



Socrata was acquired by Tyler Technologies in 2018 and is now the Data and Insights division of Tyler. The platform is still powered by the same software formerly known as Socrata but you will see references to Data & Insights going forward.

[Learn more...](https://www.tylertech.com/solutions/transformativetechnology/data-insights)

(<https://www.tylertech.com/solutions/transformativetechnology/data-insights>)

Data Transform Listing

Overview

[API Endpoints \(/docs/endpoints\)](/docs/endpoints)
[Row Identifiers \(/docs/row-identifiers\)](/docs/row-identifiers)
[RESTful Verbs \(/docs/verbs\)](/docs/verbs)
[Application Tokens \(/docs/app-tokens\)](/docs/app-tokens)
[Authentication \(/docs/authentication\)](/docs/authentication)
[Response Codes & Headers \(/docs/response-codes\)](/docs/response-codes)
[System Fields \(/docs/system-fields\)](/docs/system-fields)
[CORS & JSONP \(/docs/cors-and-jsonp\)](/docs/cors-and-jsonp)

Filtering & Querying

[Simple Filters \(/docs/filtering\)](/docs/filtering)
[SoQL Queries \(/docs/queries/\)](/docs/queries/)
[Paging Through Data \(/docs/paging\)](/docs/paging)
[SoQL Function and Keyword Listing \(/docs/functions/\)](/docs/functions/)
[Data Transform Functions \(/docs/transforms/\)](/docs/transforms/)

Data Formats (/docs/formats/)

[JSON \(/docs/formats/json\)](/docs/formats/json)
[GeoJSON \(/docs/formats/geojson\)](/docs/formats/geojson)
[CSV \(/docs/formats/csv\)](/docs/formats/csv)
[RDF-XML \(/docs/formats/rdf-xml\)](/docs/formats/rdf-xml)

Datatypes (/docs/datatypes/)

[Checkbox \(/docs/datatypes/checkbox\)](/docs/datatypes/checkbox)
[Fixed Timestamp \(/docs/datatypes/fixed_timestamp\)](/docs/datatypes/fixed_timestamp)
[Floating Timestamp \(/docs/datatypes/floating_timestamp\)](/docs/datatypes/floating_timestamp)
[Line \(/docs/datatypes/line\)](/docs/datatypes/line)
[Location \(/docs/datatypes/location\)](/docs/datatypes/location)
[MultiLine \(/docs/datatypes/multiline\)](/docs/datatypes/multiline)
[MultiPoint \(/docs/datatypes/multipoint\)](/docs/datatypes/multipoint)
[MultiPolygon \(/docs/datatypes/multipolygon\)](/docs/datatypes/multipolygon)
[Number \(/docs/datatypes/number\)](/docs/datatypes/number)
[Point \(/docs/datatypes/point\)](/docs/datatypes/point)
[Polygon \(/docs/datatypes/polygon\)](/docs/datatypes/polygon)
[Text \(/docs/datatypes/text\)](/docs/datatypes/text)
[URL \(/docs/datatypes/url\)](/docs/datatypes/url)

Other APIs (/docs/other/)

These are the transformation functions available in the Dataset Management API (/publishers/dsmapi). These functions can be used to transform and validate your data before you publish your dataset for consumption.

These functions can be used in the “Data Transforms” editor of the the Dataset Management Experience (<https://support.socrata.com/hc/en-us/articles/115016067067-Using-the-Socrata-Data-Management-Experience>) interface. Check out some of the examples on our Support Portal here (<https://support.socrata.com/hc/en-us/articles/360034530954-Data-Transformation-Examples>)!

See the Dataset Management API docs (</docs/other/publishing>) for more info on how to use the transform functions as an API user.


