Mahlon Scott

API Quickstart Guide

This document is written to facilitate understanding of what an API is and how to use it for those unfamiliar so that development time can be minimized to what is necessary.

What is an API?

An API is like a website for programs, more specifically, it is a website representing methods within a certain program to the outside world. This way, the outside world can run the program within the API just by making **HTTP requests**. This is typically done so as to simplify the creation of graphical interfaces for the program by not requiring the front-end developers to write the program itself, i.e. the front-end developer does not have to worry about how the bank transaction is being made, only that it is being made.

In the real world this also ensures the security of critical information as it can be completely contained withing the API implementation/server itself and never sent to the outside world.

How to use an API

In order to use an API all you have to do is make **HTTP requests** to the **endpoints** of the API. **Endpoints** are like different pages of the website, or different methods of the program that the API represents. Most APIs only use **GET** and **POST** requests. A **GET** request contains no large amounts of data, but may contain shorter parameter information, like a username or user id. A **POST** request is used when we need to send a larger amount of data, like a JSON object. In

JavaScript, you can make an **HTTP POST request** with the XMLHttpRequest object, as follows:

```
var request = new XMLHttpRequest();
var endpoint = "/Login.php";

var data =
{
    "login": "MScott@SmallProject.com",
    "password": "COP4331m"
};

request.open("POST", "http://143.198.116.115/LAMPAPI" + endpoint, true);
request.setRequestHeader("Content-type", "application/json; charset=UTF-8");
request.send(JSON.stringify(data));
```

In order to actually capture/use the output you do need to make a callback function since we are using an asynchronous HTTP request as specified by the true in request.open (this just means that instead of waiting for the request to finish before we execute more code we execute the request and then immediately move on to the next code that needs to be executed, so the callback function just tells JavaScript what to do when it finally gets the response to the HTTP request). However, you can figure out how to write that (and that is what the sample code is for).

API Documentation

The API documentation can be found here:

https://app.swaggerhub.com/apis-docs/KingJMS1/POOSDLampProject/1.0#/