

GUROBI
OPTIMIZATION

Gurobi Intelligence Hub

Copyright © 2026, Gurobi Optimization, LLC

Aug 21, 2026

Revision: 7908756c0

CONTENTS

1	Overview	3
1.1	Terms of Service	3
1.2	Using the Web Application	4
1.3	Tools	6
1.4	How the Intelligence Hub Compares to a General-Purpose AI Assistant	8
2	Gurobot	11
2.1	Features	11
2.2	Using Gurobot	11
3	Explainer	17
3.1	Features	17
3.2	Background	18
3.3	Using Explainer	19
4	Modeler	25
4.1	Features	25
4.2	Using Modeler	26
4.3	Generating a Mathematical Formulation	31
4.4	Checking an Existing Project	32
4.5	Reimplementing or Converting a Model	33
4.6	Managing Conversations	33
5	MCP Server	35
5.1	Design Rationale	35
5.2	How It Works	36
5.3	Prerequisites	36
5.4	Licensing Options	36
5.5	Path Mapping	37
5.6	Usage Examples	37
5.7	Exposed Ports	38
5.8	Configuring AI Assistants	38
5.9	Updating the MCP Server	40
5.10	Environment Variables	40
5.11	File Handling	40
5.12	Viewing Local Logs	40
5.13	Monitoring MCP Sessions	41
6	API Keys	43
6.1	Creating an API Key	44
6.2	License File	45

6.3	Viewing API Keys	45
6.4	Managing API Keys	46
7	Settings	47
7.1	Delete All Chats	48
8	Frequently Asked Questions	49
8.1	Getting Started	49
8.2	Security and Privacy	50
9	Terms of Service	51
9.1	Introduction	51
9.2	Acceptance of Terms	51
9.3	Scope of Service	51
9.4	User Responsibilities	52
9.5	Data Collection and Privacy	52
9.6	Data Processing and Security	53
9.7	Termination of Access	54
9.8	Generative AI Disclaimer and Accuracy Notice	54
9.9	Limitation of Liability and Indemnification	54
9.10	Modifications to the Terms	55
9.11	Contact Information	55

Welcome to the [Gurobi Intelligence Hub](#)—your AI command center for mathematical optimization.

The Gurobi Intelligence Hub is a web application that provides AI-powered agents to help you work more effectively with Gurobi products. See the [Overview](#) for getting started information. The Intelligence Hub currently offers three specialized agents:

- [Gurobot](#): An AI assistant that delivers instant, expert-level Gurobi support with personalized, context-aware answers, code snippets, and documentation links backed by Gurobi’s knowledge base.
- [Explainer](#): An optimization analysis agent that diagnoses infeasibility, proposes feasibility repairs, and runs sensitivity analysis on your models, grounded in Gurobi’s curated knowledge base and optimization expertise.
- [Modeler](#): A modeling assistant that helps you co-develop a clear, plain-language optimization specification with AI, then turns that specification into a complete optimization project—including a mathematical formulation, `gurobipy` implementation, and validated unit tests.

The Intelligence Hub also provides an [MCP server](#) that runs locally on your machine, allowing you to access Gurobot, Explainer, and Modeler directly from MCP-compatible tools such as IDE extensions and AI coding assistants. [API keys](#) are used to generate a license file that authenticates the MCP server with the Intelligence Hub backend.

For configuration options, see [Settings](#). If you have questions, check the [Frequently Asked Questions](#) or review the [terms of service](#).

OVERVIEW

The Intelligence Hub is accessible from the [Gurobi portal](#) and directly at intelligence.gurobi.com.

Note

Gurobot is generally available (GA since June 2025).

Modeler is in **beta**—the current scope is implemented with more improvements to come. Please share feedback using the thumbs up / thumbs down buttons in the chat.

Explainer and the **MCP server** are **experimental**—available for you to explore, and they may change significantly before becoming generally available. Please share feedback using the thumbs up / thumbs down buttons in the chat of the Explainer.

1.1 Terms of Service

When you access the Intelligence Hub for the first time, you are presented with the [Terms of Service](#) and [Privacy Policy](#). You must review and accept these terms before you can use the agents. Click **ACCEPT** to proceed, or **DECLINE** to opt out. You will not be able to use the Intelligence Hub agents until the terms are accepted.

Key points covered by the Terms of Service:

- **Scope:** The service is designed solely for discussions and inquiries related to mathematical optimization and the usage of Gurobi products.
- **Usage limits:** Daily usage limits apply to your interactions. Access may be temporarily suspended if you exceed those limits.
- **Data storage:** All interactions with the service are stored, including inputs, outputs, and any files generated during a session. You can delete individual conversations or all conversations from your account at any time (see [Settings](#)).
- **Data privacy:** Your interactions are not used to train large language models (LLMs). They may be used to validate, upgrade, and improve the service.
- **Security:** Inputs you submit to the service are protected by:
 - **Encryption in transit:** All inputs sent between your device and the service are encrypted with HTTPS.
 - **Encryption at rest:** Inputs are encrypted using application-layer encryption, with a unique key per file.
 - **Isolated processing:** Inputs are processed in a dedicated environment with no shared state or filesystem with any other session or user.

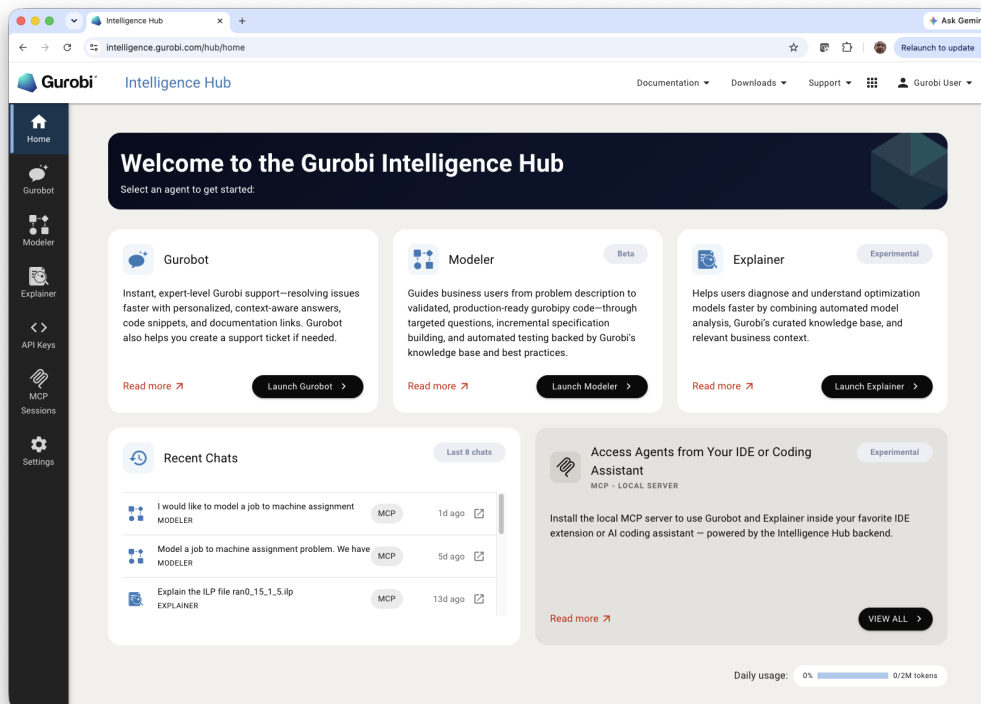
- **Limited exposure:** Portions of your inputs are transmitted to a large language model running within Gurobi’s secure cloud infrastructure solely to fulfill your request.
- **Model Input Files:** Optimization model files you submit are used only to deliver your request. They are not added to Gurobi’s model library and are not used for solver benchmarking. If you escalate an interaction to a Gurobi support ticket, the associated model files become governed by Section 7.2 of the Gurobi EULA.
- **Deletion:** Your inputs are permanently deleted when you delete the chat session, delete all chats for an agent, delete all chats, delete your account, or send a written deletion request to operations@gurobi.com, subject to applicable legal retention requirements.
- **Personal data:** Do not include data that identifies or could reasonably be linked to a natural person (names, addresses, social security numbers, payment card information, etc.) in your inputs.
- **Generative AI disclaimer:** The service uses generative AI and may produce inaccuracies. You should independently verify any output before relying on it.

1.2 Using the Web Application

The sections below cover the parts of the Intelligence Hub UI you interact with directly: the home page, the sidebar navigation, the chat insights panel, and the daily usage indicator.

1.2.1 Home Page

After accepting the terms, the home page displays the available agents with a brief description of each.



1.2.2 Navigation

The left sidebar provides access to all sections of the Intelligence Hub:

- **Home:** The main landing page displaying the available agents.
- **Gurobot:** The AI-powered Gurobi support assistant.
- **Explainer:** The optimization analysis agent (infeasibility diagnosis, feasibility restoration, sensitivity analysis).
- **Modeler:** The modeling assistant that turns a business problem into a complete `gurobipy` project.
- **API Keys:** Manage API keys for programmatic access.
- **MCP:** Monitor and manage your *MCP server* sessions.
- **Settings:** Manage your data and delete chats.

Click on an agent card or its sidebar icon to launch the agent. Each agent card includes a **Read more** link for additional details and a **Launch** button.

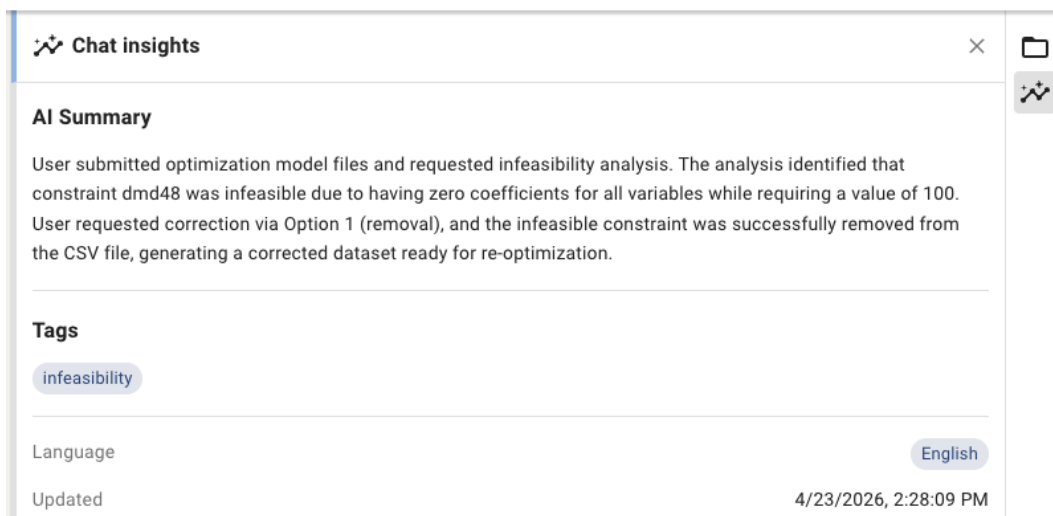
1.2.3 Insights

Each conversation has a **Chat insights** panel that you can open from the icon on the right sidebar. It provides an at-a-glance view of the conversation, including:

- **AI Summary:** A concise, automatically generated summary of what was discussed, what was uploaded, and what was resolved.
- **Tags:** Topical labels (such as **infeasibility**) that classify the conversation.
- **Language:** The language detected for the conversation.
- **Updated:** The timestamp of the most recent activity.

