

MicroFVS

a new REST API for the
Forest Vegetation Simulator

David Diaz

Growth Model User Group
April 22, 2026



vibrant planet



vibrant planet
data commons

Today's Focus

MicroFVS: A new REST API for the Forest Vegetation Simulator

- Motivating Use Case
- FVS as a Containerized Service
- Live Demo
- Where MicroFVS fits



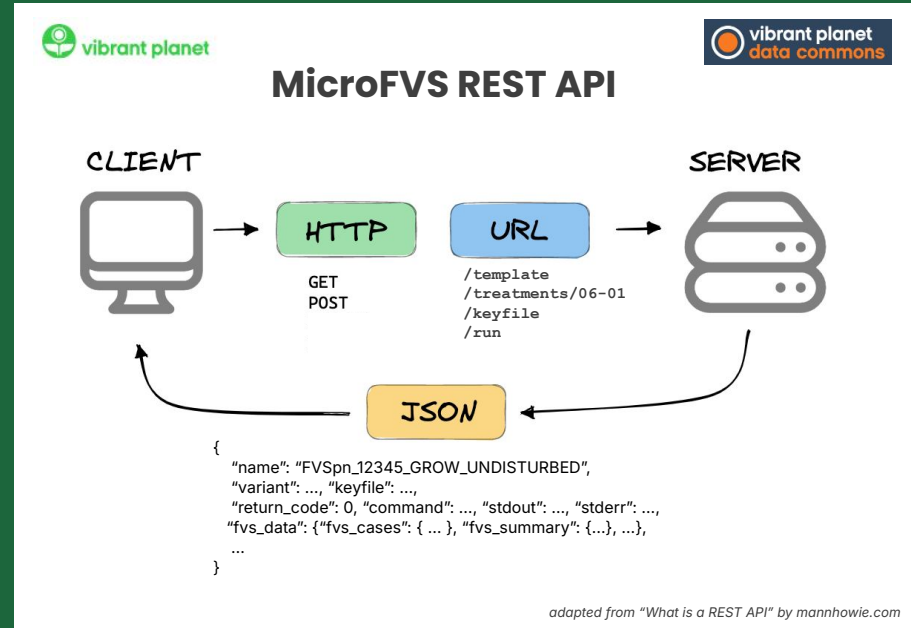
Motivation

As a forest modeler, I need a scalable, transparent, reproducible, and reliable G&Y service so that I can:

- Run simulations that always return (no silent failure) with a consistent data contract
- Run lots of simulations efficiently (outsourcing compute)
- Provide an audit trail showing how simulations were configured and executed, allowing them to be completely reproducible
- Be confident core G&Y behavior is based on best-available data (and regularly updated).

FVS in a box

- Dockerized versions of FVS built from source that can be deployed and autoscaled in the cloud.
- Automated unit & integration tests catch breaking changes before containers for new FVS versions released into service.
- REST API clearly defines data contract and building blocks.



Interacting with the service

Multiple endpoints serve distinct purposes:

- /run (implemented) - runs FVS simulation
- /keyfile (implemented) - generates a keyfile from a template
- /snapshot (not yet) - translates raw inventory into info (zero-cycle simulation)

Everything going into and out of the service is JSON.

Primary API endpoint (/run) does:

- StandInit, TreeInit, and Keyfile submitted as JSON
- + StandInit + TreeInit to DataFrames, injected into FVSData.db.
- + FVS CLI run on Keyfile with outputs to FVSData.db
- + FVSData.db scraped, extended by run metadata
- ← FVS output and run metadata returned as JSON

github.com/Vibrant-Planet-Open-Science/microfvs

The screenshot shows the GitHub repository page for `microfvs`. The main content area displays the README, which includes the following text:

MicroFVS

This repository includes the code for an API employed by a web service developed by [Vibrant Planet](#) for executing simulations using the [Forest Vegetation Simulator \(FVS\)](#), a forest growth-and-yield developed by the USDA Forest Service.

