

SwiftRiver REST API

The rest API is the interface to SwiftRiver data that allows applications to programmatically access and manage drops, rivers, buckets and accounts. This reference document is designed for those interested in consuming data or publishing data into SwiftRiver.

General Notes

Versioning

The current version of the API is 1.

Strings

All strings are UTF-8 encoded.

Dates

Dates are RFC822 encoded as shown below:

```
"Wed, 27 Aug 2008 13:08:45 +0000"
```

Authentication

The API uses OAuth 2.0 for authentication and authorisation. See [API Authentication](#) for details.

Rate Limiting

Access to all API end points is rate limited. The default limits are set to levels sufficient for common use cases.

See [Best Practices](#) to avoid common causes of rate limiting.

Partial Responses

By default, the server sends back the full representation of a resource after processing requests. For better performance, you can ask the server to send only the fields you really need and get a partial response instead.

To request a partial response, use the fields request parameter to specify the fields you want returned. You can use this parameter with any request that returns a response body.

Example

TODO: Show getting a drop with partial response

Error Handling

Errors are returned using standard [HTTP status codes](#) with any additional information given in a JSON formatted response body. Unless otherwise stated, below are the error codes returned by the API:

Code	Description
400	Bad input parameter. Error message should indicate which one and why.
401	Unauthorized. Try re-authentication or you might not have access to the resource.
403	Access was denied. Re-authentication will not help.
404	The requested resource was not found.

405	Method not allowed for the requested resource.
503	Rate limited. You are making too many calls to the given resource above you account limit.
5xx	Server error. There is a problem with the API, try again later.

Error Response Body

The response body is a JSON formatted object with the following fields:

Field	Type	Description
message	String	Human readable error message.
errors	Errors	Only present for bad input parameters errors and gives details about the field(s) in error.

Error Fields

Field	Type	Description
field	String	The name of the field in error.
error_code	String	Either "missing", "invalid" or "duplicate"

Above error codes have the following meanings:

- missing: The required field was not sent in the request.
- invalid: The specified field is not formatted correctly.
- duplicate: Another resource has the same value for the specified key field.

Sample Error Response

```
{
  "message": "Duplicate river name",
  "errors": [
    {
      "field": "name",
      "code": "duplicate"
    }
  ]
}
```

Data Formats

All data is sent and received as **JSON**.

Core Concepts

- [API Authentication](#)
- [Platform Objects](#)
- [API Resources](#)
- [Best Practices](#)