**geocode**

As of mid-2018, the Google Maps Platform requires a registered API key. While this alleviates previous burdens (e.g. query limits), it creates some challenges as well. The most immediate challenge for most R users is that ggmap functions that use Google's services no longer function out of the box, since the user has to setup an account with Google, enable the relevant APIs, and then tell R about the user's setup.

To obtain an API key and enable services, go to <https://cloud.google.com/maps-platform/>. This documentation shows you how to input the requisite information (e.g. your API key) into R, and it also shows you a few tools that can help you work with the credentialing.

To tell ggmap about your API key, use the register\_google() function, e.g. register\_google(key = "mQkzTpiaLYjPqXQBotesgif3EfGL2dbrNVOrogg") (that's a fake key). This will set your API key for the current session, but if you restart R, you'll need to do it again. You can set it permanently by setting write = TRUE, see the examples. If you set it permanently it will be stored in your .Renviron file, and that will be accessed by ggmap persistently across sessions.

Users should be aware that the API key, a string of jarbled characters/numbers/symbols, is a PRIVATE key - it uniquely identifies and authenticates you to Google's services. If anyone gets your API key, they can use it to masquerade as you to Google and potentially use services that you have enabled. Since Google requires a valid credit card to use its online cloud services, this also means that anyone who obtains your key can potentially make charges to your card in the form of Google services. So be sure to not share your API key. To mitigate against users inadvertantly sharing their keys, by default ggmap never displays a user's key in messages displayed to the console.

Users should also be aware that ggmap has no mechanism with which to safeguard the private key once registered with R. That is to say, once you register your API key, any function R will have access to it. As a consequence, ggmap will not know if another function, potentially from a compromised package, accesses the key and uploads it to a third party. For this reason, when using ggmap we recommend a heightened sense of security and self-awareness: only use trusted packages, do not save API keys in script files, routinely cycle keys (regenerate new keys and retire old ones), etc. Google offers features to help in securing your API key, including things like limiting queries using that key to a particular IP address, as well as guidance on security best practices. See<https://cloud.google.com/docs/authentication/api-keys#securing_an_api_key> for details.

**Author(s)**