.string },

{ id : "ParentID", alias : "Parent ID", dataType : tableau.dataTypeEnum.string },

{ id : "State", alias : "State", dataType : tableau.dataTypeEnum.string },

{ id : "AcceptedLeafStoryCount", alias : "Accepted Leaf Story Count", dataType : tableau.dataTypeEnum.float },

{ id : "AcceptedLeafPlanEstimate", alias : "Accepted Leaf Plan Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "ActualEndDate", alias : "Actual End Date", dataType : tableau.dataTypeEnum.date },

{ id : "ActualStartDate", alias : "Actual Start Date", dataType : tableau.dataTypeEnum.date },

{ id : "InvestmentCategory", alias : "Investment Category", dataType : tableau.dataTypeEnum.string },

{ id : "JobSize", alias : "Job Size", dataType : tableau.dataTypeEnum.float },

{ id : "LeafStoryCount", alias : "Leaf Story Count", dataType : tableau.dataTypeEnum.float },

{ id : "LeafStoryPlanEstimateTotal", alias : "Leaf Story Plan Estimate Total ", dataType : tableau.dataTypeEnum.float },

{ id : "PercentDoneByStoryCount", alias : "Percent Done By Stop Count", dataType : tableau.dataTypeEnum.float },

{ id : "PercentDoneByStoryPlaneEstimate", alias : "Percent Done By Plan Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "PlannedEndDate", alias : "Planned End Date", dataType : tableau.dataTypeEnum.date },

{ id : "PlannedStartDate", alias : "Planned Start Date", dataType : tableau.dataTypeEnum.date },

{ id : "PortfolioItem\_Type", alias : "Type", dataType : tableau.dataTypeEnum.string },

{ id : "PerliminaryEstimate", alias : "Perliminary Estimate", dataType : tableau.dataTypeEnum.string },

{ id : "UnEstimatedLeafStoryCount", alias : "UnEstimated Leaf Story Count", dataType : tableau.dataTypeEnum.string },

{ id : "Tags", alias : "Tags", dataType : tableau.dataTypeEnum.string },

{ id : "RefinedEstimate", alias : "Refined Estimate", dataType : tableau.dataTypeEnum.float },

{ id : "Description", alias : "Description", dataType : tableau.dataTypeEnum.string },

{ id : "EPMSid", alias : "EPMSid", dataType : tableau.dataTypeEnum.string },

{ id : "Release", alias : "Release", dataType : tableau.dataTypeEnum.string },

{ id : "ReleaseID", alias : "Release ID", dataType : tableau.dataTypeEnum.string },

{ id : "RunDate", alias : "RunDate", dataType : tableau.dataTypeEnum.date },

// { id : "RunProject", alias : "Run", dataType : tableau.dataTypeEnum.string },

];

var portfolioItemTabel = {

id : "PortfolioItem",

alias : "Portfolio Item Data",

columns : portfolioItem\_cols

};

schemaCallback([userStoryTabel, iterationTable, projectTable, releaseTable,defectTabel,taskTabel,portfolioItemTabel]);