Instructions for Basic Integration, Mobile311 API v.2.0.

Last Updated 2014-12-22

OVERVIEW

Development Server Web Viewer URL:

https://beta.mobile311.com

Development Server Web Service URL:

https://beta.mobile311.com/ws4/t.asmx

The same user name and password will work for both platforms (note this is not a valid account on the production system):

There is only one web service call ("f") which accepts a single parameter ("json"), which is a JSON formatted command. It emits a standardized XML response which contains JSON data. Please feel free to use the ASMX page provided to test server requests. Your code should be able to POST the "json" to "f".

This format makes testing very easy and independent of your development environment, and standardized commands can be saved in a text file and quickly modified and input into a single field for testing.

All requests require a json encoded sub-parameter (simply called parameters from now on) named "cmd" which is a string representing which server operation to perform. All commands ("cmd") except "signin" also require a security token parameter ("token") to be passed with each request. By default, the token has a 24 hour lifetime, but as this can change in the future, please be sure to check the actual timeout value that is returned by the "signin" function.

All date values are submitted and returned in Microsoft JSON format, which is /DATE(number of milliseconds since 1/1/1970)/. Please note that the forward slashes should be escaped which a backslash, so the final value becomes: \/DATE(number of milliseconds since 1/1/1970)\/

Please note that the JSON encoded parameters and return values are not position dependent, so please utilize coding practice that does not require a precise index or ordering of the values. Please also note that this format allows for us add additional optional parameters and return values within the JSON without breaking the original code signature, so your code should also be written such that the addition of new JSON encoded return parameters will not break your code.

Once everything has been tested, I will send you the appropriate URL for the production site.

All JSON Responses have the following basic signature:

```
<TaskResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"xmlns="http://ws4.mobil
e311.com/">
```

<Data>
{"ClientId":99,"LoginId":844,"UserName":"Demo","Email":"support@mobile311.com
","Token":"nwnbt6lnby8bpfre","TokenExpiration":"\/Date(1397046258843)\/","Mob
ileSecurityLevel":2,"WebSecurityLevel":1,"LastButtonUpdate":"\/Date(139353398
0520)\/"}
</Data>
<SecurityStatus>Success</SecurityStatus>
<HasErrors>false</HasErrors>
<Errors/>
<EllapsedTime>40</EllapsedTime>
</TaskResponse>

<Data/>: Contains the JSON encoded response specific to each request.
<SecurityStatus/>: Contains information regarding the sign on attempt or token related to permissions. Possible Values are: “Unchecked” (for functions that are open to the world), “InvalidUsernamePassword”, “InvalidToken”, “InsufficientPermissions” (token is fine, but user is trying to do something they are not allowed to), and “Success”.
<HasErrors/>: Boolean
<Errors/>: A array of Application Error objects, each of which contain to properties, “ErrorCode” and “Message”
<EllapsedTime/>: How long the server took to process the request, does not take into account any network traffic time. Good for checking performance of requests.

COMMAND REFERENCE

To avoid over complication, I am sending the minimum number of commands you should need for this project. In this case of return values, I will not explain any that are not pertinent. If you have any specific question about additional capability or feel I have left anything out, please do not hesitate to contact me.

Signing in:

Sample JSON Request:

{cmd:'signin',username:'demo',password:'demo'}

Sample JSON Response:

{"ClientId":99,"LoginId":844,"UserName":"Demo","Email":"support@mobile311.com
","Token":"nwnbt6lnby8bpfre","TokenExpiration":"\/Date(1397046258843)\/","Mob
ileSecurityLevel":2,"WebSecurityLevel":1,"LastButtonUpdate":"\/Date(139353398
0520)\/"}

ClientId: The unique id in our system for the client
LoginId: The unique id for this user
UserName: Reflecting the username that was entered
Email: The user’s email in the system (optional)
Token: The token to use for subsequent requests
TokenExpiration: When the token will expire, currently 24 hours from time of request, but could change in the future
MobileSecurityLevel: Security status for certain requests

WebSecurityLevel: When combined with MobileSecurityLevel, determines what access the user has to specific Web API commands.
LastButtonUpdate: The last time there was any configuration change that impacts this user, it can include items such as adding or modifying WorkTypes, Statuses, Logins, etc.