Note

Insights are generated in the background, so it may take some time for the data to appear or refresh after new activity.

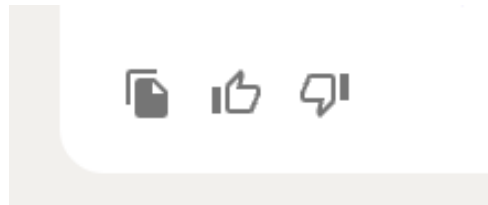


1.2.4 Daily Usage

A daily usage indicator is displayed at the bottom of the home page. It shows the percentage of the daily token limit consumed across all agents, along with the exact token count and total allocation. Usage resets daily. If you exceed your daily limit, access to the agents may be temporarily suspended until the next reset.

1.2.5 Feedback

Every agent response carries a pair of feedback controls—a **thumbs up** and a **thumbs down**—that let you tell us whether the answer was helpful.



Clicking either one opens a short form where you can add a comment explaining what was good or what fell short. The comment is optional, but the more specific you are—what you expected, what was missing or wrong, what worked well—the more useful your feedback becomes.

Gurobi’s mathematical optimization experts review this feedback and use it to improve the knowledge base and the agents’ instructions and processes. Please use it liberally—both positive and negative. Your interactions are never used to train large language models, and the optimization model files you submit are not added to Gurobi’s model library or used for solver benchmarking (see the [Terms of Service](#)).

1.3 Tools

The agents share a common set of tools that shape their behavior across every conversation: a curated **knowledge base** for retrieving authoritative documentation, a **coding guide** that captures Gurobi’s API best practices, a per-conversation **workspace** for uploaded and generated files, and a sandboxed **code execution** environment for running Python.

1.3.1 Knowledge Base

All three agents are backed by a Gurobi-curated knowledge base—a vector database that indexes content authored or vetted by Gurobi experts. Whenever you ask a question, the agent retrieves the most relevant passages and uses them to ground its response, with citations back to the original sources.

The knowledge base is updated daily and currently includes:

- **Expert articles** written and reviewed by Gurobi optimization specialists.
- **Reference manuals** for the Gurobi Optimizer and related products.
- **Code from official Gurobi GitHub repositories**, including the example models and integration libraries.
- **Modeling and performance best practices** drawn from Gurobi’s training material and customer-facing guidance.
- **Public content from gurobi.com**, such as blog posts and product pages.

This curated retrieval layer is a key differentiator over a stand-alone LLM:

- **Grounded answers.** Responses are anchored in documents Gurobi has reviewed, so you get guidance that reflects how the products actually behave—not generic advice scraped from the open web.
- **Reduced hallucinations.** When the agent has authoritative passages to cite, it is far less likely to invent APIs, parameters, or behaviors that do not exist.
- **Always current.** Daily indexing means new articles, parameter changes, and recent releases are picked up quickly. The knowledge base covers the gap between an LLM’s training cutoff and what shipped this week.
- **Traceable citations.** Every response can link back to the underlying document, so you can verify the source and read further if needed.
- **Consistency across agents.** Gurobot, Explainer, and Modeler all draw from the same vetted corpus, so their recommendations stay aligned with Gurobi’s official guidance.

1.3.2 Coding Guide

Whenever an agent writes or reviews `gurobipy` code, it consults a dedicated **coding guide** maintained by Gurobi. The guide compiles, in a single reference, how to use the Gurobi API correctly—covering the common pitfalls our experts see in customer code as well as the patterns they recommend for clear, performant, and maintainable models.

The guide is built from the ground up by Gurobi optimization specialists and is continuously refined based on:

- **Recurring issues** observed in support tickets and customer code reviews.
- **Best practices** for model building, parameter tuning, and performance.
- **API usage patterns** that take full advantage of `gurobipy` and `gurobipy-pandas`.

Note

The coding guide is a work in progress. It currently focuses on `gurobipy`, and we may extend it to other Gurobi APIs and languages, as well as additional programming styles, in future releases.

1.3.3 Workspace

Each agent conversation has its own workspace—a storage area for files that are uploaded by you or generated by the agent during the session. Files in the workspace persist across messages within the same conversation and remain accessible when you revisit the chat later.

All three agents—Gurobot, Explainer, and Modeler—use the workspace. Files you upload (such as ILP files, source code, or documentation) are stored in the workspace so the agent can reference them throughout the conversation. Files generated by the agent are also saved there: infeasibility reports from Explainer, code files and execution logs from Gurobot, and specifications, `gurobipy` implementations, `pytest` test suites, and mathematical formulations from Modeler.

Generated files appear as **Attachments** in the chat message. For each file you can:

- **Copy** the content to your clipboard.
- **Download** the file for offline use.

Clicking on an attachment opens it in a side panel next to the conversation, so you can review the file content while continuing to chat.

See the [Gurobot](#), [Explainer](#), and [Modeler](#) pages for details on the types of files each agent generates.

1.3.4 Code Execution

Agents can run code on your behalf—for example, to verify a `gurobipy` script generated by Gurobot or to run the `pytest` suite produced by Modeler. Code is executed in an isolated, dedicated environment that is created fresh for each chat.

Available runtime and packages. Python 3 is available with the following pre-installed packages:

- `gurobipy` — Gurobi Python API for building and solving optimization models.
- `gurobipy-pandas` — Gurobi and pandas integration helpers.
- `numpy` — numerical computing.
- `pandas` — data manipulation and analysis.
- `scipy` — scientific computing (optimization, linear algebra, statistics).
- `pytest` — test framework; used to run `.py` test files from the workspace.

Limitations.

- Additional packages cannot be installed.
- No internet access.
- Compute Server and Instant Cloud cannot be used.
- The bundled license is limited to 2000 variables and 2000 constraints.
- CPU and memory resources are very limited, so unit tests must be small in size.
- Execution time is limited to 60 seconds per command.

1.4 How the Intelligence Hub Compares to a General-Purpose AI Assistant

General-purpose assistants such as ChatGPT and Claude are highly capable, and they continue to add features very rapidly. The Gurobi Intelligence Hub is not a replacement for those tools—it is a specialized environment purpose-built for mathematical optimization with Gurobi, with concrete advantages where it matters most:

- **Expert knowledge, curated by Gurobi.** Every agent draws on a vector *knowledge base* that indexes expert articles, reference manuals, official Gurobi GitHub repositories, modeling and performance best practices, and content from `gurobi.com`—written, reviewed, and maintained by Gurobi optimization specialists. Responses cite the underlying sources, so you always know where the answer came from.
- **Always up to date, beyond the LLM cutoff.** The knowledge base is re-indexed daily, so new articles, parameter changes, and recent product releases are typically reflected within a day. This closes the gap between an LLM’s training cutoff and what shipped this week—and ensures the answers you get reflect the latest Gurobi version.
- **Expert-driven continuous improvement.** The *feedback* you leave—the thumbs up / thumbs down ratings and comments on responses—is reviewed by Gurobi’s mathematical optimization experts. They act on it to keep the agents sharp: adding new articles to the knowledge base, correcting and clarifying existing ones, and updating the agents’ instructions and processes. Your interactions are never used to train large language models, and the optimization model files you submit are not added to Gurobi’s model library or used for solver benchmarking.
- **Built-in optimization analysis workflows.** *Explainer* invokes Gurobi’s own algorithms—**IIS** for infeasibility diagnosis, **FeasRelax** for feasibility restoration, and basis-derived **sensitivity analysis**—directly on your model, and runs them through dedicated, specialized scripts that encode Gurobi’s diagnostic best practices. You get results computed by the solver itself, structured the way a Gurobi expert would present them.

- **From business problem to working model, end to end.** *Modeler* co-develops a structured specification with you, generates a complete `gurobipy` implementation and a `pytest` suite, executes the tests, iterates on the implementation, and halts to surface genuine inconsistencies between the specification, the code, and the tests—a full modeling loop in a single conversation.
- **A Gurobi runtime ready out of the box.** Each conversation comes with a sandboxed Python environment that already has `gurobipy`, `gurobipy-pandas`, `numpy`, `pandas`, `scipy`, and `pytest` installed, plus a limited Gurobi license available.
- **Connected to your Gurobi license and support.** *Gurobot* looks up your current Gurobi license and answers questions specific to Gurobi Optimizer, Compute Server, Cluster Manager, and Instant Cloud. From any chat, you can open a formal support request that is automatically classified, summarized, and routed to the Gurobi Expert Team with the conversation attached.

The Intelligence Hub is best understood as a specialized companion to a general-purpose assistant: it brings Gurobi’s curated content, solver routines, license context, and support workflow into a conversational interface, purpose-built for mathematical optimization.

GUROBOT

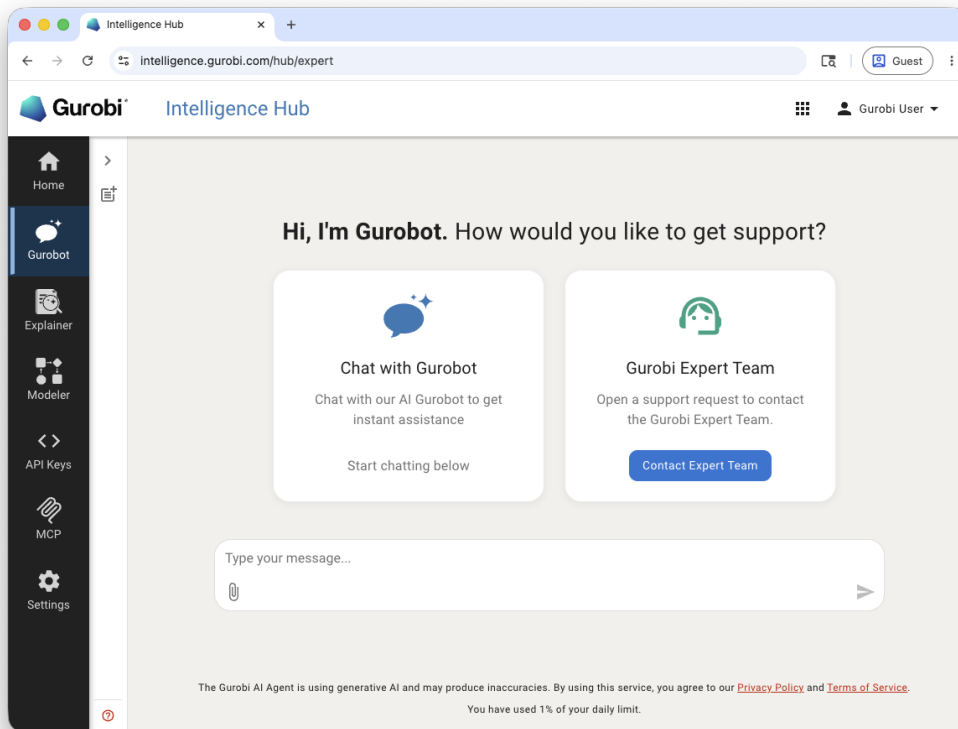
Gurobot is an AI-powered assistant that delivers instant, expert-level Gurobi support. It is designed to help you resolve issues faster, access expert guidance, and improve your overall experience with Gurobi products. Gurobot provides personalized, context-aware answers, code snippets, and documentation links backed by Gurobi's knowledge base.

