

Baker Real Estate

Baker PMS API

API Specification

Version 0.8

19 September 2014

Frank Li  
frank.li@qualicom.com

Qualicom Internal Use Only

This document contains information that is the property of Qualicom Innovations Inc. Reproduction, disclosure or any other use of the proprietary information in this document is prohibited without written permission from Qualicom Innovations Inc. Product names used herein are for identification purposes only and may be trademarks of their respective companies.

Copyright © 2014. All rights reserved.

##### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 2014-07-10 | Frank Li | * Initial draft. |
| 0.2 | 2014-07-18 | Karen Bennet | * Updates based on document review meeting. |
| 0.3 | 2014-08-27 | Frank Li | * Changed Project to Phase. * Partial phase name accepted. * Model name and Phase ID in Get Units. * Removed unittype, pricea, priceb from Get Phases. |
| 0.4 | 2014-08-28 | Frank Li | * Changed "availability" to "status" in Get Units. * Added new properties to Get UserProfiles. * Expanded sample response in Get UserProfiles. * Changed alternatephone type from string to array. |
| 0.5 | 2014-09-09 | Frank Li | * Added Floor parameter to Get Units. * Limited status for Get Units to Available and Not Available. * Added note on embedded spaces in "About the API". |
| 0.6 | 2014-09-11 | Andre Riolo | * Added Sample Call and Data Structure to 2.2/PUT; for Worksheet and Users |
| 0.7 | 2014-09-11 | Steve Chalastra | * Reformatted into appropriate calls. |
| 0.8 | 2014-09-19 | Steve Chalastra | * Clarifications after call with Rad. |

Contents

[1 About the API 4](#_Toc398909790)

[1.1 Authentication 4](#_Toc398909791)

[2 API Methods 5](#_Toc398909792)

[2.1 GET /userprofiles 5](#_Toc398909793)

[2.2 PUT/userprofiles 8](#_Toc398909794)

[2.3 GET /phases 8](#_Toc398909795)

[2.4 GET /models 11](#_Toc398909796)

[2.5 GET /units 13](#_Toc398909797)

[2.6 PUT/worksheets 15](#_Toc398909798)

# About the API

This document outlines the Baker PMS Web Services API, a collection of HTTP interfaces to Baker Web Services that enables a programmer to read and write data from and to the Baker registry. Operations such as lookups or searches for users, phases, models and units of work can be performed using this API. Additional functionality may be developed and deployed as needed by Baker.

To be able to use this API, users should be registered as authorized users.

These services produce data exclusively in JSON format. Client applications should parse JSON responses using a JSON parser library to address elements in the appropriate format.

Only one parameter value is supported for each property.

All GET methods return a representation in JSON format and an HTTP response code of 200 (OK). In the event of errors, either 404 (NOT FOUND) or 400 (BAD REQUEST) are returned.

All PUT methods return an HTTP response code of 201 (CREATED). In the event of errors, either @@ to be completed.

## Authentication

To be able to access the Baker API, all users must authenticate using the HTTP Basic authentication method with their Baker API user account; otherwise, they will receive a 401 (AUTHORIZATION REQUIRED) response code.

The required HTTP Authorization header should have the following form:

#Request

Authorization: apiSignatureValue

where apiSignatureValue is obtained by computing a base-64 encoding of the username and password.

apiSignatureValue = Base64 (api\_username:api\_password)

Note: User names and passwords are case sensitive. Contact Qualicom Innovations for values.

# API Methods

## GET /userprofiles

Retrieves all user profiles matching the specified type.

