

The Window Became a Climate Interface

A Hedegreen Research build note on Windowsill, the public TID plant tool and API that treats a window, balcony or garden edge as a local climate interface.

2026.06.07 21:20 DENNIS HEDEGREEN BUILD V1.0

[HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-WINDOW-BECAME-A-CLIMATE-INTERFACE/](https://hedegreenresearch.com/articles/the-window-became-a-climate-interface/)

A window is not just a hole in a wall.

For a plant, it is a climate interface.

That sounds larger than the thing itself. The thing itself is small: a web tool, an API, a plant library, a map click, a direction, a month, and a recommendation list. But the smallness matters, because the question is small too.

What can live here?

Not in Denmark in general.

Not indoors in general.

Here.

Beside this window, on this balcony, or in this garden. At this latitude. Facing this direction. In this month. Under the kind of climate history this place tends to have.

That is the premise behind Windowsill.

The Generic Answer Was Too Big

Most plant advice begins by making the place disappear.

It says basil likes sun. It says mint is easy. It says tomatoes need warmth. It says parsley can tolerate partial shade. These sentences are not useless. They are just not yet local.

A south-facing kitchen window in Copenhagen is not the same place as a north-facing balcony in Aarhus. A garden bed is not a windowsill. A balcony is not a field. June is not February. A plant record is not a promise.

The ordinary answer compresses too much.

Windowsill starts from the other side. It asks for the place first.

The tool takes a location, a direction, a growing context, and a time of year. Then it compares that context against a constrained library of plant varieties and returns candidates with scores, caveats, and seasonal timing.

That is not magic.

It is a decision surface.

What Was Built

Windowsill now exists as a public Hedegreen Research TID tool and as a live API.

At the time of publication, the live API reported version 0.6.0, library version 2026-06-07, scoring version 0.8.0, and 148 plant varieties. Those numbers are not the point of the article, but they matter because a build note should not lean on stale public facts.

The TID tool lets a person choose a point on a map, select a context such as windowsill, balcony, or garden, set an orientation, and ask for recommendations. It can also generate a seasonal calendar and export a report PDF.

The plant library is variety-level rather than only species-level. That matters because the question is not just "can basil grow?" The better question is closer to "which kind of basil, in what conditions, with what limits visible?"

The system also carries risk and uncertainty markers. Some plants should not be treated casually. Some entries are better sourced than others. Some conditions cannot be inferred from latitude, direction, and historical weather data. A room can be warmer than the outside average. A balcony can be sheltered. A window can be shaded by a nearby building. A person can overwater the plant.

The tool does not know all of that.

So it should not pretend to know all of that.

Not an AI Gardening Answer

Windowsill is not designed as a generic AI gardening assistant.

That distinction is important.

A generic AI answer can sound helpful while floating above the actual situation. It can produce a confident sentence about a plant without making the location, season, direction, and growing context explicit. The answer may be plausible, but the person reading it still has to translate the advice back into the place they actually have.

Windowsill moves the translation step into the interface.

It does not ask, "What should I grow?" as an abstract lifestyle question. It asks something narrower:

What might fit this edge of the home, under these stated conditions, according to this current library and scoring model?

That narrower question is less impressive.

It is also more useful.

The Library Should Be Correctable

The plant library should not become a private little oracle.

If someone knows a missing variety, a local name, a growing failure, a better source, or a correction to an existing entry, that belongs near the project. Windowsill is built around structured plant records, but the knowledge behind those records does not only live in code.

It lives with people who grow things.

Someone may know that a plant is common under a local name that does not match the formal taxonomic entry. Someone may have tried it on a balcony and failed every time. Someone may know that a plant works in a sheltered courtyard but not on an exposed sill. Someone may know that a safety note is too weak, or that an entry is missing a serious caveat.

That kind of knowledge should not be flattened into false certainty.

The Windowsill direction is to keep expert review, source traceability, uncertainty, and community field signal visible as separate things. A taxonomic correction is not the same as a grow report. A grow report is not the same as proof. A local name is not the same as a botanical identity.

But all of them can help make the entry more readable.

That is why participation matters.

Windowsill welcomes plant suggestions, corrections, source notes, uncertainty notes, and real growing observations. GitHub pull requests are useful when someone can prepare a full plant JSON entry. But the project should also be possible to approach from a phone: a plant name, a region, a few links, a personal observation, and a clear statement of uncertainty can still be useful.

If something is wrong, missing, or worth adding, open an issue or email api@windowsill.dk.