2.1 Features

- **Smart Documentation Links:** Gurobot searches a curated knowledge base of up-to-date Gurobi documentation and examples, and can supplement its answers with web searches when needed. Responses include direct links to the most relevant documentation.
- **Built-in Best Practices:** Receive guidance grounded in content backed by Gurobi experts.
- **Product and License Support:** Ask questions about Gurobi Optimizer, Compute Server, Cluster Manager, and Instant Cloud. Gurobot can also look up your current license information.
- **Contextualized Responses:** Get personalized responses that take your licensing information and conversation history into account.
- **Modeling Questions:** Get quick answers to ad-hoc modeling questions— how to express a particular constraint, when to use one variable type over another, how to interpret solver output. For building a complete optimization project from a business problem, use the dedicated *Modeler* agent.
- **Code Generation and Review:** Generate, review, rewrite, and convert optimization code across languages, with Python and `gurobipy` as the default. Gurobot follows language-aware coding best practices and can run code to verify correctness.
- **Easy Escalation:** Turn any chat into a formal support request when human assistance is needed.

2.2 Using Gurobot

To open Gurobot, click **Gurobot** in the left sidebar or select the Gurobot card from the home page. The welcome screen presents two options:



- **Chat with Gurobot:** Start a conversation with the AI assistant for instant help with optimization questions, code, and documentation.
- **Gurobi Expert Team:** Open a support request to contact the Gurobi support team directly. Commercial users can use this to go straight to human support if they prefer.

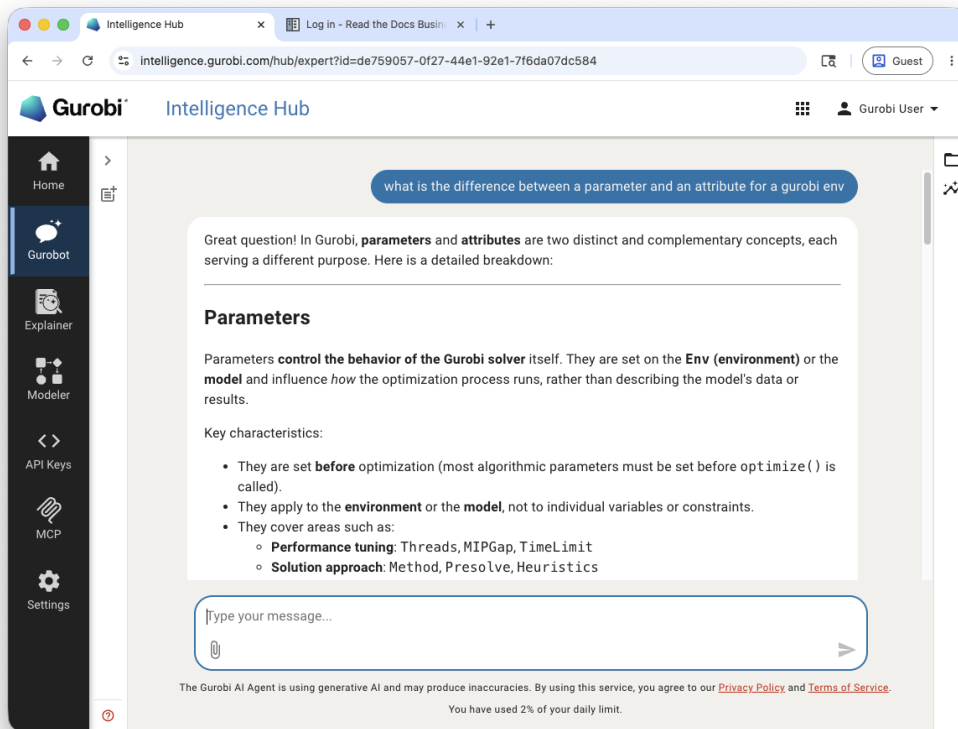
2.2.1 Starting a Chat

1. Type your message in the text field at the bottom of the screen.
2. You can attach files using the attachment icon next to the text field.
3. Press the send button or hit Enter to submit your message.

Gurobot responds with context-aware answers that may include:

- Explanations and recommendations
- Code snippets in Python, Java, C++, and other supported languages
- Links to relevant Gurobi documentation
- Best practices for modeling and optimization

Gurobot can also generate files as part of its responses—for example, complete `gurobipy` scripts, execution logs, or optimization reports. These files are saved to the conversation *workspace* and appear as attachments that you can open, copy, or download. See the *Workspace* section in the Overview for details.

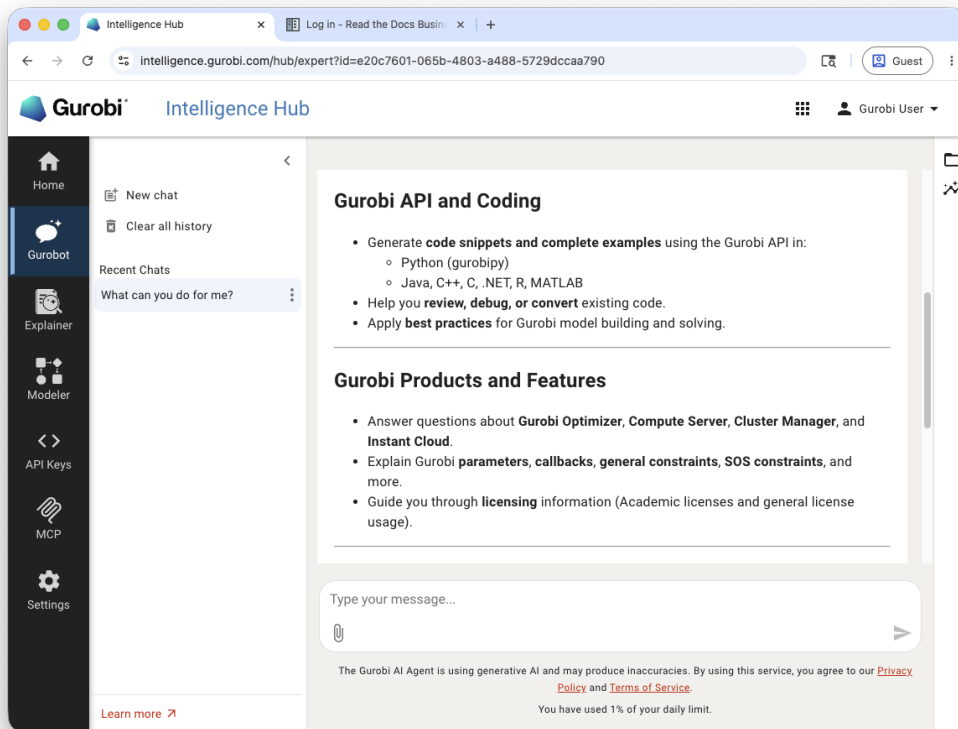


Each response includes action buttons to:

- **Copy** the response text
- **Thumbs up / Thumbs down** to rate the quality of the answer
- **Create a support request** to escalate the conversation to the Gurobi support team

2.2.2 Managing Conversations

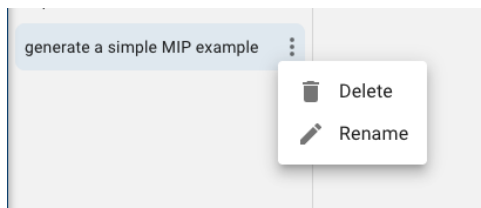
The sidebar panel provides conversation management features:



- **New chat:** Start a fresh conversation.
- **Clear all history:** Remove all previous conversations.
- **Recent Chats:** Access your conversation history. Each chat is labeled with the first message or topic discussed.

Click the sidebar toggle arrow to expand or collapse the conversation panel.

Each chat in the sidebar has a ... menu that lets you **Rename** or **Delete** the conversation.



If the chat is associated with a support request, a confirmation dialog warns you that deleting the chat will not delete the support request but you may lose context related to it.

Delete chat?

This chat will be permanently deleted and cannot be undone.

This chat is associated with support request [#118538](#). Deleting the chat will not delete the support request, but you may lose context related to it.

Are you sure you want to delete "is creation of env thread safe"?

CANCEL

DELETE

2.2.3 Support Escalation

If Gurobot cannot fully resolve your issue, you can escalate to human support directly from the chat. Click the **Create a support request** link below any response, or use the **support request** link displayed beneath the message input field.

When you create a support request, a dedicated AI agent automatically classifies and summarizes your conversation. The resulting ticket is pre-populated with a suggested category, title, and description that you can review and edit before submitting. A reference to the original chat is attached to the ticket so the support team has full context.

Create a Support Request

Support Category

Programming interface

Title

Thread Safety of Gurobi Environment Creation

Description

This support request is created on behalf of the user:

The user asked whether the creation of a Gurobi environment is thread-safe.
Initial question: is creation of env thread safe

☐ Our production system is down

Check only for EMERGENCY production system interruptions affecting end users (not developers)

BACK TO CHAT

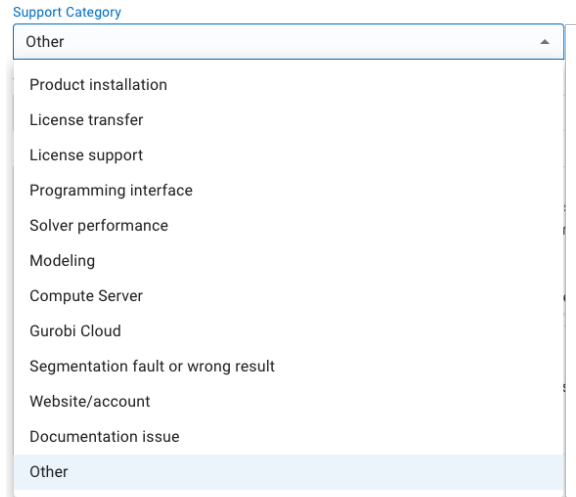
SUBMIT

You can click **BACK TO CHAT** to return to Gurobot, or **SUBMIT** to send the request.

The escalation experience differs depending on your license type.

Commercial Users

- You can open a support request at any point during the conversation, or go directly to support using the **Gurobi Expert Team** option on the Gurobot welcome screen if you prefer.
- The support request form includes a **production flag** with the label “Our production system is down”. Check this only for emergency production system interruptions affecting end users.
- The available support categories are:



A screenshot of a web form titled "Support Category". It features a dropdown menu with "Other" selected. The menu is open, showing a list of categories: Product installation, License transfer, License support, Programming interface, Solver performance, Modeling, Compute Server, Gurobi Cloud, Segmentation fault or wrong result, Website/account, Documentation issue, and Other. The "Other" option at the bottom is highlighted in light blue.

Support Category
Other
Product installation
License transfer
License support
Programming interface
Solver performance
Modeling
Compute Server
Gurobi Cloud
Segmentation fault or wrong result
Website/account
Documentation issue
Other

Academic Users

- You are encouraged to describe your issue to Gurobot first so the conversation can be summarized in the ticket.
- Academic support is limited to the following categories:
 1. Installation issues
 2. License key assistance
 3. Bug reports

For all other questions, refer to the [Gurobi Knowledge Base](#) and the [community discussion forum](#).

- If you have not yet described the issue, you can still open a support request and fill in the category, title, and description manually.
- The academic support request form does not include a production flag.
- Academic Site License holders are directed to the standard support portal to submit their request.

EXPLAINER

Explainer helps you diagnose and understand optimization models faster by combining automated model analysis, Gurobi’s curated knowledge base, and the business context you provide. It offers three predefined use cases: diagnosing infeasibility, restoring feasibility, and analyzing model sensitivity.