##### URL

https://www.bakersales.info/BakerPublic/api/userprofiles

##### Parameters

* type:
* 11 – Broker: Allocation BRP
* 12 – Broker: General BRP
* 13 – Broker: Baker Agent
* 14 – Broker: Standard Agent
* 21 – Customer: Purchase
* 22 – Customer: eWorksheet
* 23 – Customer: Standard Account
* 31 – Developer: Full Access Developers
* 32 – Developer: Limited Access Developers
* 41 – Admin: Super User
* 42 – Admin: Publisher
* 43 – Admin: Author
* firstname
* lastname
* email

##### Notes

* Existing Types – reuse existing definition in Qualicom’s database for Customer, Broker and Developer
* New Types:

11 – Broker – Allocation BRP – roles and permission definition for what content can be viewed/updated

31 – Developer: Full Access – roles and permission definition for what content can be viewed/updated

32 - Developer: Limited Access – TBD the information rules

42 – Admin : Publisher – roles and permission for what content that can be viewed/updated

43 – Admin : Author – roles and permission for what content that can be viewed/updated

##### Sample call

[https://www.bakersales.info/BakerPublic/api/userprofiles?id=001](https://www.bakersales.info/api/userprofiles?type=11&id=001)

https://www.bakersales.info/BakerPublic/api/userprofiles/001

##### Sample response

Note throughout entire document: standardization on "licence" rather than "license". I already used license in our model, not sure if it makes sense to make the change or not…

{

"id": "001", "firstname": "Jo", "lastname": "Lee", "email": jo.lee@test.com", "brokeragename": "Sutton", "country": "20", "province": "2009", "city": "Toronto", "address": "1 Main St", "address1": "", "postalcode": "M1A B2C", "password": "", "licencenumber": "", "optin": "1", "mainphone": "4161234567", "alternatephone": "", "dob": "1970-12-01", "sinnumber": "", "driverlicencenumber": "", "driverlicenceexpirydate": "", "fintrac": "", "maritalstatus": "", "type": "21", "usercategory": "Customer"

}

##### Data structure

{

"type": "object",

"properties": {

"id": {"type": "string"},

"firstname": {"type": "string"},

"lastname": {"type": "string"},

"email": {"type": "string"},

"brokeragename": {"type": "string"},

"country": {"type": "string"},

"province": {"type": "string"},

"city": {"type": "string"},

"address": {"type": "string"},

"address1": {"type": "string"},

"postalcode": {"type": "string"},

"password": {"type": "string"},

"licencenumber": {"type": "string"},

"optin": {"type": "string"},

"mainphone": {"type": "string"},

"alternatephone": {"type": "array"},

"dob": {"type": "string"},

"sinnumber": {"type": "string"},

"driverlicencenumber": {"type": "string"},

"driverlicenceexpirydate": {"type": "string"},

"fintrac": {"type": "string"},

"maritalstatus": {"type": "string"},

"type": {"type": "string"},

"usercategory": {"type": "string"}

}

}

## PUT/userprofiles

Updates a specific user profile if it exists; if not, a new profile is created.

##### URL

https://www.bakersales.info/BakerPublic/api/userprofiles

##### Data structure

(How Union will send it)

{

"type": "object",

"properties": {

"first\_name": {"type": "string"},

"last\_name": {"type": "string"},

"email": {"type": "string"},

"brokerage": {"type": "string"},

"other\_brokerage": {"type": "string"},

"country": {"type": "string"},

"province": {"type": "string"},

"city": {"type": "string"},

"address": {"type": "string"},

"postal\_code": {"type": "string"},

"reco\_no": {"type": "string"},

"is\_marketing": {"type": "string"},

"phone\_number": {"type": "string"},

"alt\_phone\_number": {"type": "array"},

}

}

## GET /phases

Retrieves all phase profiles matching the specified type.

##### URL

https://www.bakersales.info/BakerPublic/api/phases

##### Parameters

* name (can be a partial string)
* status:
* 111 – Pre-Construction
* 112 – Under Construction
* 113 – Post-Construction
* 114 – On Hold
* 115 – Sold Out
* 120 – Pre-Opening
* country:
* 20 – Canada
* 31 – United States of America
* province:
* 2001 – Alberta
* 2002 – British Columbia
* 2003 – Manitoba
* 2004 – New Brunswick
* 2005 – Newfoundland and Labrador
* 2006 – Northwest Territories
* 2007 – Nova Scotia
* 2008 – Nunavut
* 2009 – Ontario
* 2010 – Prince Edward Island
* 2011 – Quebec
* 2012 – Saskatchewan
* 2013 – Yukon
* latitude
* longitude
* radius