Function Name	Description
+ (/docs/transforms/add)	Keep a number's sign
and (/docs/transforms/and)	Logical and of two boolean values
(/docs/transforms/concatenate)	concatenate two strings
/ (/docs/transforms/divide)	Divide a number by another
= (/docs/transforms/equal)	Return true if the left side equals the right
== (/docs/transforms/equal)	Return true if the left side equals the right
^ (/docs/transforms/exponent)	No documentation is available.
> (/docs/transforms/greater_than)	Return true if the value on the left is greater than the value on the right
>= (/docs/transforms/greater_than_equal)	Return true if the value on the left is greater than or equal to the value on the right
< (/docs/transforms/less_than)	Return true if the value on the left is less than the value on the right
<= (/docs/transforms/less_than_equal)	Return true if the value on the left is less than or equal to the value on the right
% (/docs/transforms/modulo)	Find the remainder(modulus) of one number divided by another
* (/docs/transforms/multiply)	Multiply two numbers together
not (/docs/transforms/not)	Invert a boolean
<> (/docs/transforms/not_equal)	Return true if the left side does not equal the right
!= (/docs/transforms/not_equal)	Return true if the left side does not equal the right
or (/docs/transforms/or)	Logical or of two boolean values
- (/docs/transforms/subtract)	Subtract a number from another
abs (/docs/transforms/abs)	Produce the absolute value of a number
between (/docs/transforms/between)	Return true if the left is within the range of the right values
case (/docs/transforms/case)	Evaluate a series of true/false expressions (predicates) and return the next consequent.
centroid (/docs/transforms/centroid)	returns the geometric centroid of a polygon or multipolygon. Please refer to
coalesce (/docs/transforms/coalesce)	Take the leftmost non-null value.
contains (/docs/transforms/contains)	tell whether or not a string contains another string
county_boundary (/docs/transforms/county_boundary)	Returns the boundary of the US county as a multipolygon. The state name is not case sensitive.
date_extract_d (/docs/transforms/date_extract_d)	Extract the day from the date as an integer
date_extract_dow (/docs/transforms/date_extract_dow)	Extracts the day of the week as an integer between 0 and 6 where
date_extract_hh (/docs/transforms/date_extract_hh)	Extract the hour the date as an integer
date_extract_m (/docs/transforms/date_extract_m)	Extract the month as an integer
date_extract_mm (/docs/transforms/date_extract_mm)	Extract the minute from the date as an integer
date_extract_ss (/docs/transforms/date_extract_ss)	Extract the second from the date as an integer
date_extract_woy (/docs/transforms/date_extract_woy)	Extracts the week of the year as an integer between 0 and 51
date_extract_y (/docs/transforms/date_extract_y)	Extract the year as an integer