Business context is a first-class input. The same constraint may be non-negotiable in one setting and easily adjusted in another—a physical law or safety limit cannot move, while a target inventory level or a staffing preference often can. By telling Explainer which constraints and variables reflect hard realities and which are policy choices, you help the agent rank the candidates that are actually worth modifying and skip the ones you cannot or will not change.

This is why it is worth uploading the supporting material you already have: business-context documents, specification or design documents, and even partial source code. The more Explainer knows about *what* the model represents and *why* each piece is there, the more targeted and trustworthy its diagnosis and remediation suggestions will be.

3.1 Features

- **Three Predefined Use Cases:** Diagnose infeasibility with the IIS, restore feasibility with FeasRelax, and quantify how much your inputs can shift before the optimum changes with sensitivity analysis—each driven by Gurobi’s automated routines under the hood.
- **Easy Model Ingestion:** Upload a model file in plain text or compressed form for any use case. For infeasibility diagnosis, Explainer computes the IIS automatically from the model file. If the model is too large or the computation would take too long, you can generate the ILP file locally and upload it directly instead.
- **Context-Aware Analysis:** Attach supporting files—source code, specifications, or domain documentation—so the agent interprets your model in the context of your actual business problem and identifies which parts are realistic to change.
- **Targeted Clarification:** Explainer asks precise follow-up questions to fill gaps, sharpen its analysis, and avoid generic recommendations.
- **Gurobi-Backed Analysis:** Leverage Gurobi’s deep optimization expertise and curated knowledge base to surface patterns and insights that general-purpose tools miss.
- **Structured, Actionable Output:** Get a ranked, human-readable breakdown of root causes, repair plans, or sensitivity ranges—tied to both your model and your use case—with concrete next steps you can apply immediately. For infeasibility diagnosis, Explainer also produces a downloadable Markdown report saved to the workspace.

3.2 Background

3.2.1 Infeasibility

A mathematical optimization model is **infeasible** when no solution exists that satisfies all of its constraints and variable bounds simultaneously. This typically happens when two or more requirements contradict each other—for example, a constraint that demands a variable be at least 100 combined with a bound that limits the same variable to at most 50. Infeasibility can also arise from data errors, overly restrictive business rules, or incorrect model formulations.

Diagnosing infeasibility manually can be extremely difficult, especially in large models with thousands of constraints. Gurobi provides built-in tools to help isolate the source of the conflict.

3.2.2 Irreducible Inconsistent Subsystem (IIS)

When Gurobi detects that a model is infeasible, it can compute an **Irreducible Inconsistent Subsystem (IIS)**—the smallest subset of constraints and variable bounds that still forms an infeasible system. Removing any single element from the IIS would make the remaining subset feasible, which means every element in the IIS contributes to the conflict.

The IIS is the starting point for understanding *why* a model is infeasible. By focusing on this minimal set rather than the entire model, you can efficiently identify the root cause and decide which constraints or bounds to relax, modify, or remove.

3.2.3 ILP Files

An **ILP file** is the standard file format for representing an IIS. It contains only the constraints and variable bounds that belong to the irreducible inconsistent subsystem, written in a human-readable LP format.

When you upload an infeasible model, Explainer computes the IIS automatically. If the model is too large or the IIS computation would take too long, you can generate the ILP file locally and upload it directly. You can do this using the Gurobi command-line tool, the `gurobipy` API, or other Gurobi interfaces. For example, using the command line:

```
gurobi_cl ResultFile=model_iis.ilp model.mps
```

Or programmatically in Python:

```
model.computeIIS()  
model.write("model_iis.ilp")
```

Explainer also accepts compressed ILP files in `gz`, `bz2`, `zip`, `7z`, and `xz` formats.

3.2.4 Feasibility Relaxation (FeasRelax)

Identifying *why* a model is infeasible is only half the job—you usually also need to know *what to change to make it feasible again*. Gurobi's **FeasRelax** procedure answers that question. It searches for the smallest set of adjustments to constraint right-hand sides and variable bounds that restores feasibility, and reports both the adjustments and the resulting feasible solution.

You can configure FeasRelax to minimize the number of violations, the total violation, or the squared violation, and you can restrict relaxation to a specific subset of constraints or variables when only certain parts of the model are negotiable. The output is an actionable repair plan—a concrete list of which right-hand sides or bounds need to shift, and by how much—that you can review against your business context before applying any change.

3.2.5 Sensitivity Analysis

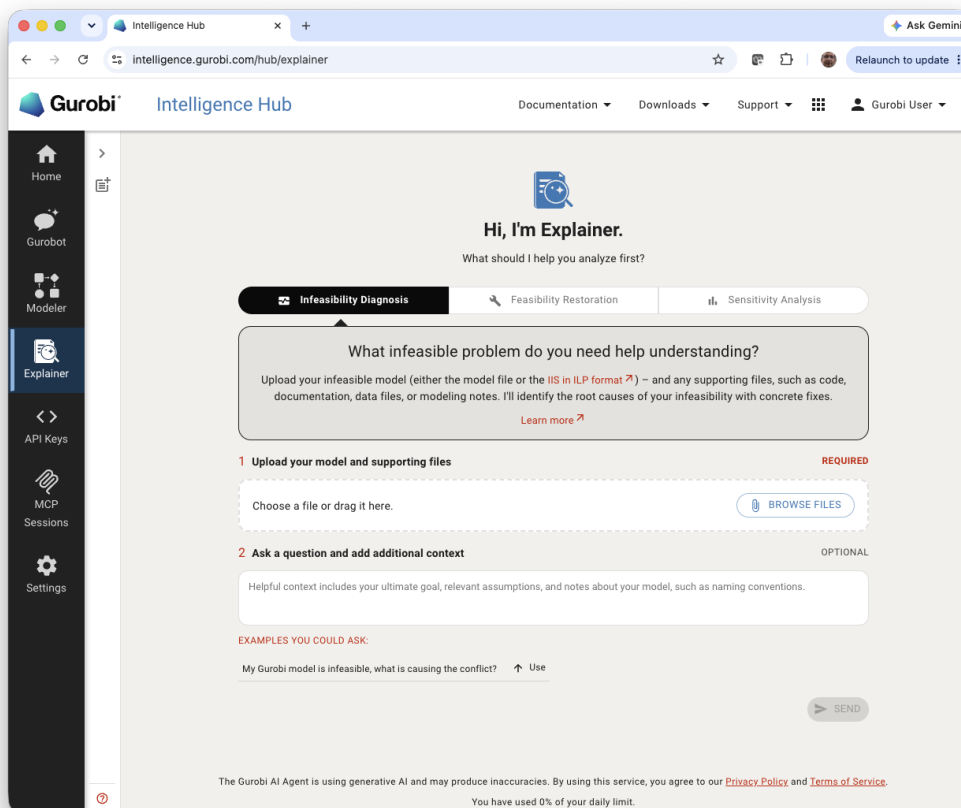
Once a linear program has been solved to optimality, **sensitivity analysis** quantifies how robust that solution is to changes in the input data. It uses information from the optimal basis to answer questions such as:

- How much can a constraint's right-hand side change before the current set of binding constraints—and the corresponding solution—changes? The associated **shadow price** measures how much the objective would improve per unit of additional capacity within that range.
- How much can a variable's objective coefficient change before that variable enters or leaves the optimal basis? For a variable currently at one of its bounds, the **reduced cost** measures the gap between its coefficient and the value at which it would become attractive enough to use.

Sensitivity analysis currently only applies to linear programs and requires a basic optimal solution. It gives you a quantitative view of where the model is tight and where it has slack, so you can focus attention on the inputs that actually move the answer.

3.3 Using Explainer

To open Explainer, click **Explainer** in the left sidebar or select the Explainer card from the home page.



Explainer provides three predefined use cases to guide your analysis. Select the one that matches your situation to get started.

3.3.1 Infeasibility Diagnosis

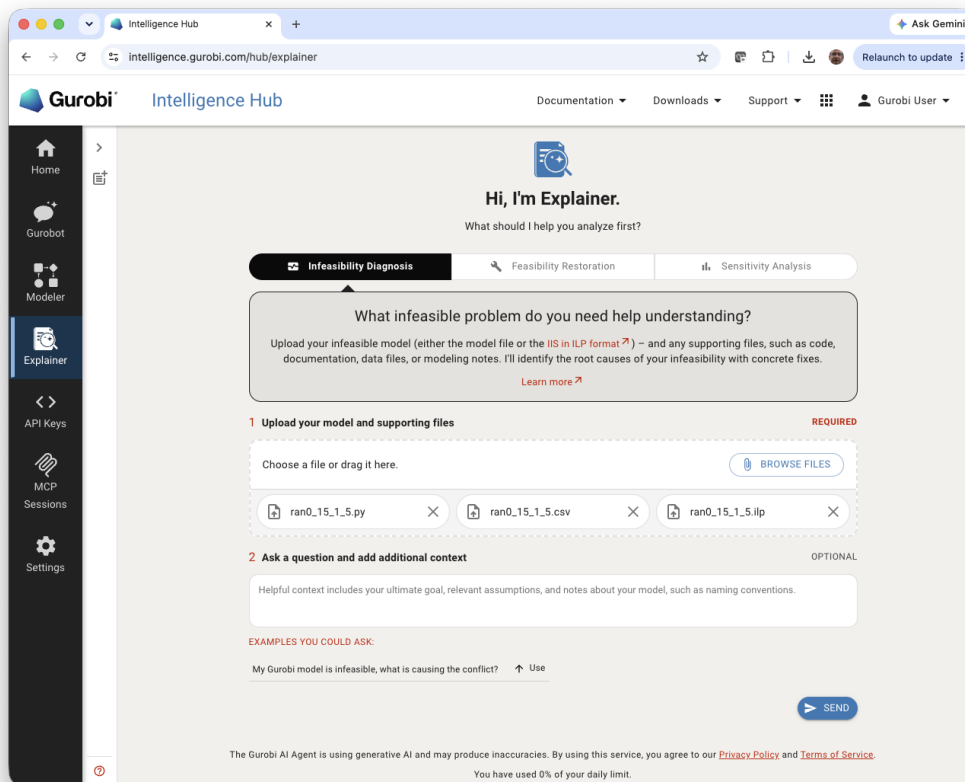
Infeasibility Diagnosis identifies why your model has no feasible solution. It analyzes the IIS from your ILP file, pinpoints the conflicting constraints and bounds, and produces a ranked report of probable root causes with concrete remediation options.

Uploading Files

Explainer accepts multiple files to provide a thorough analysis:

1. **Model file** (required): The infeasible model file. Explainer will compute the IIS automatically. Alternatively, if the model is too large or the IIS computation would take too long, you can upload a precomputed *ILP file* directly.
2. **Supporting files** (optional): Additional context files such as source code, CSV data, or domain documentation that help Explainer interpret your model.