Retrieving Help

Sample JSON Request (remember to get a new token!):

{cmd:"help"} or {cmd:"help", token:"nwnbt6lnby8bpfre"}

You will receive information and a listing of all commands that you can see under your current security context.

You can also add the following parameter to any other command in order to get specific information on a command's parameters: {help:true} For example:

{cmd:"signin", help,true}

Retreiving New and/or Updated WorkItems from Mobile311:

Sample JSON Request (remember to get a new token!):

{cmd:"findworkrequestsbymodifieddate",token:"nwnbt6lnby8bpfre",fromdate:"\/Date(1389846800000)\/",todate:"\/Date(1390000000000)\/",gethistory:"false",getfiles:"false"}

Sample JSON Response:


```
{"WorkRequests":[{"type":"workrequest","requesttitle":"Work Request
Data","workrequestid":2224019,"clientid":99,"clientname":"DemoSite","worktype
id":8405,"worktypename":"Cut
Off","statusname":"Completed","username":"Demo","userid":844,"latitude":35.54
49457634299,"longtitude":-
77.055345135399293,"collecteddate":"\/Date(1389887445600)\/","posteddate":"\/
Date(1389887445600)\/","modifieddate":"\/Date(1389887820537)\/","totalrecords
":1,"address":"139 W 3rd St","city":null,"state":null,"zip":null,"pointx":-
8577762,"pointy":4238186,"authorized":true,"hasphoto":false,"hyperlink":"","c
omments":"","statusid":354,"assetid":null,"assigneduserids":[],"iscodeenforce
ment":false,"issignmaintenance":false,"workgroupname":"","guid":"00000000-
0000-0000-0000-
000000000000","gislayer":null,"color":"#00CCFF","customvalues":null,"priority
":0},{"type":"workrequest","requesttitle":"Work Request
Data","workrequestid":2225639,"clientid":99,"clientname":"DemoSite","worktype
id":8346,"worktypename":"Sign
Down","statusname":"New","username":"DemoSite_CitizenGuest","userid":1286,"la
titude":36.070465945441605,"longtitude":-
79.50881815993155,"collecteddate":"\/Date(1389971310550)\/","posteddate":"\/D
ate(1389971310583)\/","modifieddate":"\/Date(1389971310583)\/","totalrecords"
:1,"address":"1003 Boston Dr","city":null,"state":null,"zip":null,"pointx":-
8850881,"pointy":4310321.5,"authorized":true,"hasphoto":false,"hyperlink":"",
"comments":"","statusid":356,"assetid":null,"assigneduserids":[],"iscodeenfor
cement":false,"issignmaintenance":false,"workgroupname":"","guid":"00000000-
```

```
0000-0000-0000-
000000000000","gislayer":null,"color":"#FF9900","customvalues":null,"priority
":0},{"type":"workrequest","requesttitle":"Work Request
Data","workrequestid":2225712,"clientid":99,"clientname":"DemoSite","worktype
id":8389,"worktypename":"Pot
Hole","statusname":"New","username":"DemoSite_CitizenGuest","userid":1286,"la
titude":36.091752099999411,"longtitude":-
79.436240299998659,"collecteddate":"\/Date(1389974918883)\/","posteddate":"\/
Date(1389974918900)\/","modifieddate":"\/Date(1389974918900)\/","totalrecords
":1,"address":"294 E Davis St","city":null,"state":null,"zip":null,"pointx":-
8842802,"pointy":4313253.5,"authorized":true,"hasphoto":false,"hyperlink":"",
"comments":"","statusid":356,"assetid":null,"assigneduserids":[],"iscodeenfor
cement":false,"issignmaintenance":false,"workgroupname":"","guid":"00000000-
0000-0000-0000-
000000000000","gislayer":null,"color":"#CCCC00","customvalues":null,"priority
":0}],"WorkRequestHistories":[],"Files":[]}
```


This represents an array of WorkItems. The return parameters should be self-explanatory.

