JavaScript Access to Language Weaver

# Overview

Accessing any web resource from JavaScript is likely to result in a CORS (Cross-Origin Resource Sharing) error. JavaScript is very restrictive, and blocks most cross-domain requests. Even using a server in the same domain but with a different port counts as a security violation.

To work around this, you need a server that explicitly allows calls from other domains. For security reasons, RWS can't grant CORS access to everybody, so you need some sort of proxy server you control, and set that up to allow access. Communication from the proxy server to Language Weaver is no problem: CORS only limits access from browsers.

# Prerequisites

The sample server was developed in Python 3.9 (other versions should be fine). The *requests* and *web* modules need to be installed in Python.

Python provides a simple way to run a web server with just a few lines of code, but of course you can use any server technology you are familiar with instead.

# The Client Side

User access is from a simple web page: *LanguageWeaverTest.html*. You can enter a text string and Source and Target languages ( <https://translate-api.sdlbeglobal.com/documentation/#language-codes> ). For testing, it's ok to load it from your local file system. The sample uses jquery to call the proxy API, but plain JS would work, too.

The translate string is converted to base64 before sending it, and likewise the returned translation is in base64. That is just a suggestion, you may choose to use other ways of passing data around.

# The Server Side

The server consists of two main parts:

## LanguageWeaverProxy.py

This is the actual server. It implements a simple REST api with just one GET call: /translate/<text>/<sourceLng>/<targetLng>, where text is base64 encoded, for example: /translate/VGhpcyBpcyBhIHRlc3Q=/eng/spa"

It runs on port 8080 by default, if required, that can be reconfigured.

This is also where the CORS doors are opened, by adding a header value:

web.header('Access-Control-Allow-Origin', '\*')

This is the quick and easy version, open to the world. In a real life scenario, you may want to be more restrictive and only allow access from a given domain, for example:

'Access-Control-Allow-Origin', 'https://domain1.example'

## LanguageWeaverTranslator.py and LanguageWeaverUtil.py

This is the core functionality to obtain translations from Language Weaver. Since it is re-used from a standalone python sample, there is a bit more functionality then required for this demo, like file translation.

# How to use

Edit *LanguageWeaverProxy.py*, providing your own client id and secret.

Assuming Python is installed, run *StartProxy.cmd*. It should open a command window displaying: http://0.0.0.0:8080/

**About RWS**

RWS Holdings plc is the world’s leading provider of technology-enabled language, content management and intellectual property services. We help our clients to connect with and bring new ideas to people globally by communicating business critical content at scale and enabling the protection and realization of their innovations.

Our vision is to help organizations interact effectively with people anywhere in the world by solving their language, content and market access challenges through our collective global intelligence, deep expertise, and smart technology.

Clients include 90 of the globe’s top 100 brands, the top 10 pharmaceutical companies and approximately half of the top 20 patent filers worldwide. Our client base spans Europe, Asia Pacific, and North and South America across the technology, pharmaceutical, medical, legal, chemical, automotive, government and telecommunications sectors, which we serve from offices across five continents.

Founded in 1958, RWS is headquartered in the UK and publicly listed on AIM, the London Stock Exchange regulated market (RWS.L).

For further information, please visit: [**www.rws.com**](http://www.rws.com/)

© All Rights Reserved. Information contained herein is deemed confidential and the proprietary information of RWS Group\*.   
\*RWS Group shall mean RWS Holdings PLC for and on behalf of its affiliates and subsidiaries.

Open *LanguageWeaverTest.html* in a browser, and you're ready to translate.

You can also open the client on a different machine than your test server, as long as your network/firewall allows communication between the machines. In that case, you need to edit the html file and change this line:

url: 'http://localhost:8080/translate/' + b64EncodeUnicode(text) + "/" + src + "/" + trg,

to this (replacing the 11.11.111.111 by the address of the proxy server):

url: 'http://11.11.111.111:8080/translate/' + b64EncodeUnicode(text) + "/" + src + "/" + trg,