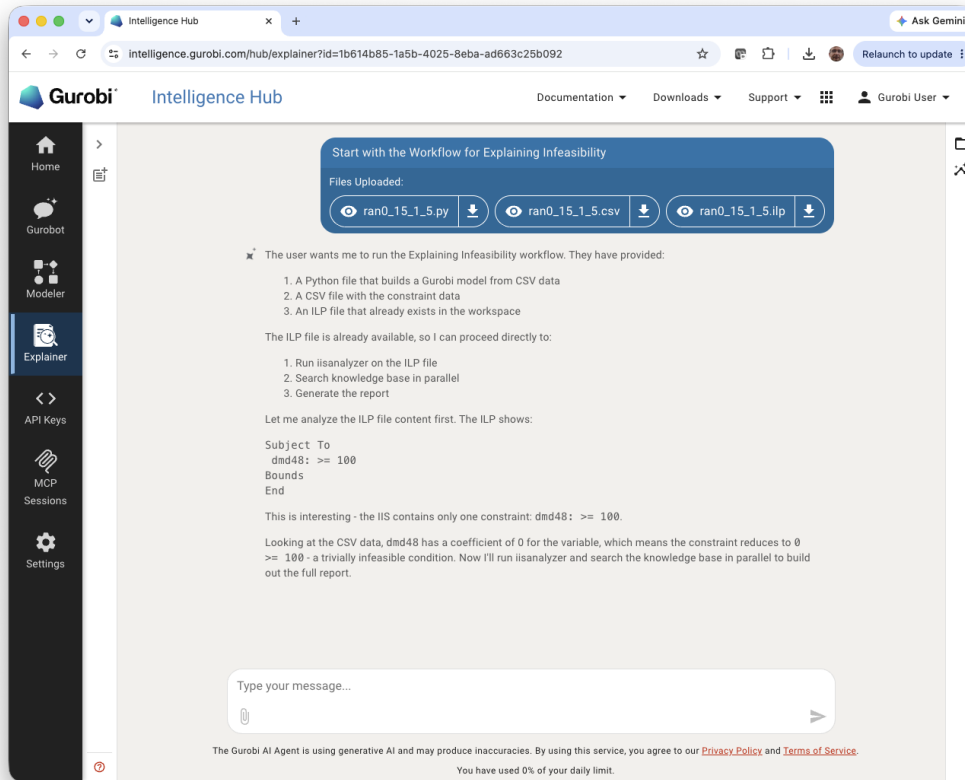
To upload files, either drag them into the upload area or click the **BROWSE FILE** button. Once your files are attached, click the **ANALYZE** button to start the analysis.



Reviewing the Analysis

Explainer processes your files and produces a detailed analysis. The response includes:

- A summary of the uploaded files and their contents
- Identification of the constraints in the IIS
- Root cause analysis explaining why the model is infeasible
- Follow-up questions to refine the diagnosis
- Links to relevant Gurobi documentation with relevance ratings



Infeasibility Report

Explainer generates a downloadable Infeasibility Analysis Report. The report is saved to the conversation *workspace* and appears as an attachment in the chat message. Click on it to open the report in a side panel alongside the conversation, or use the **Copy** and **Download** buttons to save it for offline review.

The screenshot shows the Gurobi Intelligence Hub web interface. On the left is a sidebar with navigation options: Home, Gurobot, Modeler, Explainer (selected), API Keys, MCP Sessions, and Settings. The main content area is titled 'How to Fix It' and lists three steps to resolve infeasibility: 1. Correct the coefficient, 2. Adjust the RHS, and 3. Remove the constraint. Below this is a section for 'Learn More' with links to documentation. On the right, a 'Chat files' panel shows an attached file 'infeasibility_report_ran0_15_1_5.md (4.5 KB)'. The main report area is titled 'Infeasibility Analysis Report' and includes a table for '1. IIS Overview'.

Constraint	Expression	Sense
dmd48	0	>=

Below the table, the report explains that there are zero variables involved in the IIS and provides a detailed summary of the infeasibility.

The report includes:

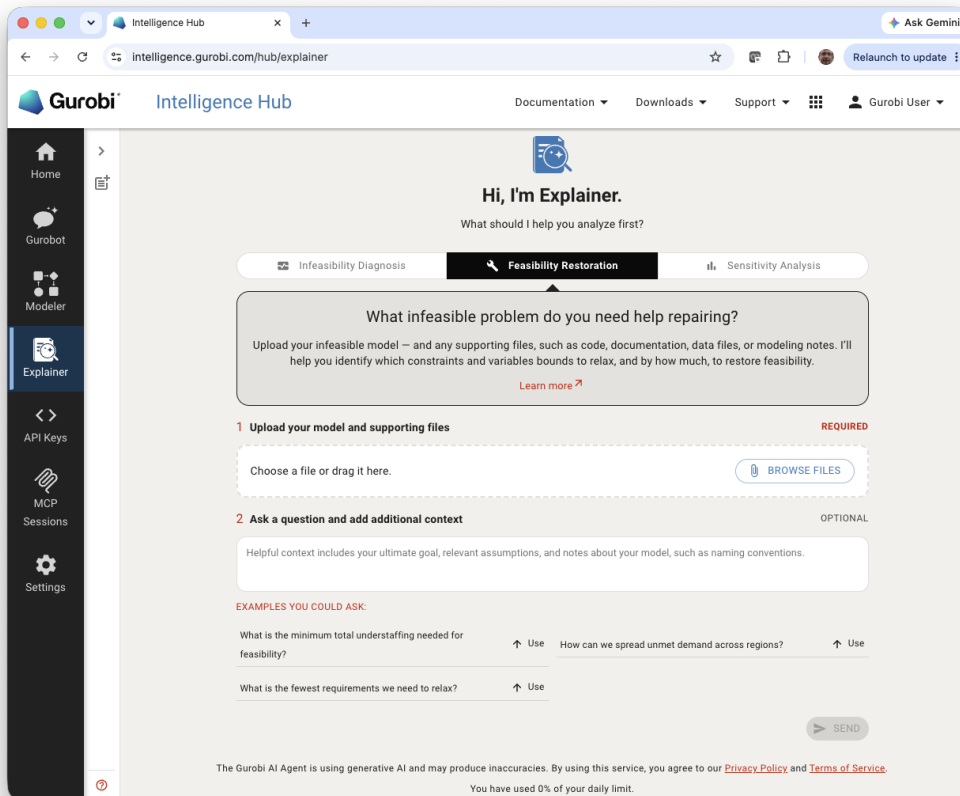
- **IIS Introduction:** A description of the constraints and variable bounds identified in the IIS, grouped by type and pattern. Notes where constraint naming could be improved for better interpretability.
- **Infeasibility Summary:** A concise, plain-language explanation of why the model is infeasible.
- **Detailed Analysis:** A mathematical walkthrough showing exactly how the identified constraints and bounds contradict each other.
- **Remediation Options:** Concrete, prioritized suggestions for resolving the infeasibility—such as adjusting right-hand-side values, relaxing variable bounds, or removing constraints—ranked by effectiveness and minimal disruption to the original model.

The response also includes follow-up questions to help refine the diagnosis and links to relevant Gurobi documentation with relevance ratings. You can continue the conversation by typing follow-up questions in the message field to get additional guidance.

3.3.2 Feasibility Restoration

Feasibility Restoration goes beyond identifying *what* is wrong—it computes the smallest concrete set of changes to your model that would make it feasible again. Rather than leaving you with a diagnosis and a blank page, it tells you exactly which constraint right-hand sides or variable bounds need to shift, and by how much, so you can evaluate the trade-offs and decide which adjustments are acceptable in your business context.

You can focus the analysis on a specific subset of constraints or variables, or allow the agent to consider the full model. The result is an actionable repair plan you can apply directly to your model.



3.3.3 Sensitivity Analysis

Sensitivity Analysis answers the question: *how fragile is my optimal solution?* Given a model that has already been solved to optimality, it measures how much each input—objective coefficients, constraint right-hand sides, and variable bounds—can change before the current solution is no longer optimal.

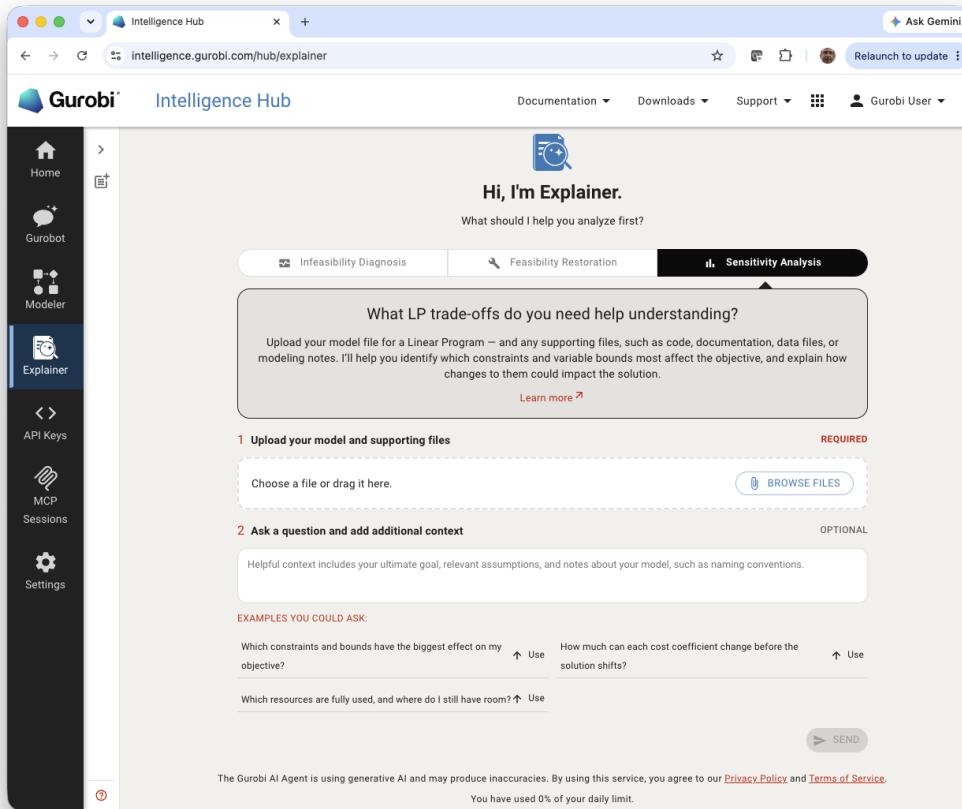
The analysis produces two complementary views:

- **Binding constraints:** which constraints are active at the optimum, how much the objective would improve per unit of additional capacity (shadow price), and over what range of the right-hand side that shadow price remains valid.
- **Variable sensitivity:** for variables that are fixed at a bound, how far the objective coefficient would need to shift before that variable would enter the solution (reduced cost and break-even point). For variables already in the solution, the range of their objective coefficient and bounds over which the current basis stays optimal.

This gives you a precise, quantitative picture of where the model is flexible and where it is tight, so you can focus attention on the constraints and decisions that matter most.

Note

Sensitivity Analysis currently only applies to linear programs (LP). It requires a basic optimal solution.



3.3.4 Managing Conversations

Like Gurobot, Explainer maintains a history of your analysis sessions. Use the sidebar panel to start a new chat or revisit previous analyses. Each chat has a ... menu that lets you **Rename** or **Delete** the conversation.

