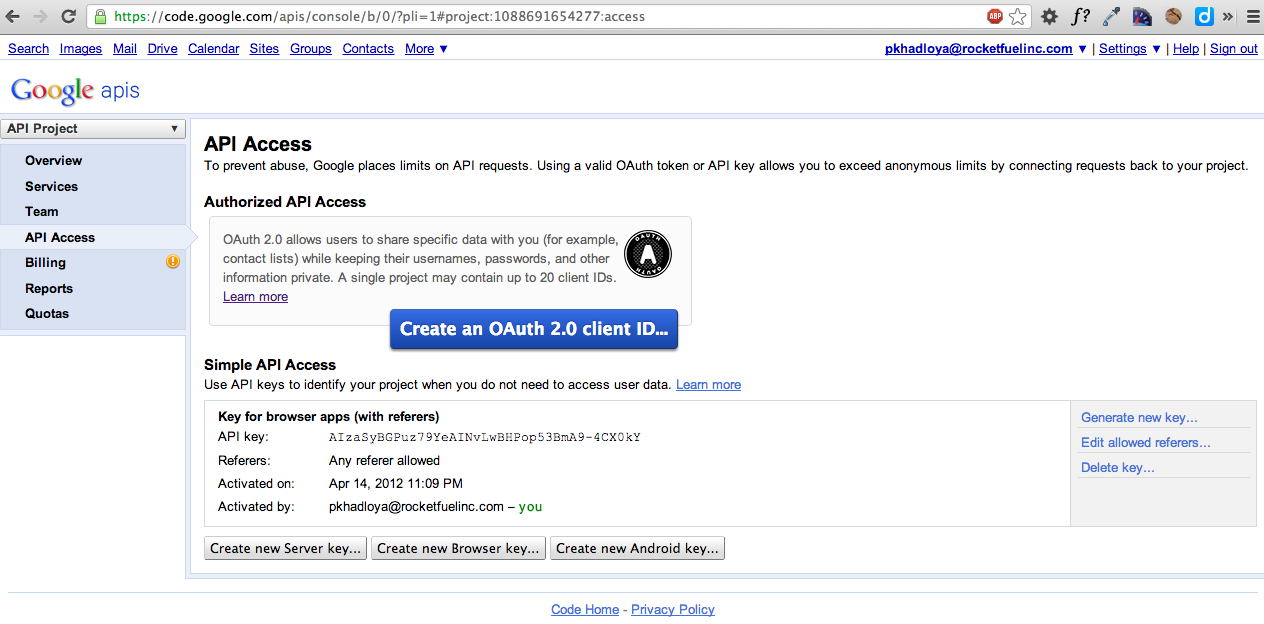
1. Gem **google\_places**

2. This gem provides a Ruby wrapper around the Google Places API, using HTTParty. At this moment the gem does not support OAuth authentication and will only work with an **API key**.

3. **Obtaining an API Key**

To be able to use this gem, you’ll need a Google Places API key. To request an API key, point your browser to [code.google.com/apis/console](https://code.google.com/apis/console) and select your project from the drop list on the left hand side. You’ll find your API key on the \*API Access\* tab under \*Simple API Access\*.



4. But if your app already uses OAuth2 protocol to authenticate to Google, they would not allow you to create an API Key as it would be redundant.

The APIs represented in the Google APIs Console require that requests include a unique project identifier. This enables the Console to tie a request to a specific project in order to monitor traffic, enforce quotas, and handle billing. Google supports two separate mechanisms to create a unique project identifier: OAuth 2.0 client IDs and API keys.

* An [OAuth 2.0](https://developers.google.com/accounts/docs/OAuth2) client ID can generate an access token for an application that uses the OAuth 2.0 protocol to call Google APIs. The token contains an identifier that uniquely identifies your application, and allows the Console to properly attribute corresponding API traffic, quotas, and billing data to the Console project that the API is enabled within.
* An [API key](https://developers.google.com/console/help/#generatingdevkeys) is a unique key that you generate using the Console. When your application needs to call an API that's enabled in this project, the application passes this key into all API requests as a key=API\_key parameter. Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

If you are only calling APIs that do not require user data, such as the Google Custom Search API, then API keys may be simpler to implement. However, if your application already uses an OAuth 2.0 access token, then there is no need to generate an API key as well. In fact, Google ignores passed API keys if an OAuth 2.0 access token is already associated with the corresponding project.

**Note**: You must use either an OAuth 2.0 access token or an API key for all applications. Not all APIs require authorized calls. See your API documentation to learn whether authorization is required for specific calls.