Beeminder Plain Data Integration

A Beeminder Spec

Last updated: 2024-01-02

Spec level: 0.40

Overview

A generic Beeminder integration for connecting to simple APIs.

Beeminder allows users to create integrations with various external services, such as weight trackers and chess apps. But what if a user wants to have a goal for learning flashcards? Or running the laundry machine? Or practicing juggling with their internet connected juggling balls? There are many more websites and tools out there than Beeminder has time to create new integrations for! There *are* integrations available for generic tools like IFTTT and Make, but they're often difficult to work with and come at an additional cost - and who wants to introduce a third party into the mix? They also don't allow for Beeminder to "pull" data, so goals with them are often out of date. This spec proposes a generic integration that allows users to connect to any compliant data source.

Nongoals

There are many possible APIs out there, and we're not going to support all of them. Future versions of the integration might add support for authentication or for parsing a complicated response body, but that won't be in this. In fact, we're going to support precisely one: an HTTP(S) GET request to a URL that returns a single number.

There will be no way to add comments from the API. This could be a future feature though.

This is a tool for a semi-technical user, and that's a fine audience. Beeminder's users often are already, so we'll double down on it. The documentation should have enough for one of those technical users to use it, but it's not going to explain how to add support for this API format to applications.

There will not be a way to "prepackage" an API into a pseudo-integration; developers of APIs should just tell their users what URL to enter when creating the goal. There are possible ways to do this (eg. allow the encoded API URL to be a request parameter in the new goal page URL), but that is outside of the scope of this spec.

Use Cases

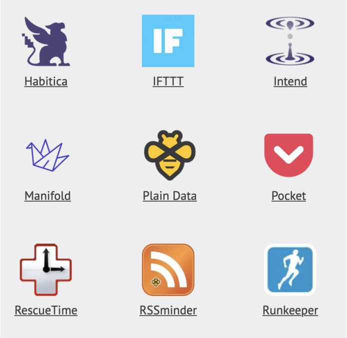
Francine is learning French but is too akrasiatic to do her flashcards every day, so she has a large number of cards that have built up and need to be reviewed. She uses Mnemosyne to handle her flashcards, but there's no Mnemosyne integration with Beeminder. Luckily, she has Mnemosyne hosted on her web server, at a publicly available web address, and Mnemosyne has plain data APIs available, both for the number of cards that need to be reviewed and the number of cards that are unmemorized. She goes onto the Beeminder and creates a "whittle down" goal with the plain data integration, setting the URL to be the API for the cards to be reviewed. Francine will gradually whittle down that amount, until she's at zero, at which point she will only need to do the new cards that become due each day.

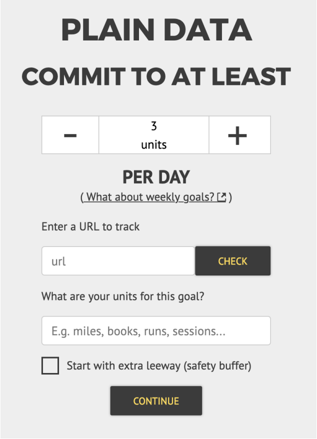
Developer Dave is creating a tool that he'd like to offer a Beeminder integration for. But he doesn't have the time to actually go and create an integration using the real API, especially since he wants to make integrations for lots of different tools. Now, Dave can create an extremely simple endpoint that just returns a single number - something that he already knows how to do - and go on his way.

Description

Goal Creation

A new integration option will be added to the Beeminder new goal page, "Plain Data" (not "Plane Data", though tracking plane flights could be a custom integration done with this!).




PLAIN DATA

COMMIT TO AT LEAST

− 3 units +

PER DAY

(What about weekly goals?)

Enter a URL to track

url CHECK

What are your units for this goal?

E.g. miles, books, runs, sessions...

□ Start with extra leeway (safety buffer)

CONTINUE

When clicked on, the user will be taken to the "Select goal type" page, just like if they had clicked on the "Add progress manually" option on the first screen, except without the do more/less goal types. (See the Data section side notes.)

After choosing a goal type, the user will be taken to a page with a header of "Plain Data". This will have the normal goal creation forms from creating a normal goal, along with a subheader saying "Enter a URL to track", a text box for entering a URL, and a "Check" button. When the "Check" button is clicked, it will get the data from that URL (see "API" section later) and verify that it is valid. If it is not valid, an error message will display using the normal top bar error display.

When the continue button is clicked, the user will be taken through the rest of the goal creation flow like normal, and the goal will be created. An initial data point will be loaded from the URL.

Data

API data is loaded in the simplest possible way - it's even the default for cURL! For example, if you type `curl mywebsite.com/myAPIEndpoint/` into a terminal, then it must return just a single number for it to be valid for this. The number can be an integer or a float, just like normal Beeminder data, but that's it! No comments, nada. The HTTP status should be 200.

Just like other integrations that "pull" data, Beeminder will automatically make requests to the endpoint throughout the day, along with whenever the user clicks the refresh button. If this is the first time of the day loading data, a new datapoint will be added. If this is a later time, and the data is different from the previous datapoint, it will update the existing datapoint and replace its value with the new data.

Side note: data that is loaded will be assumed to be idempotent, so that any number of requests to the endpoint will always return the same number, unless the underlying data has changed. For example, if a user wants to track how many times a day they open their fridge, the API should return the total number of times the fridge has ever been opened (with an odometer/whittle-down/weight goal). This is technically correct REST behavior for a GET endpoint, but people do crazy things sometimes. Advanced users can also set up a custom goal if they really feel like it, and then maybe they could break these assumptions, but they're on their own if they really want to do that.

Deeper side note: We would also like to support do more/less goals. For the example above, that would be the total number of times that day the fridge has ever been opened. However, that assumes that the data source's "day" aligns precisely with the goal's "day", which is a complicated assumption to make. Since this is a more advanced feature, that may be okay, but we would want to provide clear documentation around this. Because of that, this initial spec recommends not allowing do more/less goals.

Open Questions

- What documentation needs to be added for new integrations? Any? This should probably get a bit more explanation than some other ones.
- What icon should be used on the new goal page?
- What error handling should be done if the API doesn't return a 200 status or returns something other than a plain number?