<http://developer.echonest.com/tutorial-overview.html>

* Intro
  + Page 1: Welcome to an informal Echo Nest API tutorial! Before we begin, let me show you some of what Echo Nest can do. Tell me the name of one of your favorite artists: <textbox> <submit button>
  + Page 2
    - <fieldset>Info about artist</fieldset>
    - Echo Nest collects and distributes information about music. By requesting certain attributes about specific artists, genres or songs, we can learn, listen to and watch a wealth of information about today’s and history’s favorite aural adventures. This tutorial will be a springboard to helping you request your own musical information.
    - <Link to Table of Contents>
* Before We Request
  + What you should already know
    - This tutorial assumes you are familiar with
      * JavaScript
      * JSON objects
      * The concept of clients and servers
  + Getting a Key & Why We Need One
    - Echo Nest requires that developers create an Echo Nest Developer Account. This account will be used to track a developer, and the developer’s use of the Echo Nest API key. This key is important: it is required as part of any request to Echo Nest. Without it, requests will be ignored.
    - Using this API key allows Echo Nest to track how often an application makes requests to the Echo Nest API. If too many requests are received within a set amount of time, rate limiting will be applied. This ensures access to other applications using the same API. For our purposes, it is unlikely that we will encounter rate limiting.
    - To set up a developer account, go to Echo Nest’s Create an Account page: <https://developer.echonest.com/account/register> and fill in the requested information. Follow the instructions for obtaining your key.
  + Responses: Was the Request a Success or a Failure?
    - Responses can be in JSON, XML or JSONP. We’re sticking with JSON for this How-To. XML and JSONP requests and responses will hold the same information in a slightly different format. The only differences are
      * What format was requested
      * How to parse the response, which will be based on the requested format
    - Responses generally contain two objects: the payload, the contents of which depend on the initial request, and a status object, which provides information on the success, or lack thereof, of the request.
  + Warning About Dynamic Data: Because music data is constantly shifting, evolving and growing, it is foolish to assume all portions of a response will have complete data, or data at all. Code dealing with responses should be robust enough to handle null or undefined data, new data fields or a new order of these data fields. In other words, search the dictionary response object for the required data and protect against missing information. Do not assume a field exists simply because it was there during development, and do not rely on hardcoded indices to locate the data.
  + Formatting a Request
* Basic Artist Requests
  + Getting Images
    - Because Echo Nest pulls music data from a variety of Internet sources, several of our requests can have multiple responses. These requests generally have default parameters for the number of responses returned (and even the default index of all possible results).
    - To get one or several images of a particular artist, we can make an <code>images</code> request. This will require:
      * URL + Base Category (/artist/)
      * Query: images?
      * Key: api\_key=<your key>
      * Artist Name || ID (name or ID)
      * Number of Results (defaults to 15)
      * Start Index (defaults to 0)
      * License
        + License is where we specify which license the results should use. Several license options exist. For the purposes of this How To, we will stick to the Public Domain license.
    - The response will come in two parts:
      * Status
        + The status object tells us if the request was successful in “code”. If it was not, it will provide an idea of what went wrong in the “message” attribute. “version” is the current version of the API; useful in the more robust of response handlers.
      * Payload
        + Payload will vary significantly based upon the initial request. This is the response object to pay attention to when reading documentation. Most of the other stuff stays relatively the same, with a couple changes here and there. This stuff changes for everything.
  + Getting Videos
    - To get one or several videos of a particular artist, put together a request with this information:
      * URL
      * Base Category
      * Query: videos?
      * Key
      * Artist Name || ID
      * Number of Results
      * Start Index