MODELER

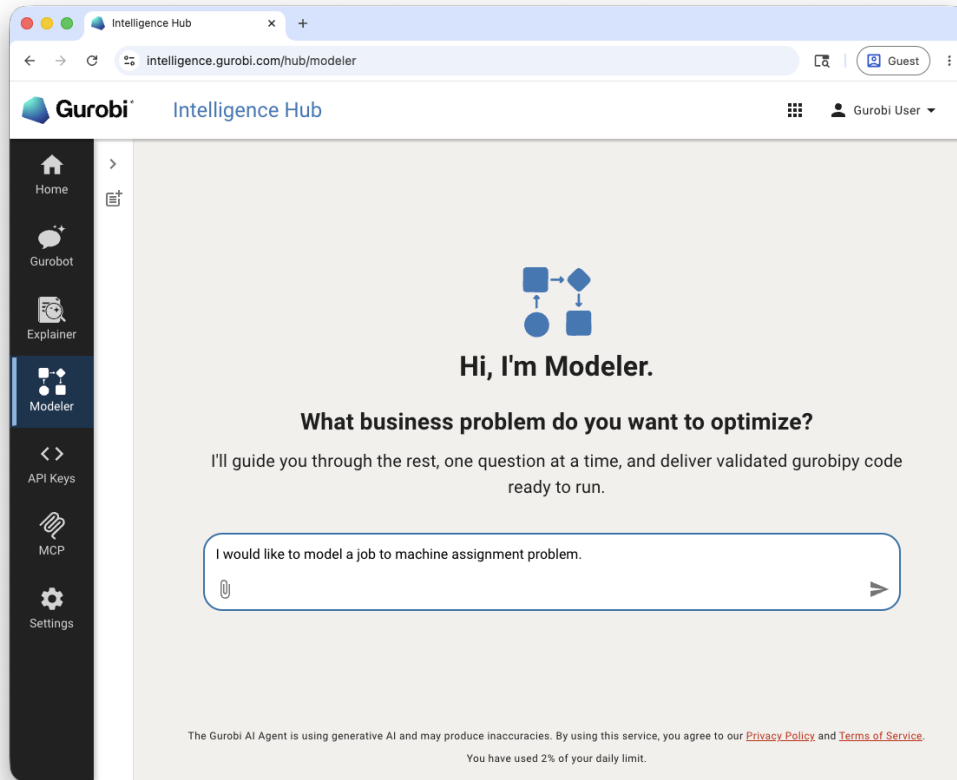
Modeler is an AI agent that guides business users and developers through mathematical optimization problem modeling and implementation using Gurobi. Through an interactive plain-language workflow, Modeler helps you co-develop and validate a business specification, then uses that specification to produce a complete optimization project: a working `gurobipy` implementation, an executable test suite, and—on demand—a mathematical formulation.

4.1 Features

- **Conversational Specification:** Modeler asks one clarifying question at a time, grounded in best practices and a knowledge base of optimization examples, and incrementally builds a structured specification of your business problem.
- **Business-First Approach:** The specification phase focuses on the business problem—objective, constraints, data, scenarios—rather than on mathematics, so non-experts can drive the conversation.
- **`gurobipy` Code Generation:** Once the specification is complete, Modeler generates a full `gurobipy` implementation that follows Gurobi coding best practices.
- **Automatic Unit Tests:** Modeler generates a `pytest` test suite covering the scenarios captured in the specification, runs the tests, and iterates on the implementation until either every test passes or it detects an inconsistency it cannot resolve on its own—in which case it stops and asks you to weigh in.
- **Mathematical Formulation on Demand:** Modeler can produce a clean LaTeX mathematical formulation (sets, parameters, decision variables, objective, constraints) consistent with the specification and the implementation.
- **Agile Updates:** When you ask for a change, Modeler updates the specification first, then propagates the change to the implementation and tests, keeping all artifacts consistent.
- **Project Check:** Point Modeler at an existing project (specification, code, tests) and it will identify gaps, fill in missing pieces, and verify that the implementation matches the specification.
- **Reimplementation and Conversion:** Modeler can recover a specification from an existing model file and reimplement it in `gurobipy` following current best practices.

4.2 Using Modeler

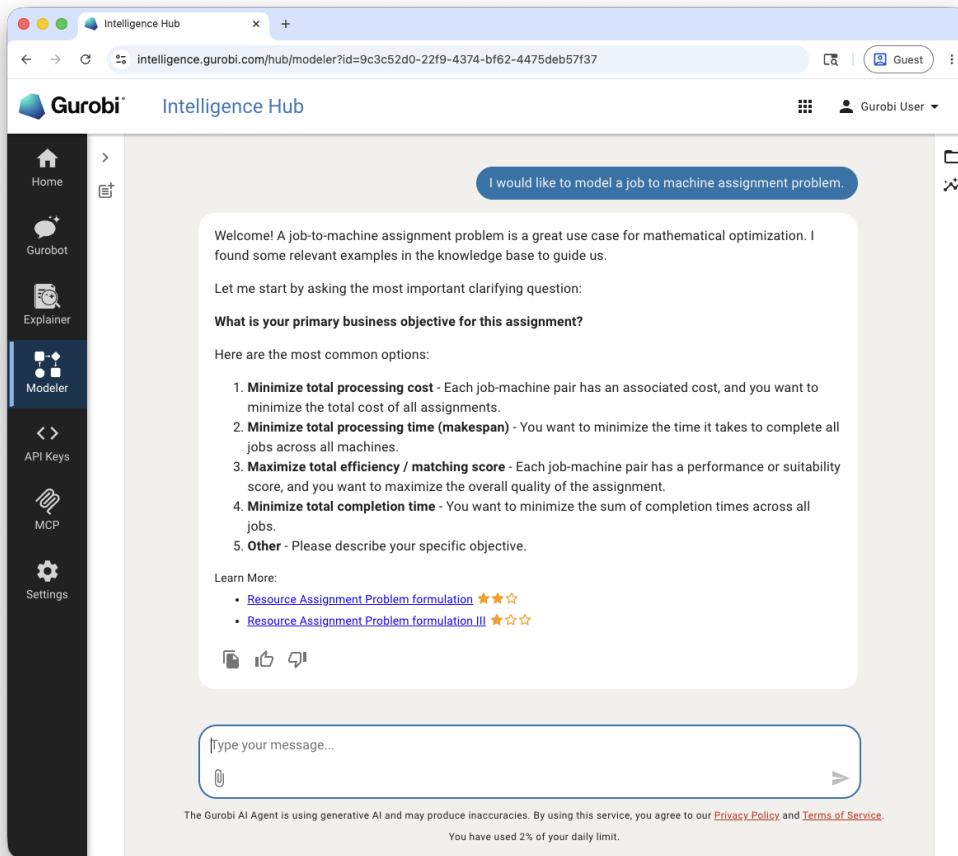
To open Modeler, click **Modeler** in the left sidebar or select the Modeler card from the home page. The welcome screen prompts you to describe the business problem you want to optimize.

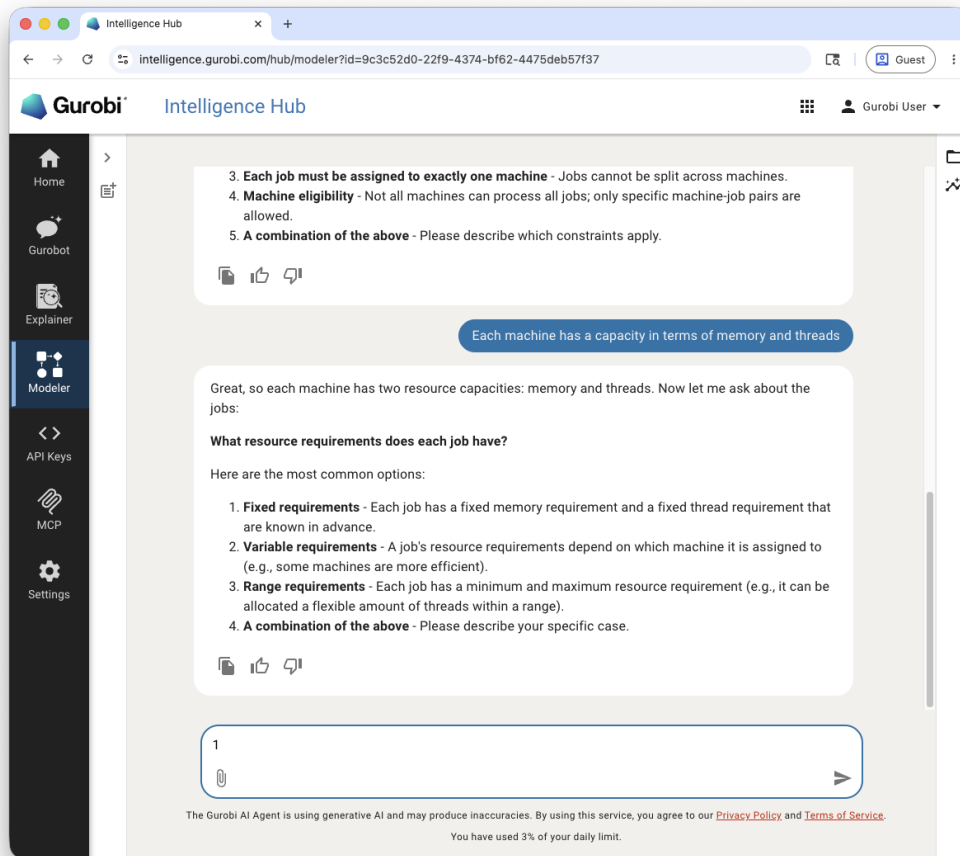


Modeler guides you through the project in a structured sequence of steps. All generated artifacts are saved to the conversation *workspace* as attachments that you can open in a side panel, copy, or download.

4.2.1 Step 1: Specification

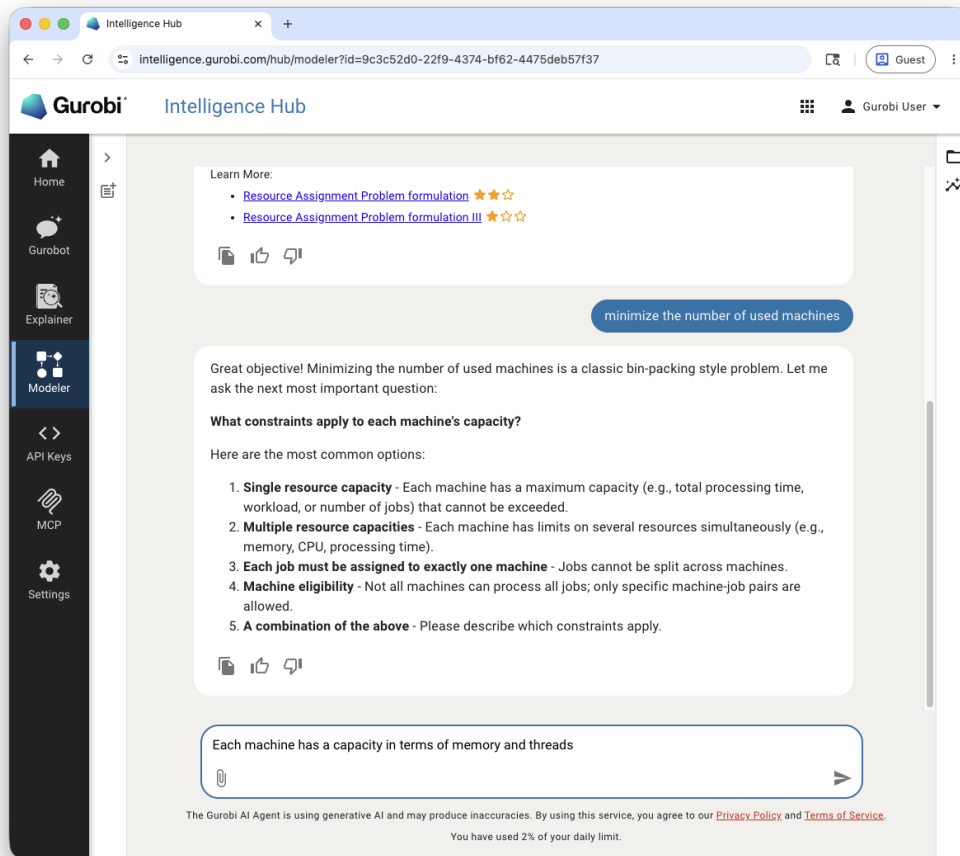
Modeler asks clarifying questions **one at a time**, starting with the most important one. Each question typically comes with an ordered list of common options. These options are drawn from examples in the knowledge base and from general knowledge about the specific problem and industry you are working in, so the choices reflect how similar situations are usually modeled. Pick the choice that matches your situation or describe your own. To select an option, just type its number—you do not need to retype the full text.