Updating an existing WorkItem in Mobile311:

Sample JSON Request:

{cmd:"updateworkrequest",token:"nwnbt6lnby8bpfre",workrequestid:2342973,statusid:356,worktypeid:8340,datemodified:"\/Date(1392941711518)\/",address:"2515 Jefferson Davis Hwy",city:"",state:"",zip:"",longitude:-79.22036588,latitude:35.429346,priority:5,description:"",comment:"Height:8ft",assetid:"",gislayer:""}

The only required JSON parameters are cmd, token, and workrequestid. All others are optional and will retain their current values if omitted. NOTE – Providing an empty string is NOT the same as omitting the parameter and will result (in the case of strings) in the value being updated to an empty string!

So the same command as above for just updating the status would be:

{cmd:"updateworkrequest",token:"nwnbt6lnby8bpfre",workrequestid:2342973,statusid:356"}

In this case, the datemodified will be the server time when the request was received.

Sample JSON Response:


```
{"Result":true}
```


Boolean - In the case of false, please be sure to check the error array.

Inserting a new WorkItem in Mobile311:

Sample JSON Request:

{cmd:"createworkrequest",token:" nwnbt6lnby8bpfre ",longitude:-79.22036588,latitude:35.429346,worktypeid:8340,datecollected:"\/Date(1401712706631)\/"}

The only required JSON parameters are cmd, token, workrequestid. All others are optional and will result in default values if omitted. NOTE – Providing an empty string is NOT the same as omitting the parameter and will result (in the case of strings) in the value being inserted as an empty string!

It is highly recommended that the optional parameters guid, datecollected, latitude, and longitude are always submitted by the client. If the guid is omitted, caching & duplicate detection may be more difficult to implement and if datecollected is omitted then the original date collected will always appear to be 1/1/2001 00:00:00. If the latitude and longitude are omitted, then the work item will default to 0 for both values.

Here is a complete list of parameters on of this document's date:

- guid String (If left blank, this will be randomly assigned by the server.)

- worktypeid Integer (Required)
- citizenrequest Boolean (Is this a citizen request? If so different security requirements apply.)
- datecollected Date (Date this was collected by the remote device, required because request may be cached.)
- addresssource String (Optional value to note where the address came from, typically used to denote a webservice or user confirmed input.)
- address String (The address of work item. If left blank, the server will use the Lat/Long and configured geocode service to determine the address automatically.)
- city String (Additional address field.)
- state String (Additional address field.)
- zip String (Additional address field.)
- longitude Decimal (Required, -180.0 to 180.0)
- latitude Decimal (Required, -90.0 to 90.0)
- priority Integer (The priority value of the work request. If left blank it will use the default priority for the work type. 1 to 9)
- description String (The initial description of the work item. For citizen requests, this is visible to the citizen.)
- comment String (The comment for the work item.)
- uniqueclientid String (Used for linking data from different systems, hidden to the GUI.)
- assetid String (Used for linking to GIS assets, barcodes, etc. This represents the unique id in the other system, though it could be used for any other data as well.)
- gislayer String (Used for linking to GIS, this represents the layer that the GIS asset is on.)
- gpsinfo String (Stores information about the quality of the GPS when used on mobile devices to get the Lat/Long. Hidden from the GUI.)
- programstatus String (Additional device information used for diagnostics. Hidden from the GUI.)
- assignedloginids Integer() (An array of login ids used to assign this work item to the employees in the list.)
- customvalues CustomValue() (An array of custom values object to be passed in. These represent the information needed to populate custom fields on this work item. The structure of a CustomValue is as follows:
    - id Integer (The unique Id for this custom value. Ignored when used in the createworkrequest function as the custom value will not be generated yet, and this will be system assigned.)
    - customfieldid Integer (Required - The id of the custom field to be populated.)

