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/transformative-technology/data-insights)

CORS & JSONP

For security reasons, web browsers prevent what are called “cross-origin” or “cross-site” requests from one domain to another. JavaScript `XMLHTTPRequest` (https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest)s (commonly called “AJAX” requests) inherit all of the authentication context of the currently logged in user, so a malicious web page could attempt to make malicious requests that cross

domain contexts and cause trouble. Historically, that has made it difficult for web developers to build web applications making use of third-party APIs.

Fortunately, techniques have since been developed that allow developers to securely access APIs cross-domain. The two most popular ones, and the techniques that Socrata supports, are CORS (https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) and JSONP (https://en.wikipedia.org/wiki/JSONP).

A note on CORS, JSONP, and dataset permissions

In order to prevent the aforementioned malicious cross-site attacks, Socrata automatically drops all authentication and authorization on requests that come in via CORS and JSONP. As a result, these techniques can only be used to access public datasets in a read-only fashion.

Cross-Origin Resource Sharing (CORS)

CORS (https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) is a proposed standard for allowing your web browser and a web server to negotiate and allow requests to be made across domain contexts. CORS is currently supported in modern Chrome, Firefox, Safari, and Internet Explorer (10+) web browsers. The standard itself is working its way through the W3C (https://www.w3.org/) on its way to becoming official.

You don't need to do anything special to use CORS with JavaScript in a modern browser. Your web browser and our servers will automatically negotiate the cross-origin request. For example, to make a CORS request with jQuery (https://jquery.com/), you'd make your request just like you were performing it within the context of your own domain.

```
$.ajax({
  url: "https://data.chattlibrary.org/resource/e968-fnk9.json",
  method: "GET",
  dataType: "json",
  data: {
    "status": "CLOSED",
    "$$app_token": app_token
  },
  success: function( data, status, jqxhr ){
    console.log( "Request received:", data );
  },
  error: function( jqxhr, status, error ){
    console.log( "Something went wrong!" );
  }
});
```

"JavaScript with Padding" (JSONP)

If you're developing for older browsers, or you just feel like being nostalgic, you can also make use of our support for JSONP (https://en.wikipedia.org/wiki/JSONP). Also called "JSON with Padding", it is a technique for fooling a web browser into performing cross-origin requests using a special `<script>` tag that uses the `src` attribute to make a special API request. Instead of responding with just a JSON object, the server responds with JavaScript code that calls a client-declared callback function, passing the data as that function's first parameter. With the Socrata API, the name of that callback function is declared using the `$jsonp` parameter.

Sounds hacky, huh? Fortunately, tools like jQuery make it easy to use JSONP:

```
$.ajax({
  url: "https://data.chattlibrary.org/resource/e968-fnk9.json",
  jsonp: "$jsonp",
  dataType: "jsonp"
}).done(function(data) {
  console.log("Request received: " + data);
});
```

But, as we mentioned, you should only need to use JSONP as a fallback in cases where you're working with a browser that doesn't support CORS.

(cc) BY-NC-SA (http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US) Licensed by Tyler Technologies (https://www.tylertech.com/solutions/transformative-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)