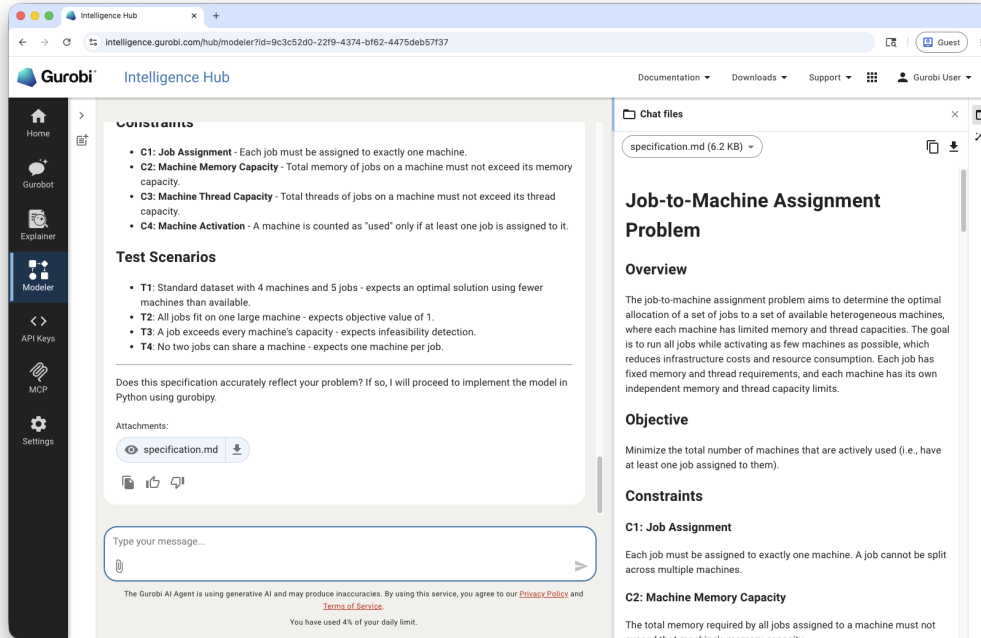
Typical questions cover:

- The **business objective** (for example, minimize cost, minimize the number of resources used, maximize throughput).
- The **constraints** that must be respected (capacity limits, exclusivity rules, assignment rules, etc.).
- The **data entities** involved and their fields.
- The **expected output** structure.



After every answer, Modeler updates the specification document, even when it is still incomplete. The specification follows a fixed structure:

1. **Overview** - a brief description of the business problem.
2. **Objective** - the business objective.
3. **Constraints** - one subsection per constraint, each with an identifier (C1, C2, ...).
4. **Input** - data entities, their fields, and types.
5. **Output** - the structure of the business solution.
6. **Test scenarios** - expected and edge cases at the business level, each with an identifier (T1, T2, ...), input data, and acceptance criteria.
7. **Out of scope** - items explicitly excluded from the model.



When the specification is complete, Modeler proposes to move on to the implementation.

4.2.2 Step 2: Implementation

Modeler generates a complete `gurobipy` implementation of the model following Gurobi coding best practices. Python with `gurobipy` is the default; you can request a different language at the start of the conversation.

The implementation is saved to the workspace as a Python file you can open, copy, or download.

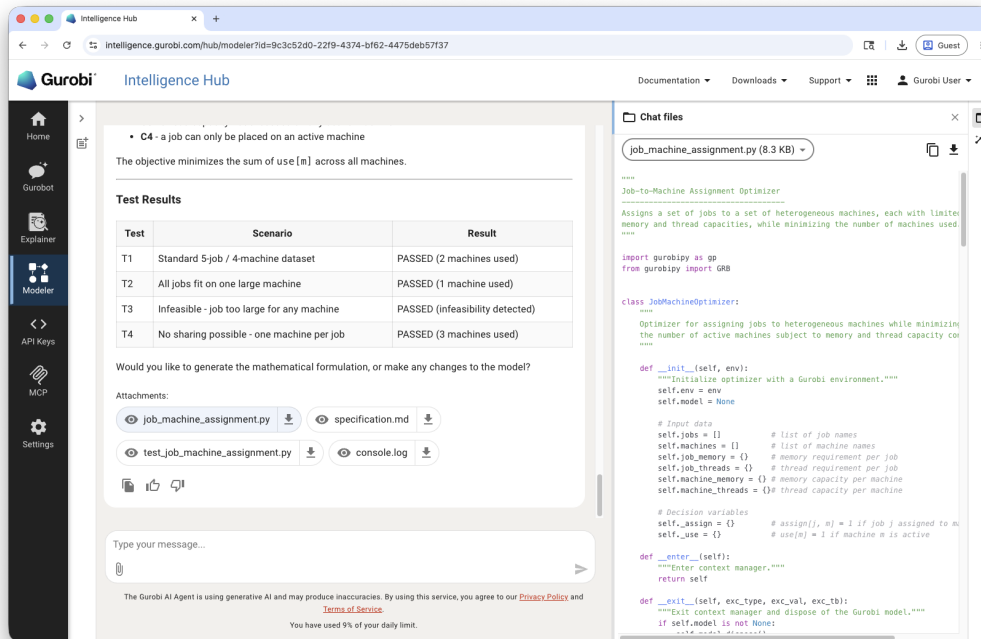
4.2.3 Step 3: Unit Tests

Modeler generates a `pytest` test suite in a separate file, with one test per scenario defined in the specification. Each test references its scenario identifier, uses small hard-coded sample data to keep execution fast, and applies a short solver time limit.

Modeler runs the tests, reads the output, and iterates on the implementation. The loop ends in one of two ways:

- **All tests pass.** Modeler resolved the failures automatically and the implementation now matches every scenario in the specification.
- **An inconsistency is detected.** If Modeler concludes that a failure reflects a contradiction between the specification, the implementation, and the tests—rather than something it can fix on its own—it stops the loop and warns you, summarizing what it found so you can decide which side to correct.

The execution log is saved to the workspace alongside the code and tests.



4.2.4 Step 4: Agile Updates

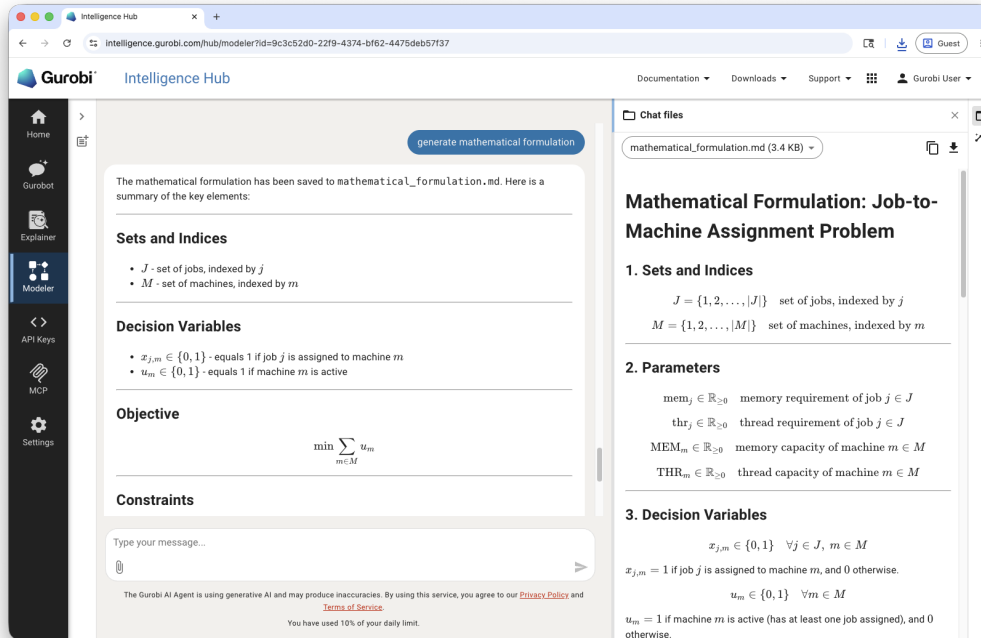
When you request a change—a new constraint, a different objective, an adjusted data structure—Modeler updates the specification first, then the implementation and tests, in that order. If it detects an inconsistency between the existing specification and implementation, it asks you to confirm which side is correct before making any change.

4.3 Generating a Mathematical Formulation

At any point you can ask Modeler to generate a mathematical formulation of the model. Modeler produces a Markdown document with LaTeX expressions covering:

1. Sets and indices
2. Parameters
3. Decision variables
4. Objective function
5. Constraints
6. Complete model formulation
7. Additional notes

The formulation is consistent with the current specification and implementation, and is saved to the workspace.



4.4 Checking an Existing Project

If you upload the files of an existing optimization project (specification, code, tests), you can ask Modeler to **check the project**. Modeler will:

1. Verify that the specification is complete; if not, ask clarifying questions to complete it.
2. Generate the implementation and basic tests if they are missing.
3. Verify that the implementation matches the specification—the specification takes precedence—and update the code or tests where they diverge.
4. Run the tests and fix any failures.

4.5 Reimplementing or Converting a Model

You can also ask Modeler to reimplement an existing model from a source file—for example, to convert a model from another language into `gurobipy` or to refresh an older implementation. Modeler will recover a specification from the original file, ask you to confirm it, then generate a new implementation and tests that are functionally equivalent to the original.

4.6 Managing Conversations

Like Gurobot and Explainer, Modeler maintains a history of your sessions. Use the sidebar panel to start a new chat or revisit previous projects. Click **Clear all history** to delete every Modeler conversation, or use the **...** menu next to a chat to **Rename** or **Delete** it individually.

MCP SERVER

The Intelligence Hub provides a Model Context Protocol (MCP) server that runs locally on your machine and connects to the Intelligence Hub backend. The MCP server exposes Gurobot, Explainer, and Modeler to MCP-compatible tools such as IDE extensions and AI coding assistants, so you can use them without leaving your development environment.

5.1 Design Rationale

The MCP server is delivered as a Docker container running locally on your machine. This is a deliberate design choice that addresses several limitations of the MCP protocol and provides concrete benefits over a purely cloud-hosted or in-process approach.

Simple installation and automatic updates. Docker is a single, widely available prerequisite. Pulling the latest image is all you need to pick up improvements and fixes—no package managers, version conflicts, or manual upgrades.

Working with large and binary files. MCP passes all data through the agent context window. This works well for short questions and lightweight integrations, but it cannot handle large files or binary formats. For mathematical optimization work, the ability to read model files (.lp, .mps) and compressed archives is essential. Because the container runs locally, it can read input files directly from your filesystem and write generated files—reports, scripts, formulations—back to it, without routing any of that content through the MCP context.

Preserving output fidelity. Writing generated files directly to your local filesystem means the output from the Gurobi agents reaches you exactly as produced. If files were returned through MCP, the client-side AI assistant would need to relay or reconstruct them, introducing the risk of paraphrasing or altering the content.

