

Autodesk Lab's Project Draw API Developer Guide

version 0.7.6 beta

Purpose of the API

The Project Draw team has received numerous feature requests , but one of the most popular than:

“Hey, how about an API so I can save my diagrams on my server and not yours?”

Fair enough. The Project Draw API is still beta, but it's pretty neat, if we say so ourselves. We're busy figuring out all the ways in which we can use Draw mash-ups in our own applications, wikis and blogs.

Send us your feedback, thoughts, bug reports, ideas and we'll work our tails off to make sure things get turned around quickly.

Audience

This document is written for developers that want to take advantage of the Draw API. We assume that you have some basic familiarity with JavaScript, HTML, and object-oriented programming principles.

Getting Started

Obtaining an API Key

We have an automated API Key generator you can use at <http://draw.labs.autodesk.com/labsapi/index.jsp>

Testing out the API

We provide a couple of ways to get you started on your journey with the Draw API. The intention of the first release of the API is to allow you to interact with Draw from the server-side. We have some client-side code that we're experimenting with and will release it as soon as we're happy with it.

apitester.html – once you have an API key, you can try out the Draw API calls currently available at <http://draw.labs.autodesk.com/ADDdraw/apitester.html> . This page is intended to demonstrate the way the Draw API works in general.

DrawWebServicePost.java – a very simple Java class that demonstrates the way in which you can interact with the Draw API from the server-side. You don't have to use Java, but you can use the code for a jumpstart in your language of choice.

The API

There are six API calls available currently (with more coming soon):

startSession - returns a Draw URL with a unique session id. This call is similar to the **setXML** call, but is intended for starting a new diagram. Open a browser with this URL, create a diagram and save it. The diagram XML and rasterized image (PNG) are saved in a temporary location on the Draw server (don't worry, you can delete it later).

setXML – similar to **startSession**, but is intended to be used when you already have the diagram XML and need to send it to the Draw server. **setXML** returns a URL with a unique session id. Open a browser with this URL, create a diagram and save it. The diagram XML and rasterized image (PNG) are saved in a temporary location on the Draw server.

getXML – using the session id provided in the **setXML** or **startSession** call, retrieve the XML of the diagram.

getRaster - using the session id provided in the **setXML** or **startSession** call, retrieve the raster (PNG for now, more raster options coming in future API release) of the diagram.

deleteSession - using the session id provided in the **setXML** or **startSession** call, delete the XML, raster, and session associated with this session id. *Please delete ***every*** session you create after you've called **getXML** and **getRaster** successfully.*

deleteAllSessions - using your API key, delete **all** cached XML, rasters, and sessions associated with this API key.

Programming Notes

Project Draw supports IE 6+ and Firefox 1.5+. We'll have Opera support very soon too.