Function Name	Description
<code>date_trunc_y (/docs/transforms/date_trunc_y)</code>	Truncates a calendar date at the year threshold
<code>date_trunc_ym (/docs/transforms/date_trunc_ym)</code>	Truncates a calendar date at the year/month threshold
<code>date_trunc_ymd (/docs/transforms/date_trunc_ymd)</code>	Truncates a calendar date at the year/month/day threshold
<code>datetime_add_d (/docs/transforms/datetime_add_d)</code>	Adds or subtracts the specified number of days to the timestamp
<code>datetime_add_hh (/docs/transforms/datetime_add_hh)</code>	Adds or subtracts the specified number of hours to the timestamp
<code>datetime_add_mm (/docs/transforms/datetime_add_mm)</code>	Adds or subtracts the specified number of minutes to the timestamp
<code>datetime_add_ss (/docs/transforms/datetime_add_ss)</code>	Adds or subtracts the specified number of seconds to the timestamp
<code>datetime_diff (/docs/transforms/datetime_diff)</code>	Calculates the difference between two dates in seconds, minutes, hours, days, business days, weeks, calendar weeks, months, or years.
<code>domain_categories (/docs/transforms/domain_categories)</code>	Returns the categories currently configured on the domain. Useful primarily
<code>domain_licenses (/docs/transforms/domain_licenses)</code>	Returns the licenses currently configured on the domain. Useful primarily
<code>email_parse (/docs/transforms/email_parse)</code>	Parse an email. This is best effort as most things are actually
<code>ensure_within (/docs/transforms/ensure_within)</code>	<code>ensure_within</code> is a function which takes a point and a multipolygon
<code>error (/docs/transforms/error)</code>	Make an error. This is useful in conjunction with a case function,
<code>floating_timestamp_day (/docs/transforms/floating_timestamp_day)</code>	Extract the day from a calendar date
<code>floating_timestamp_day_of_week (/docs/transforms/floating_timestamp_day_of_week)</code>	Extract the day of the week as an integer between 0 and 6 where Sunday is 0.
<code>floating_timestamp_hour (/docs/transforms/floating_timestamp_hour)</code>	Extract the hour from a calendar date
<code>floating_timestamp_minute (/docs/transforms/floating_timestamp_minute)</code>	Extract the minute from a calendar date
<code>floating_timestamp_month (/docs/transforms/floating_timestamp_month)</code>	Extract the month from a calendar date
<code>floating_timestamp_second (/docs/transforms/floating_timestamp_second)</code>	Extract the second from a calendar date
<code>floating_timestamp_week_of_year (/docs/transforms/floating_timestamp_week_of_year)</code>	Extract the week from a calendar date as an integer between 0 and 51.
<code>floating_timestamp_year (/docs/transforms/floating_timestamp_year)</code>	Extract the year from a calendar date
<code>forgive (/docs/transforms/forgive)</code>	<code>forgive</code> can take an optional default argument
<code>from_polyline (/docs/transforms/from_polyline)</code>	convert a linestring encode in Google's polyline format with the given precision to a Line
<code>geocode (/docs/transforms/geocode)</code>	<code>geocode</code> is a function which takes human readable addresses
<code>geocode_esri (/docs/transforms/geocode_esri)</code>	<code>geocode_esri</code> is a function which takes human readable addresses
<code>grapheme_length (/docs/transforms/grapheme_length)</code>	the length of a piece of text in unicode grapheme clusters.
<code>greatest (/docs/transforms/greatest)</code>	return the largest value among its arguments (ignoring null)
<code>hash (/docs/transforms/hash)</code>	Construct a hash value from a string value using either the md5 or sha256 algorithm.
<code>haversine_distance (/docs/transforms/haversine_distance)</code>	Return the distance of the line using haversine formula
<code>http_get (/docs/transforms/http_get)</code>	Make an HTTP Get request to a URL. The response is returned. If the server
<code>in (/docs/transforms/in)</code>	Whether or not a value is in a set of other values
<code>is_empty (/docs/transforms/is_empty)</code>	Returns whether or not the input is empty. Empty means null values,
<code>is_not_null (/docs/transforms/is_not_null)</code>	Whether or not a value is not null