##### Sample call 1

https://www.bakersales.info/BakerPublic/api/phases?province=2009&status=111

##### Sample response 1

{

"phaseprofiles": [

{"projectid": "1862637", phaseid": "1862637", "name": "Canary District South", "website": "http://www.canarydistrict.com", "status": "Post-Construction"},

{"projectid": "839487", "phaseid": "839487", "name": "MYC Condos", "website": "http://myccondo.com/", "status": "Post-Construction"},

…

]}

##### Sample call 2

https://www.bakersales.info/BakerPublic/api/phases?Phaseid=839487

https://www.bakersales.info/BakerPublic/api/phases/839487

##### Sample response 2

{

"projectid": "839487", "phaseid": "839487", "name": "MYC Condos", "website": "http://myccondo.com/", "mainintersection": "N/A", "country": "CANADA", "province": "Ontario", "city": "Toronto", "address": "175 Bloor Street", "postalcode": "M4W 3R8", "latitude": "41.12", "longitude": "-79.90", “radius”: ”50”, "notes": "notes field", "status": "Post-Construction ", "currency": "CAD","currencysymbol": "CAD", "siteadmincontact": "Allison", "emailstosendonlinesales": "Allison@baker-re.com", "toatalnumofparkingspots": "2", "totalnumofstoragelockers": "2","siteadmincontact": "N/A"

}

##### Data structure

{

"type": "object",

"properties": {

"projectid": {"type": "string"},

"phaseid": {"type": "string"},

"name": {"type": "string"},

"website": {"type": "string"},

"mainintersection": {"type": "string"},

"country": {"type": "string"},

"province": {"type": "string"},

"city": {"type": "string"},

"address": {"type": "string"},

"postalcode": {"type": "string"},

"latitude": {"type": "string"},

"longitude":{"type": "string"},

“radius”:{type: “string”],

"notes": {"type": "string"},

"status": {"type": "string"},

"currency": {"type": "string"},

"currencysymbol": {"type": "string"},

"siteadmincontact": {"type": "string"},

"emailstosendonlinesales": {"type": "string"},

"toatalnumofparkingspots": {"type": "string"},

"totalnumofstoragelockers": {"type": "string"},

"siteadmincontact": {"type": "string"}

}

}

## GET /models

Retrieves all model profiles matching the specified type.

##### URL

https://www.bakersales.info/BakerPublic/api/models

##### Parameters

* phaseid
* projectid
* modelid
* areainsquarefeeta
* areainsquarefeetb
* bedroomsa
* bedroomsb
* bathroomsa
* bathroomsb

##### Notes

1. areainsquarefeeta and areainsquirefeetb enables ability to specify a range; a and b are inclusive
2. beroomsa and bedroomsb enables ability to specify a range; a and b are inclusive
3. bathroomsa and bathroomb enables ability specify a range; a and b are inclusive

##### Sample call

https://www.bakersales.info/BakerPublic/api/models?modelid=2655591

https://www.bakersales.info/BakerPublic/api/models/2655591

##### Sample response

{

"modelid": "2655591", "phaseid": "2455567", "modelname": "model2", "legalname": "model2", "direction": "sw", areainsquarefeet": "1000", "bedrooms": "2","bathrooms": "1.5", "notes": "this is note field"

}

##### Data structure

{

"type": "object",

"properties": {

"modelid": {"type": "string"},

"phaseid": {"type": "string"},

"modelname": {"type": "string"},

"legalname": {"type": "string"},

"direction": {"type": "string"},

"areainsquarefeet": {"type": "string"},

"bedrooms": {"type": "string"},

"bathrooms": {"type": "string"},

"notes": {"type": "string"}

}

}

## GET /units

Retrieves all unit profiles matching the specified type.