This API is the backbone of a microservice (hence the name "MicroFVS") allowing users to interact with FVS in a contained and well-maintained environment without requiring the installation of any specific software. The core functionality enabled by the API is the execution of FVS simulations and retrieval of results from them in a standardized format (JSON).

The API exposes basic templates and building blocks for constructing plain-text FVS Keyfiles which provide the instructions to FVS for executing a growth-and-yield simulation on a single forest stand. These resources are intended to help users develop their own Keyfile templates and to use their own management and natural disturbance recipes in these simulations.

The API also provides access to a [national compendium of silvicultural treatments^{\[1\]}](#) prepared by a team of silviculture experts from each National Forest System Region and Research Station of the US Forest Service in the form of FVS Keyword Component (KCP) files.

Getting Started

You'll need to have Docker and Git installed to be able to get this working.

The right sidebar shows the repository's metadata:

- Contributors 2**: [d-diaz](#) David Diaz, [vlandau-vp](#) Vincent Landau
- Languages**: Python 96.0%, Jinja 2.9%, Dockerfile 1.1%



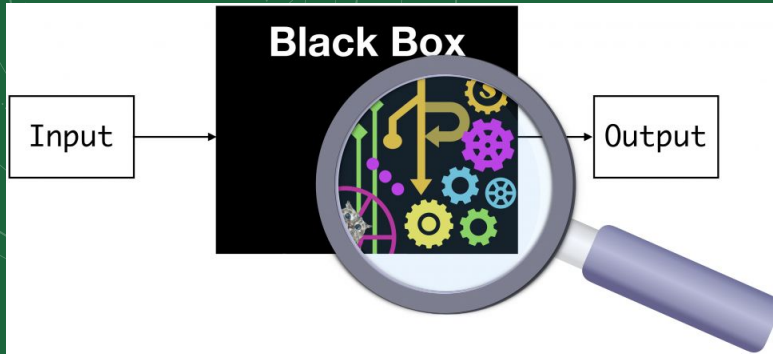
vibrant planet

LIVE DEMO

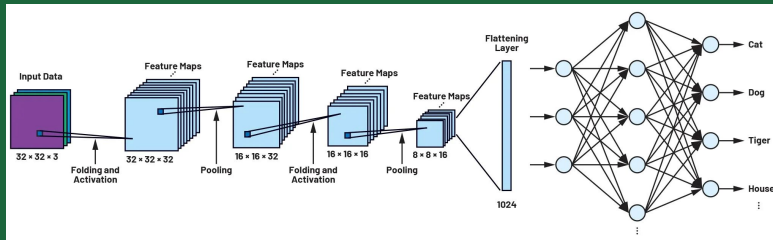
MicroFVS



What's in the box?



<https://blog.ml.cmu.edu>



<https://www.embedded.com/understanding-convolutional-neural-networks/>

- G&Y engine is a black box for mapping tree/stand data and configs to standard outputs.
- Stable API enables community development while maintaining data contract with users.
- We can put any new model in the box that satisfies the data contract.
- We'll start with an FVS keyfile that uses keywords to bend FVS to best reproduce FIA observations.

What's next?



2026 Q1 • MicroFVS v1.0 on GitHub

Q2

- MicroFVS presented at GMUG
 - Demonstration of MicroFVS running on cloud infrastructure
 - R & Python wrappers on GitHub
 - User interviews begin
-

Q3

- Example workflows as Jupyter Notebooks on GitHub
 - FVS keyword adjustments from contemporary FIA fitted
 - User interviews completed
-

Q4

- Self-hosting guidance or VPDC MicroFVS service access instructions
 - Improved FVS defaults on GitHub
-

FVS and what comes next.

- Lean into CAFS existing partnerships, build a community of practice to outline a path forward
- Differentiate needs and sequencing of work oriented to software architects/maintainers versus users/partners
- Meet quarterly to develop shared working guidelines and future priorities
- Explore and pursue opportunities for leveraged funding
- Build shared repositories of code, data, and documents
- Engage and update broader audiences like GMUG





Thank you.

Find me

ddiaz@vibrantplanet.net