The point is not to pretend that every contribution is equally strong.

The point is to let the system improve without hiding where the knowledge came from.

The API Should Stay Usable

There is another small design choice here.

The API has limits, because a public endpoint without limits becomes fragile. No key is needed for small casual use. API keys exist for people who want to build a widget, a tiny app, a school project, a local tool, or some other small thing that should not break the service.

But the point is not to turn Windowsill into a paywall.

If a project is small, useful, curious, educational, local, or just honestly non-abusive, the intended answer is simple: ask. The plan is to give free keys where that makes sense.

Email api@windowsill.dk if you need one.

That is not a business model. At this scale, it is closer to maintenance.

Railway still has to be paid. Servers still cost money. Someone still has to keep the endpoint from falling over. But the project is not being built to extract rent from small plant tools.

If necessary, the funding model can be as undignified and exact as collecting deposit bottles to pay the hosting bill.

Pant for a plant API.

That joke is also a boundary. The API should be treated as public infrastructure at a tiny scale, not as a growth funnel.

The Household Edge

The reason this belongs in TID is that TID is the public instrument layer of Hedegreen Research.

An article can describe a problem. An instrument lets someone touch the problem.

Windowsill is an instrument for a very small public question: how does an ordinary domestic edge become more readable?

The edge can be a windowsill. It can be a balcony. It can be a garden position. In each case, the tool treats the surface as an interface between household life and the outside world.

This connects to Heat Pressure, but carefully.

In Heat Pressure, the Windowsill/Rima relation is a household side-note. It is not part of rent pressure, cooling pressure, tourism pressure, or any housing score. A plant recommendation beside Rima can make a dwelling feel more concrete, but it is not evidence that the dwelling is safe, affordable, or climate-resilient.

That boundary matters.

A small tool becomes misleading when it is allowed to count as more than it knows.

The Limits Are Part of the Product

Windowsill should not be read as horticultural certification.

It is not a weather forecast. It is not a plant safety database. It is not medical, culinary, or herbal advice. It does not prove that a specific windowsill receives the light it appears to receive. It cannot know whether a radiator sits below the window, whether the glass is shaded, whether the room is dry, or whether the plant will actually be watered.

The honest output is therefore not "grow this."

The honest output is closer to:

These are plausible candidates for the context you described. Here is why they scored that way. Here are the limits you should keep in view.

That is the level the tool can defend.

Even the generated report has to obey that discipline. A PDF with the wrong page total is a small bug, but it breaks trust in exactly the wrong place. If the tool is about making a small context readable, the document it exports has to be readable too.

Small instruments earn trust through boring details.

Why This Matters

There is a broader Hedegreen Research pattern here.

The interesting question is not always whether a machine can produce a fluent answer. Sometimes the better question is whether a machine can help make a real situation smaller, clearer, and more inspectable.

Windowsill does that at a domestic scale.

It does not claim that the home has become fully knowable. It does not claim that plants are simple. It does not claim that climate, light, water, soil, containers, and human care can be collapsed into a single score.

It also does not claim that plant knowledge should only come from a repo.

The public tool is one surface. The API is one surface. The plant JSON files are one surface. The person who writes in because a local variety is missing is also part of the system, if the correction can be traced and reviewed honestly.

The person who asks for a small free API key is part of it too, if they build something that keeps the source and uncertainty visible.

That is the community version of the same idea.

Make the place readable.

Make the source readable.

Make the uncertainty readable.

Then let the record improve.

The window became a climate interface because the tool stopped treating it as decoration.

It became a place with direction, season, location, limits, cost, access, and consequences.

That is what the build is for.

RELATION MEMORY

OPENS [Windowsill TID Tool](#) · The public tool surface described by the article.

NEARBY [Heat Pressure Is Not a Housing Audit](#) · Windowsill appears there only as a household Rima side-note, not as Heat Pressure evidence.

NEARBY [If the Data Exists, the Door Can Be Built](#) · Shares the principle that public data and tools need legible doors, not just hidden files or endpoints.

NEARBY [The Tool Learned to Say Not Yet](#) · Connects to the principle that public tools should expose limits instead of pretending certainty.

CANONICAL URL: [HTTPS://HEDEGREENRESEARCH.COM/ARTICLES/THE-WINDOW-BECAME-A-CLIMATE-INTERFACE/](https://hedegreenresearch.com/articles/the-window-became-a-climate-interface/)

PDF VIEW GENERATED: 2026-06-07T19:28:03.145Z