##### URL

https://www.bakersales.info/BakerPublic/api/units

##### Parameters

* phaseid
* modelid
* modelname
* unitid
* floor
* floora
* floorb
* status
* 1110 – Available
* 1130 – Not Available
* pricea
* priceb
* parkingspotsa
* parkingspotsb
* lockersa
* lockersb

##### Notes

1. floora and floorb enables ability to specify a range; a and b are inclusive
2. pricea and priceb enables ability to specify a range; a and b are inclusive
3. parkingspota and parkingspotb enables ability to specify a range; a and b are inclusive
4. lockersa and lockersb enables ability to specify a range; a and b are inclusive

##### Sample call

https://www.bakersales.info/BakerPublic/api/units?unitid=2455997

https://www.bakersales.info/BakerPublic/api/units/2455997

##### Sample response

{

"unitid": "2455997", "unitnumber": "01", "modelid": "2455590", "modelname": "penthouse", "phaseid": "338", "floor": "1", "direction": "s/w", "areainsquarefeet": "1500", "notes": "this is another note field", "baseprice": "350000", "mintenancefee": "200", "parkingspotsincluded": "1", "storagelockersincluded": "2","costofadditionalparking": "1000", "costofadditionalstorage": "1500", "maxparkingspots": "1", "maxstoragelockers": "2", "status": "available", "suitenumber": "101", "ceilingheightinfeet": "9", "legalsuitenumber": "9"

}

##### Data structure

{

"type": "object",

"properties": {

"unitid": {"type": "string"},

"unitnumber": {"type": "string"},

"modelid": {"type": "string"},

"modelname": {"type": "string"},

"phaseid": {"type": "string"},

"floor": {"type": "string"},

"direction": {"type": "string"},

"areainsquarefeet": {"type": "string"},

"notes": {"type": "string"},

"baseprice": {"type": "string"},

"mintenancefee": {"type": "string"},

"parkingspotsincluded": {"type": "string"},

"storagelockersincluded": {"type": "string"},

"costofadditionalparking": {"type": "string"},

"costofadditionalstorage": {"type": "string"},

"maxparkingspots": {"type": "string"},

"maxstoragelockers": {"type": "string"},

"status": {"type": "string"},

"suitenumber": {"type": "string"},

"ceilingheightinfeet": {"type": "string"},

"legalsuitenumber": {"type": "string"}

}

}

## PUT/worksheets

@@ TBD how the information will be sent – assume three data structures for now. Frank to investigate upon getting access to Union's staging server.

Creates sales information in PMS based on worksheet submission.

##### URL

https://www.bakersales.info/BakerPublic/api/worksheets

##### Data structure: Worksheet

{

"type": "object",

"properties": {

"status": {"type": "string"}, // ignore

"selected\_unit": {"type": "string"}, // unit ID

"end\_user\_investor": {"type": "string"},

"profession": {"type": "string"},

"marital\_status": {"type": "string"},

"register\_online": {"type": "string"},

"nationality": {"type": "string"},

"dependents": {"type": "string"},

"dependents\_ages": {"type": "string"},

"hear\_about\_us": {"type": "string"},

"income\_range": {"type": "string"}, @@ Do we need to add this? Sean to clarify.

}

##### }Data structure: Agent

{

"type": "object",

"properties": {

"first\_name": {"type": "string"},

"last\_name": {"type": "string"},

"email": {"type": "string"},

"brokerage": {"type": "string"},

"country": {"type": "string"},

"province": {"type": "string"},

"city": {"type": "string"},

"address": {"type": "string"},

"suite": {"type": "string"},

"postal\_code": {"type": "string"},

"reco\_no": {"type": "string"},

"phone\_number": {"type": "string"},

"alt\_phone\_number": {"type": "string"},

"is\_marketing": {"type": "string"}, // optin

}

}

##### Data structure: Purchaser

{

"type": "array",

"properties": {

"first\_name": {"type": "string"},

"last\_name": {"type": "string"},

"email": {"type": "string"},

"province": {"type": "string"},

"city": {"type": "string"},

"address": {"type": "string"},

"postal\_code": {"type": "string"},

"phone\_number": {"type": "string"},

"alt\_phone\_number": {"type": "string"},

"date\_of\_birth": {"type": "string"},

"sin": {"type": "string"},

"drivers\_license": {"type": "string"},

"drivers\_expiry": {"type": "string"},

"suite": {"type": "string"},

}

}