Glide API Documentation

Glide Utilities Ltd.

EXT/CUST/API-DOCS
Version 2.001, July 2, 2013

Contents

1 Introduction **2**
- 1.1 Purpose . . . 2
- 1.2 Intended audience and reading suggestions . . . 2
- 1.3 Assumptions . . . 2
- 1.4 API key . . . 2
- 1.5 Document Change History . . . 2

2 Consuming API methods **3**
- 2.1 Endpoints . . . 3
- 2.2 Payloads . . . 3
- 2.3 Example output . . . 3

3 Signup methods **5**
- 3.1 address . . . 5
  - 3.1.1 searchPremiseByPostcode . . . 5
- 3.2 referralHouse . . . 5
  - 3.2.1 complete . . . 5
- 3.3 quote . . . 5
  - 3.3.1 allServices . . . 6

Glossary **7**

1 Introduction

1.1 Purpose

The purpose of this document is to provide instructions to customers and partners for using RESTful web services provided through the Glide website.

1.2 Intended audience and reading suggestions

This document is intended to be read by the following parties. All sections should be read by all parties unless otherwise noted in brackets.

- Glide software manager.
- Software development and other technical personnel.
- Technical employees and contrators of Glide clients (sections pertinent to the services to which access has been granted.)

1.3 Assumptions

This reference is a technical document designed for software developers. It assumes familiarity with web-related technologies such as HTTP, REST, XML amd JSON.

1.4 API key

In order to consume any Glide web service you will require an API key. This is a string which is issued on a per-company basis. The access key grants access only to services to which your company has been granted access. You may not have access to all services included in this document. Please ask your Glide account manager if you do not have an API key[1].

1.5 Document Change History

Date	Version	Author	Remarks
20/10/2011	1.000	SBF	Initial document setup.
20/10/2011	1.001	SBF	First draft.
21/10/2011	2.000	SBF	First published version.
25/06/2013	2.001	SBF	Overhaul following launch of Website 3.5.

Table 1: Document change history

[1] Glide developers should note that Glide itself also has such an access key; we are treated no differently by the API.

2 Consuming API methods

2.1 Endpoints

The API base endpoint address is as follows:

`https://www.glide.uk.com/api/4.0/`

Methods specified the following sections are then appended to this endpoint as show in the following example, which will return JSON-formatted output. Replace "`.json`" with "`.xml`" if XML is preferred.

```
https://www.glide.uk.com/api/4.0/signUp/address/searchPremiseByPostcode.json
```

2.2 Payloads

Data is sent to the API in the body of a POST request as a JSON-formatted payload. Each payload must as a minimum carry your API key:

`{"key":"0123456789abcdef0123456789abcdef"}`

Other data can then be passed in the payload as required by the method in question, for example:

`{"key":"0123456789abcdef0123456789abcdef", "postcode":"B1 1TT"}`

2.3 Example output

The following sample is output from the example method and payload above:

```
[
  {
    "udprn":"00327632",
    "address":"Glide, Alpha Tower, Suffolk Street Queensway, Birmingham",
    "address1":"Alpha Tower",
    "address2":"Suffolk Street Queensway ",
    "town":"Birmingham",
    "postcode":"B1  1TT"
  },
  {
    "udprn":"52825989",
    "address":"Roar, Alpha Tower, Suffolk Street Queensway, Birmingham",
    "address1":"Alpha Tower",
    "address2":"Suffolk Street Queensway ",
    "town":"Birmingham",
    "postcode":"B1  1TT"
  }
]
```

Listing 1: Output from JSON method

```
<?xml version="1.0" encoding="UTF-8"?>
<response>
  <item key="0">
    <udprn>00327632</udprn>
    <address>Glide, Alpha Tower, Suffolk Street Queensway, Birmingham</address>
    <address1>Alpha Tower</address1>
    <address2>Suffolk Street Queensway</address2>
    <town>Birmingham</town>
    <postcode>B1  1TT</postcode>
  </item>
  <item key="1">
```

```
12      <udprn>52825989</udprn>
13      <address>Roar, Alpha Tower, Suffolk Street Queensway, Birmingham</address>
14      <address1>Alpha Tower</address1>
15      <address2>Suffolk Street Queensway</address2>
16      <town>Birmingham</town>
17      <postcode>B1  1TT</postcode>
18    </item>
19 </response>
```

