**API Socket Creation Process**

5/18/16

**Getting to know the API functions and pages**

Videos summarizing API socket set-up and use are a good place to start:

Type in URL: 127.0.0.1:8500/www\_adilas/api and press enter. This will redirect to <http://127.0.0.1:8500/www_adilas/web/show_adilas_api_calls.cfm>. Click on ‘api help file’ at the top of the page. Scroll down to watch the 3 videos. The descriptions below are a quick outline of what is shown in the videos and can be a reminder if you’ve done it all before.

Video 1: Setting up permissions to use the API.

1. **Set settings:** Go to: settings>corp wide settings. Scroll down to the ‘Show Corp On The Web’ area (currently #69). In the drop down select ‘Yes, Show on Web’. Scroll to the bottom of the page and click ‘Save Settings’ button. This will allow you to show your inventory online to your customers. If no (default) you can only use information internally in Adilas.
2. **Setup username and Password:** Go to: settings>more options>adilas api socket settings. Set up username, password and user id’s allowed. Click on the API socket methods that you would like to use, then click the ‘submit’ button at the bottom of the page. This will allow you to use the sockets that you have opened by selecting the checkbox.

Video 2 & 3: The videos show plugging in live, but can be used in learning about local API development as well. If developing watch the videos and use local instead of live (127.0.0.1:8500/www\_adilas/api instead of https://www.adilas.biz/api.

1. **Got to API page**. Type in URL: 127.0.0.1:8500/www\_adilas/api and press enter. This brings you to the ‘Show/interact with the adilas API’ page. Click ‘api documentation and samples’ link at the top of the page. This will take you to documentations (click on the name of API to view). Click on ‘View Sample’ button to create the API call in demo mode.
2. **Input username and password**. Open the API permissions page from section 2 above to have reference to the Corp key, username, password, and user id that you set up. Put that information in the sample. Click the ‘Prep & Convert It’ button. This will create the JSON that will run. Click the ‘Run It’ button. This should return an array of structures with all of the information requested from the API.
3. **Set test mode**. If developing, play with this test mode so that all of the debugging shows: <http://127.0.0.1:8500/www_adilas/web/test_adilas_api_calls.cfm> (just switch out the ‘show’ and the ‘test’ in the URL.
4. **Creating the wrapper to call the cfc.** The API function starts on the adilas\_api\_calls.cfm page. Validation is carried out, then on line 547 the web page black box wrapper is included (pulled into the page- example: addWebCustomer.cfm). The wrapper calls the .cfc where most of the work is done. You will only be making changes on the black box page and the .cfc. When creating the wrapper, check for all outbound pieces coming from the cfc. Not all of them are set right at the very bottom. Most of the outbound values are tied to the retData\_st. These outbound values could be strings, numbers, lists, arrays, queries, etc. This is how you get information out of the cfc’s and method calls.

**These steps not found in the video:**

1. **Get the API page number.** All new web page id numbers are issued from the live data 0 server. Go to ‘Add/Edit Web API Documentation’ under ‘System Maintenance’ on the classic home page. The ’page number’/’page id’ is on the left hand side of the page name. If starting a new page click the “Add A New Page/Function” link at the top of the page. When a new page is created the page number will be automatically assigned. As a note, make sure that the new method does not exist before making a new one. Eventually, this documentation that is on data 0 will be passed to all other servers including local testing environments. If you need to pull down a copy of the live API socket documentation, do the following. Finish up things on the live server. Login to your local environment. From the classic homepage, go to the ‘system maintenance’ section. Look for the ‘Developer’s Homepage’ option. Click go. Once on that page, choose the option from the top that says ‘update tables’. Pull the different tables down from data 0 to your local machine. Once again, use data 0 as the master and then pull down data locally from there.
2. **Turn on the specific API socket permission for the specific corporation.** This is still on your local box. You may need to fake out your local copy and flip the main web/API documentation password for the current method to “mainAPI”. This is required for the method to show up in the adilas API assignment page. Go to ‘Manage Corp Info & Permissions’ under ‘System Management’ on the classic home page. On the corp\_administrator\_home.cfm page click ‘more options’, then ‘adilas api socket settings’. Select the API Socket Methods that you want to allow, then click ‘Submit’ at the bottom of the page. Once again, this section is all done from your local environment.
3. **Editing documentation**: When it is time to really edit the documentation, go back to live and do the documentation on data 0. From Classic Homepage, click Add/Edit Web API Documentation on System Maintenance section then click ‘go’. Search for the name of your API, then click on the name. It will take you to API Documentation Subs page. There you can add or modify subs of the API. As a note, the way you can update and add things to live but not have it effect anything is the main password field on the web/API documentation. Leave it blank until the very end. When the timing is right (ready to be launched and go live), change that password value to the string “mainAPI”. That is kinda like the final switch for turning on the API socket as a new live option.
4. **Pass documentation to servers.** Here are some more general notes on pushing documentation from server to server. As a reminder, the main server for the documentation is live on data 0. To pass the documentation to another server, go from the classic homepage. Select ‘Developer’s Hompage’ from the ‘System Homepage’ drop down and click ‘go’. Click the ‘update tables’ link at the top of the page. If for some reason you don’t have the correct permission, use this local URL to skip the permission: http://127.0.0.1:8500/www\_adilas/top\_secret/secure/developer\_update\_tables.cfm?sec=skip

Update the ‘Web Pages’, ‘Help Files’, ‘Payee Permission List’, ‘Web Page Parameters’, ‘Web Page Results’, ‘Web Page Sub Results’ and the ‘Web Page Samples’. **\*\*\*\*\*Important Note:** **pull only from live to local, not the other way around.**

**Quick Overview:**

Building an API from ‘Scratch’:

Start with the cfc to create the function needed. Create the wrapper and design how we want to use the function. Create the parameters and validation on the wrapper (black box). Then we document the parameters, and make example. Then update the web pages and documentation.

**Quick Checklist:**

1. Make sure the desired API process exists on a cfc or build the main method call: There are three possible locations for cfcs: 1. top\_secret/cfc, 2. shop/webComponents, and 3. web/cfc. Most of the core ones will be in top\_secret/cfc.
2. Go claim an id number for the method. This is done through the api documentation. Make sure it doesn’t already exist. This needs to be done live on data 0 (see long description #7 above).
3. On your local box, pull down the live settings from **data 0** using the developer homepage update tables. Note: eventually we will have to go back and enable all of settings to turn it on- this is done as the very last thing.
4. Start building the wrapper. Within the wrapper, include defaults, validation, and history calls. Make sure and get all of the outbound pieces from the main cfc or internal method.
5. On your local box pretend that this is a valid API. To do this you will need to change the password 1 on your local documentation to mainAPI. This is required on your local machine in order to see it as a selectable option.
6. Turn on everything locally and start testing.
7. Once testing and signoff are done, push files up to all servers.
8. Add your documentation (live- on data 0). Note: if you have added anything locally, make sure you paste it up to the live documentation.
9. Turn on the specific API socket permission for the specific corporation (see #5 above).
10. The final step is making sure that the live password 1 value is set to mainAPI. This is the final turn on switch. If you are doing anything live, leave this value off until we are totally ready. This is the final ‘done’ switch. Page/function output type also must be JSON.
11. Log into the live server and test the API calls.
12. Cascade live documentation to all servers.