Record APIs - Record Types

Record Types

Data for most types of *Object* in the platform can be exposed via Record APIs (subject to a couple of basic restrictions listed in Getting Started). Each *Object* will correspond to a *Record Type* in the Record APIs.

API Naming Conventions

The platform places few, if any, restrictions on the naming of objects, fields, and relationships. In order to reliably use these as part of APIs, however, some basic rules are used to create "safe" equivalents for these names.

> It is the *Script Name* of an *Object*, *Field*, or *Relationship* that is used in APIs.

Names in Paths

When used as part of a path (i.e. URL), names are converted to "snake case" (e.g. "Hello, BeautifulWorld!" becomes "hello-beautiful-world").

1. Names are split into chunks wherever there is whitespace or an upper-case letter followed by a lower-case letter
2. Each chunk's letters are converted to lower-case
3. The chunks are joined together again using the "-" character

Names in JSON

When used as in JSON (e.g. to represent the value of a field or relationship), names are converted to "camel case" (e.g. "Hello, BeautifulWorld!" becomes "helloBeautifulWorld").

- Names are split into chunks wherever there is whitespace or an upper-case letter followed by a lower-case letter
- The first letter of each chunk (except the first one) is converted to upper-case
- The remaining letters of each chunk (including the first one) are converted to lower-case
- The chunks are then directly joined together

As a general rule, if you're not sure of the naming of a *Record Type*, *Field*, or *Relationship*, take a look at the Swagger Sandbox to see details of the expected request / response format(s).

API Forms

When Record APIs are enabled for an Object, an *API Form* is created. This is a special type of *Edit Form* whose controls describe the fields and relationships exposed via the Object's Record API.

In other words, the API Form defines the schema for its Object's Record APIs.

> Note: fields and/or relationships that are marked as mandatory at the object level *must* be included on the API form, or you will not be able to create new records.

Preview APIs

Preview APIs are "live" reflections of the current system configuration. If you make changes to an Object's API Form, they are immediately reflected in the API.

The general URL pattern for Preview APIs is:

```
https://{domain}.readinow.com/api/v1/preview/records/{record-type-name}/{record-type-revision-number}
```

Preview APIs are for use during the API-design phase, and are not supported for production usage.

Published APIs

Published APIs are a "snapshot" of the current system configuration. They are produced by publishing (or re-publishing) an existing preview API. If you make changes to an Object's API Form, they will not be reflected in the published API until you re-publish.

```
https://{domain}.readinow.com/api/v1/records/{record-type-name}/{record-type-revision-number}
```

Record-type Revision Numbers

Each time a record type is published, its revision number is incremented. This means that publishing changes to a record type will change the URL (slightly) and clients will need to be updated to match. This behaviour is by design as modifying the API usually means clients will need to be updated anyway.