Function Name	Description
<code>is_null (/docs/transforms/is_null)</code>	Whether or not a value is null
<code>is_within (/docs/transforms/is_within)</code>	<code>is_within</code> is a function which takes a point and a multipolygon
<code>json_array_contains (/docs/transforms/json_array_contains)</code>	Test if a json array contains an item. If the JSON passed to this function is not an array,
<code>json_pluck (/docs/transforms/json_pluck)</code>	Pluck a value out of a JSON string. The returned value will be a SoQL Json value.
<code>json_pluck_boolean (/docs/transforms/json_pluck_boolean)</code>	Pluck a boolean value out of a JSON string. The returned value must be a boolean, otherwise
<code>json_pluck_number (/docs/transforms/json_pluck_number)</code>	Pluck a number value out of a JSON string. The returned value must be a number, otherwise
<code>json_pluck_text (/docs/transforms/json_pluck_text)</code>	Pluck a text value out of a JSON string. The returned value may be a primitive like a
<code>least (/docs/transforms/least)</code>	return the smallest value among its arguments (ignoring null)
<code>left_pad (/docs/transforms/left_pad)</code>	Pad <code>text</code> with the minimum number of copies of <code>pad</code> to reach <code>desired_length</code> .
<code>length (/docs/transforms/length)</code>	the length of a piece of text in unicode code points. This is usually, but not
<code>like (/docs/transforms/like)</code>	If a string is like another string.
<code>location_address (/docs/transforms/location_address)</code>	Extract the address from a location
<code>location_city (/docs/transforms/location_city)</code>	Extract the city from a location
<code>location_point (/docs/transforms/location_point)</code>	Extract the point from a location
<code>location_state (/docs/transforms/location_state)</code>	Extract the state from a location
<code>location_to_point (/docs/transforms/location_to_point)</code>	Turn a location value into a point
<code>location_zip (/docs/transforms/location_zip)</code>	Extract the zip from a location
<code>lower (/docs/transforms/lower)</code>	lowercase a string
<code>make_location (/docs/transforms/make_location)</code>	This function has been deprecated. Please use the <code>make_point</code> function instead.
<code>make_point (/docs/transforms/make_point)</code>	function to make a point out of a Y (latitude) and X (longitude) coordinate.
<code>make_url (/docs/transforms/make_url)</code>	No documentation is available.
<code>not_between (/docs/transforms/not_between)</code>	Return true if the left is not within the range of the right values
<code>not_in (/docs/transforms/not_in)</code>	Whether or not a value is absent from a set of other values
<code>not_like (/docs/transforms/not_like)</code>	If a string is not like another string.
<code>parse_address (/docs/transforms/parse_address)</code>	Extract a street address from a full US address.
<code>parse_city (/docs/transforms/parse_city)</code>	Extract a city from a full US address.
<code>parse_point (/docs/transforms/parse_point)</code>	Extract the point from a full US address with point.
<code>parse_state (/docs/transforms/parse_state)</code>	Extract a state from a full US address.
<code>parse_zip (/docs/transforms/parse_zip)</code>	Extract a ZIP code from a full US address.
<code>point_latitude (/docs/transforms/point_latitude)</code>	Extract the latitude from a point
<code>point_longitude (/docs/transforms/point_longitude)</code>	Extract the longitude from a point
<code>polylabel (/docs/transforms/polylabel)</code>	Returns a point that must exist within the polygon borders. It uses the recursive grid-based algorithm described here: https://github.com/mapbox/polylabel#how-the-algorithm-works . When given a multipolygon, the point it returns is within the largest (by area) sub-polygon.
<code>random_number_between (/docs/transforms/random_number_between)</code>	Returns a random float using a uniform distribution between the lower and upper values supplied: <code>random_number_between(lower, upper)</code>
<code>random_number_normal (/docs/transforms/random_number_normal)</code>	Returns a random float using a normal distribution with the mean and variance supplied: <code>random_number_normal(mean, variance)</code>