Listing 2: Output from XML method

3 Signup methods

Base endpoint for all methods in this section:

```
https://www.glide.uk.com/api/4.0/signUp/
```

3.1 address

Base endpoint for all methods in this section:

```
https://www.glide.uk.com/api/4.0/signUp/address/
```

3.1.1 searchPremiseByPostcode

Searches for addresses matching a given postcode.

Parameter	Description
`postcode`	A valid United Kingdom postcode.

3.2 referralHouse

Base endpoint for all methods in this section:

```
https://www.glide.uk.com/api/4.0/signUp/referralHouse/
```

3.2.1 complete

Processes a referral house signup.

Parameter	Description
`referralId`	Referral ID that we provided when you signed up as a referral landlord.
`udprn`	UDPRM of the house to add, from the `searchPremiseByPostcode` method.
`startDate`	Start date of the property (YYYYMMDD).
`source`	Software source of the signup (for example your application name).
`firstname1`	The first tenant's first name.
`surname1`	The first tenant's surname.
`email1`	he first tenant's e-mail address.
`mobile1`	The first tenant's mobile number.

The "firstname", "surname", "email" and "mobile" parameters can be repeated up to twelve times, incrementing the number each time, to represent up to twelve tenants living in the property.

3.3 quote

Base endpoint for all methods in this section:

```
https://www.glide.uk.com/api/4.0/signUp/quote/
```

3.3.1 allServices

Obtain an all-service quote for a given property.

Parameter	Description
`capacity`	Capacity of the house (1-12).
`minTerm`	Term of the house contract in months (6-36).
`gas`	Set to “true” to quote for gas, else dont specify or set to “false”.
`electricity`	Set to “true” to quote for electricity, else dont specify or set to “false”.
`water`	Set to “true” to quote for water, else dont specify or set to “false”.
`telephone`	Set to “true” to quote for telephone, else dont specify or set to “false”.
`broadband`	Set to “true” to quote for broadband, else dont specify or set to “false”.
`tv`	Set to “true” to quote for tv licence, else dont specify or set to “false”.
`postcode`	Valid UK postcode, required for water quote.
`broadbandType`	Required if ordering broadband. Pass “llu24s” for standard or “llu24p” for premium.

Glossary

GET is an HTTP method which requests a representation of the specified resource. The most common usage of this method is to retrieve a web page for rendering in a web browser and it is it the most commonly used HTTP method. 7

HTTP (Hypertext Transfer Protocol) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the Web. 2, 7

JavaScript is a client-side dynamic scripting language in use by all major web browsers, implemented in order to provide enhanced user interfaces and create dynamic websites. 7

JSON (JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for most languages. 2

POST is one of many request methods supported by the HTTP protocol used by the World Wide Web. The POST request method is used when the client needs to send data to the server as part of the request, such as when uploading a file or submitting a completed form. In contrast to the GET request method where only a URL and headers are sent to the server, POST requests also include a message body. This allows for arbitrary length data of any type to be sent to the server. 3

REST (Representational State Transfer) is a model for web services based solely on HTTP. REST takes the view that the Web already has everything necessary for web services, without having to add extra specifications like SOAP and UDDI. Any item can be made available (ie represented) at a URI, and, subject to the necessary permissions, it can be manipulated using one of the simple operations defined within HTTP (GET to retrieve information, PUT and POST to modify it, DELETE to remove it). REST is a more modern method of implementing web services than SOAP and is better supported by modern APIs and SDKs. 2

XML (Extensible Markup Language) is a set of rules for encoding documents and data in a machine-readble format. Uses of XML are diverse and varied, but it is commonly used to exchange information between third party systems such as payment gateways and remote data sources. 2