    - workrequestid Integer (The Id of the work request that is linked to this custom value. Ignored when used in the createworkrequest function as the id will be system generated and automatically populated.)
    - booleanvalue Boolean (The value of the custom field if it is a Boolean field.)
    - datevalue Date (The value of the custom field if it is a Date field.)
    - numericvalue Double (The value of the custom field if it is a Numeric field.)
    - textvalue String (The value of the custom field if it is a Text field.)
- attachedemployees AttachedEmployee() (An array of attached employee objects for time keeping, currently in under development status.)
- attachedmaterials AttachedMaterial() (An array of attached material objects for expense/inventory purpose, currently in under development status.)
- attachedequipment AttachedEquipment() (An array of attached equipment objects for expense purpose, current in under development status.)
- firstname String (The first name of a citizen making a request.)
- lastname String (The last name of a citizen making a request.)
- email String (The email of a citizen making this request.)
- phonenumber String (The phone number of a citizen making this request.)
- fileguid String (The GUID of a previously submitted photo that should be attached to this request.)

Sample JSON Response:

{"Id":2347886,"Guid":"4c4d05e3-31a6-46f9-8167-9e6d5d32a05c"}

The System assigned ID will be returned, along with the GUID, which may be system generated, or preferably, Client generated, in which case it will reflect back the same value.

WorkFlow & Design Considerations:

The largest consideration when retrieving work items the amount of data that can be retrieved in a single call. A balance must be struck between “chatty” and “chunky” data requests. This window is going to vary based on the data being retrieved. When retrieving only the work items themselves, a larger request window can be utilized. When retrieving history and especially when retrieving attached files, it is possible to run into two common errors. The first one is Error Code “00010007” “Json text exceeded set limits.” In this case you attempted to download more than 100 MB of JSON data in the request. Please reformulate the request into a smaller chunk. The second is “00010008” “Database connection error or timeout.” This error is much less likely (especially in production), but may occur if a large request is made for the first time on the beta server and SQL needs to take some extra time to spin up the data required for the query. To this end, we recommend retrieving and processing the data in increments of 1 hour or less (for work items only) or 15 minutes or less (for work items plus photos).

To implement this, specify the “fromdate” as well as the “todate” for the command and run all processing in a loop incrementing the “fromdate” and “todate” to cover the specified timespan. This loop should cease once the current time is reached. Please note that our database stores the data down to the millisecond level, so the next increment from “1/1/2001 12:00:00.000” would be “1/1/2001 12:00:00.001”.

Please note the following important details that impact the “findworkrequestsbymodifieddate” function.

1. This function retrieves all work items whose current modifieddate is within the requested timespan.
2. When NEW work items are entered into the database, their modified date is initially set to the date that the work item is inserted into the database ("posteddate").
3. When a work item is updated due to field changes or a new file being attached, its modifieddate is updated.
4. There are no other query parameters for filtering the results. (You cannot request just certain statuses for example.)

Takeaways:

1. This single function is used for retrieving both new and updated work items.
2. There is NOT a single field that determines if the request is new or updated.
    a. You can compare the "posteddate" to the "modifeddate". If these are the same, it means the work item has not been modified and is "new". If they are different, then it is certainly an update.
3. The only certain way to avoid inserting duplicate work requests into your system is to utilize the work item ID field.

Please be aware that there will be times when we are performing maintenance on the system (usually late night/weekends) and the web service may be unavailable. In these cases please make sure your program incorporates appropriate error trapping and a wait period of approximately 5 minutes before making retries.

It will be up to you to determine the best way to keep up the sync list, but in general to avoid possible redundant web calls, we recommend implementing a local sync-database. When requests are pulled from our system, cache them locally before pushing them into your system and vice-versa so data does not have to be re-requested over the wire. If you choose to keep your system ID in Mobile311, please use the UniqueClientId field. We strongly recommend keeping the Mobile311 Id in your system, so that you do not have to do any kind of lookup in Mobile311 prior to updating the status. If for some reason you do need the function to retrieve a work item by its id, I will be glad to provide it. We also recommend storing the datetime of the last successfully synced item so that the next code run knows where to start from.

The last consideration to take into account is that we have very recently introduced multiple time-zone support. If you notice anything unusual as it relates to anything being off by an hour, please let me know. ALL NEW DATA ENTERED SHOULD BE CORRECT, HOWEVER WE DID NOT UPDATE THE HISTORICAL INFORMATION AS PART OF THE PATCH, SO THE MODIFIEDDATE WILL NOT MATCH THE POSTEDDATE FOR THE BULK OF THE HISTORICAL DATA. This should not affect clients in the Eastern Standard/Daylight Savings Timezone.