Function Name	Description
<code>regex_capture (/docs/transforms/regex_capture)</code>	function to capture a piece of text based on a regular expression
<code>regex_named_capture (/docs/transforms/regex_named_capture)</code>	capture a piece of text based on a regular expression
<code>regex_replace (/docs/transforms/regex_replace)</code>	function to replace a piece of text based on a regular expression
<code>region_code (/docs/transforms/region_code)</code>	Turn a point into the ID of a region, based on which region the point falls within. For example, if this dataset can produce
<code>region_code_label (/docs/transforms/region_code_label)</code>	Identical to <code>region_code</code> , but returns a text value.
<code>repair_geometry (/docs/transforms/repair_geometry)</code>	Attempt to repair the geometry.
<code>replace (/docs/transforms/replace)</code>	replace text with another piece of text
<code>replace_first (/docs/transforms/replace_first)</code>	replace the first occurrence of a piece of text with another piece of text
<code>reproject (/docs/transforms/reproject)</code>	reproject a geometry from one projection to another.
<code>reproject_to_wgs84 (/docs/transforms/reproject_to_wgs84)</code>	function to reproject a geometry to WGS84. This will allow the geometry
<code>right_pad (/docs/transforms/right_pad)</code>	Pad <code>text</code> with the minimum number of copies of <code>pad</code> to reach <code>desired_length</code> .
<code>round (/docs/transforms/round)</code>	Round a number to a given precision. Trailing zeros are removed by default. Negative precisions round numbers to the left of the decimal.
<code>set_projection (/docs/transforms/set_projection)</code>	function to explicitly set the projection value on geometries which do not have projection
<code>simplify (/docs/transforms/simplify)</code>	Returns a simplified version of the Line, Polygon, MultiLine, or MultiPolygon using
<code>simplify_preserve_topology (/docs/transforms/simplify_preserve_topology)</code>	Returns a simplified version of the Line, Polygon, MultiLine, or MultiPolygon using
<code>slice (/docs/transforms/slice)</code>	Get a substring of a specified length of a text from a start index
<code>source_created_at (/docs/transforms/source_created_at)</code>	Get the fixed timestamp that this data source was created (ie: started uploading or importing).
<code>split_select (/docs/transforms/split_select)</code>	function to split a piece of text on a token, and then select
<code>starts_with (/docs/transforms/starts_with)</code>	tell whether or a not a string is prefixed with another string
<code>state_boundary (/docs/transforms/state_boundary)</code>	returns the boundary of the US state
<code>title_case (/docs/transforms/title_case)</code>	Make string title case with the exception of small words as defined by NYT Style Guide:
<code>to_boolean (/docs/transforms/to_boolean)</code>	cast a value to a true or false
<code>to_checkbox (/docs/transforms/to_checkbox)</code>	No documentation is available.
<code>to_fixed_timestamp (/docs/transforms/to_fixed_timestamp)</code>	Turn a text value into a datetime with a fixed timezone.
<code>to_floating_timestamp (/docs/transforms/to_floating_timestamp)</code>	Turn a text value into a floating datetime. "Floating" means the timezone
<code>to_json (/docs/transforms/to_json)</code>	cast a text value to json
<code>to_line (/docs/transforms/to_line)</code>	parse a WKT (text) representation of a line into a line value
<code>to_location (/docs/transforms/to_location)</code>	This function has been deprecated. Please use the <code>to_point</code> function instead.
<code>to_multiline (/docs/transforms/to_multiline)</code>	convert a line into a multiline
<code>to_multipoint (/docs/transforms/to_multipoint)</code>	convert a point into a multipoint
<code>to_multipolygon (/docs/transforms/to_multipolygon)</code>	convert a polygon into a multipolygon
<code>to_number (/docs/transforms/to_number)</code>	cast a value to a number
<code>to_point (/docs/transforms/to_point)</code>	parse a WKT (text) representation of a point into a point value

Function Name	Description
<code>to_polygon (/docs/transforms/to_polygon)</code>	parse a WKT (text) representation of a polygon into a polygon value
<code>to_text (/docs/transforms/to_text)</code>	No documentation is available.
<code>to_url (/docs/transforms/to_url)</code>	No documentation is available.
<code>trim (/docs/transforms/trim)</code>	trim characters off the start and end of a string
<code>trim_leading (/docs/transforms/trim_leading)</code>	trim characters off the start of a string
<code>trim_trailing (/docs/transforms/trim_trailing)</code>	trim characters off the end of a string
<code>upper (/docs/transforms/upper)</code>	uppercase a string
<code>uri_parse (/docs/transforms/uri_parse)</code>	Parse a URI.
<code>url_decode (/docs/transforms/url_decode)</code>	URL Decode a value
<code>url_description (/docs/transforms/url_description)</code>	Extract the description part of a link.
<code>url_encode (/docs/transforms/url_encode)</code>	URL Encode a value.
<code>url_url (/docs/transforms/url_url)</code>	Extract the url part of a link.
<code>validate_geometry (/docs/transforms/validate_geometry)</code>	Test that the geometry is valid.
<code>xml_pluck (/docs/transforms/xml_pluck)</code>	Pluck a value out of an XML string using XPath. The returned value will be a string.

 (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US) Licensed by Tyler Technologies (https://www.tylertech.com/solutions/transformation-technology/data-insights) under CC BY-NC-SA 3.0 (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US). Learn how you can contribute! (/contributing)