Controlled filesystem access. The container can only access the root directory you explicitly mount, along with everything beneath it. It cannot read or upload files from the rest of your system—even if an agent were somehow misdirected. This gives you a clear, auditable boundary around what the MCP server can touch.

Credential isolation. Your Gurobi Intelligence credentials are passed to the container as environment variables and used only inside it. They are never written into a project configuration file or MCP client config that could be accidentally committed or shared.

Standard observability. Because the MCP server runs as a named Docker container, its logs are available through Docker's standard interfaces—either from the command line with `docker logs -f gurobimcp` or in the Docker Desktop UI. This gives you a straightforward way to trace what happened during a session and monitor agent interactions without any additional tooling.

5.2 How It Works

The MCP server runs as a Docker container on your local machine, acting as a bridge between your MCP-compatible client and the Intelligence Hub backend:

1. The MCP server starts as a Docker container from the `gurobi/mcp` image on Docker Hub.
2. It authenticates with the Intelligence Hub using credentials (Access ID and Secret, or license file) generated from an *API key*.
3. MCP-compatible clients connect to the container and gain access to the Intelligence Hub agents.

Running the server as a container provides a controlled, secure environment and makes it easy to stay up to date—pulling the latest image is all you need to get new features and fixes.

5.3 Prerequisites

Before setting up the MCP server, you need:

- [Docker](#) installed and running on your machine.
- An active Intelligence Hub account.
- Gurobi Intelligence credentials (Access ID and Secret, or license file). Obtain your credentials from the *API Keys* section.
- A project directory to mount for file access.

5.4 Licensing Options

You can provide your Gurobi Intelligence credentials in two ways.

5.4.1 Option 1: Using Environment Variables (Recommended)

```
$ export GRB_INTELLIGENCE_ACCESS_ID=your_access_id
$ export GRB_INTELLIGENCE_SECRET=your_secret
$ export GRB_MCP_MOUNT=$HOME/work

$ docker run -d --name gurobimcp -p 61095:61095 \
  -e GRB_INTELLIGENCE_ACCESS_ID \
  -e GRB_INTELLIGENCE_SECRET \
  -v $GRB_MCP_MOUNT:/workspace \
  -e GRB_MCP_MOUNT \
  gurobi/mcp:latest
```


5.4.2 Option 2: Using a License File

```
$ export GRB_LICENSE_INTELLIGENCE_FILE=$HOME/gurobi-intelligence.lic
$ export GRB_MCP_MOUNT=$HOME/work

$ docker run -d --name gurobimcp -p 61095:61095 \
  -v $GRB_LICENSE_INTELLIGENCE_FILE:/opt/gurobi/gurobi-intelligence.lic \
  -v $GRB_MCP_MOUNT:/workspace \
  -e GRB_MCP_MOUNT \
  gurobi/mcp:latest
```

5.5 Path Mapping

The MCP server automatically translates paths between your host system and the container:

- **Host path:** \$HOME/work/mymodel.mps
- **Container path:** /workspace/mymodel.mps (automatically mapped)
- **GRB_MCP_MOUNT:** Must match the host path being mounted

Note: Only one project directory mount is supported. If you need access to files in multiple directories, mount a parent directory that contains them all.

5.6 Usage Examples

5.6.1 Single Project (Environment Variables)

```
export GRB_INTELLIGENCE_ACCESS_ID=your_access_id
export GRB_INTELLIGENCE_SECRET=your_secret
export GRB_MCP_MOUNT=$HOME/optimization-project

docker run -d --name gurobimcp -p 61095:61095 \
  -e GRB_INTELLIGENCE_ACCESS_ID \
  -e GRB_INTELLIGENCE_SECRET \
  -v $GRB_MCP_MOUNT:/workspace \
  -e GRB_MCP_MOUNT \
  gurobi/mcp:latest
```

5.6.2 Single Project (License File)

```
export GRB_LICENSE_INTELLIGENCE_FILE=$HOME/gurobi-intelligence.lic
export GRB_MCP_MOUNT=$HOME/optimization-project

docker run -d --name gurobimcp -p 61095:61095 \
  -v $GRB_LICENSE_INTELLIGENCE_FILE:/opt/gurobi/gurobi-intelligence.lic \
  -v $GRB_MCP_MOUNT:/workspace \
  -e GRB_MCP_MOUNT \
  gurobi/mcp:latest
```

5.6.3 Windows with Docker Desktop

```
set GRB_INTELLIGENCE_ACCESS_ID=your_access_id
set GRB_INTELLIGENCE_SECRET=your_secret
set GRB_MCP_MOUNT=%USERPROFILE%\work

docker run -d --name gurobimcp -p 61095:61095 ^
  -e GRB_INTELLIGENCE_ACCESS_ID ^
  -e GRB_INTELLIGENCE_SECRET ^
  -v %GRB_MCP_MOUNT%:/workspace ^
  -e GRB_MCP_MOUNT ^
  gurobi/mcp:latest
```

Note: The server automatically handles Windows path formats and converts them to Unix-style paths within the container.

5.7 Exposed Ports

- **61095:** MCP server HTTP port (required for AI assistant communication)

5.8 Configuring AI Assistants

Once the MCP server is running, you need to configure your AI assistant to connect to it.

The MCP server is designed to work with AI development environments that support the Model Context Protocol, such as:

- Claude Desktop
- Claude Code (CLI)
- GitHub Copilot (VS Code)
- Other MCP-compatible AI assistants

5.8.1 Claude Desktop and Claude Code

Claude Code

Use the CLI to add the server:

```
claude mcp add -t http gurobi http://localhost:61095/api/v1/agent/mcp
```

After running the command:

1. Restart Claude Code
2. Look for the MCP server connection indicator in the interface
3. The Gurobi Intelligence tools should now be available in your conversations

Claude Desktop

Add the following to your Claude Desktop MCP configuration file (`claude_desktop_config.json`):

```
{
  "mcpServers": {
    "gurobi": {
      "command": "npx",
      "args": [
        "-y",
        "mcp-remote",
        "http://localhost:61095/api/v1/agent/mcp"
      ]
    }
  }
}
```

After updating the configuration, restart Claude Desktop. The Gurobi Intelligence tools will become available in your conversations.

Warning

Known issue: Due to a limitation in how Claude Desktop's Chat tab maintains MCP connections ([anthropics/claude-code#41836](#)), multiple separate chat conversations will be merged into a single session in the Gurobi Intelligence portal. We advise using Claude Code instead of the Chat tab when session separation matters, until this is resolved upstream.

5.8.2 GitHub Copilot (VS Code)

Add the following to your MCP configuration file:

- **macOS/Linux:** ~/Library/Application Support/Code/User/mcp.json
- **Windows:** %APPDATA%\Roaming\Code\User\mcp.json

```
{
  "servers": {
    "gurobi": {
      "type": "http",
      "url": "http://localhost:61095/api/v1/agent/mcp",
      "headers": {
        "Accept": "text/event-stream"
      }
    }
  },
  "inputs": []
}
```

After updating the configuration:

1. Restart VS Code
2. The Gurobi Intelligence MCP server should be available to GitHub Copilot

5.9 Updating the MCP Server

To pull the latest version of the MCP server at any time, run:

```
docker pull gurobi/mcp
```

5.10 Environment Variables

5.10.1 Required for All Deployments

- **GRB_MCP_MOUNT**: Host directory path that is mounted to `/workspace`
 - Must be an absolute path
 - Must match the source path in your `-v` volume mount
 - Example: `export GRB_MCP_MOUNT=$HOME/work`

5.11 File Handling

Once connected, the MCP server exposes Gurobot, Explainer, and Modeler as tools your MCP client can invoke. See the [Gurobot](#), [Explainer](#), and [Modeler](#) pages for what each agent does—this section covers only the parts that are specific to the MCP integration.

Files you reference in your prompt are read from the directory you mounted with `GRB_MCP_MOUNT` (see [Path Mapping](#) above). Use absolute host paths in your prompt (for example, `$HOME/work/mymodel.ilp`); the server translates them into the container's `/workspace` view automatically.

Files generated by the agents—`gurobipy` scripts and execution logs from Gurobot, infeasibility and sensitivity reports from Explainer, and specifications, implementations, and test suites from Modeler—are written to a `gurobi_generated/` directory inside your current working directory. They appear directly on your local filesystem so your MCP client can open, edit, or commit them like any other file.

The same conversation also remains accessible from the web application: open the **MCP** section in the Intelligence Hub sidebar to inspect transcripts and any artifacts produced during the session (see [Monitoring MCP Sessions](#) below).

5.12 Viewing Local Logs

The MCP server container streams its logs to Docker, so you can follow file uploads and agent reasoning steps in real time. In Docker Desktop, open the `gurobimcp` container to see the live log output.

Containers / gurobimcp

grobimcp
 027406e205c4  gurobi/mcp:latest
 61095:61095

STATUS
 Running (1 minute ago)



Logs Inspect Bind mounts Exec Files Stats

```

info : Gurobi Intelligence MCP server...
info : Platform is armlinux64 (linux) - "Distroless"
info : Version is 1.0.0 (build v13.0.2rc1-1266-gb9f179132)
info : Using license file /opt/gurobi/gurobi-intelligence.lic
info : Connecting to Gurobi Intelligence Hub at https://intelligence.gurobi.com
info : Using in-memory session store for MCP sessions
info : Using Docker volume mounts
info : Volume mount: /Users/jaczynski/mcpclaude -> /workspace
info : Public root is not accessible /resources/grb_mcp/public
info : Starting Gurobi Intelligence MCP server (HTTP) on port 61095...
info : MCP endpoint URL: http://localhost:61095/api/v1/agent/mcp
info : Available tools: gurobobot, explainer
info : server connecting
info : server session connected session_id=N2MUMVFWNQ07KQHSW3BG5TL5G
info : session initialized
info : MCP Session N2MUMVFWNQ07KQHSW3BG5TL5G - Explainer tool called with prompt length: 36, files: 1
info : MCP Session N2MUMVFWNQ07KQHSW3BG5TL5G - Using chat ID: 974f3130-4ccf-4e20-811e-b2885acb1b39
[READ] Reading input file: ran0_15_1.ilp
      Host path: /Users/jaczynski/mcpclaude/ran0_15_1.ilp
      Container path: /workspace/ran0_15_1.ilp
[ OK ] Successfully uploaded file: ran0_15_1.ilp (ID: 1bb9e38d-48fb-450c-8490-0278d65d813d)

Thinking process:

[THINK] The user wants me to analyze the ILP file 'ran0_15_1.ilp'. Let me first save it to the workspace, then run the
iisanalyzer and search the knowledge base in parallel.

The file content is already provided. Let me save it and analyze it.
[THINK] Saving the user-provided ILP file to the workspace for analysis.
[THINK] Now let me run the IIS analyzer and search the knowledge base in parallel.
[THINK] Querying the knowledge base.
```

The same output is available from the command line:

```
docker logs -f gurobimcp
```

5.13 Monitoring MCP Sessions

Every MCP interaction is recorded in the Intelligence Hub. Open the **MCP** section from the left sidebar to review your MCP sessions, including the agent used, the title of the request, token usage, length, and timestamps.

The screenshot shows the Gurobi Intelligence Hub interface for the MCP (2) section. The top navigation bar includes links for Documentation, Downloads, Support, and a user profile (Gurobi User). The left sidebar contains icons for Home, Gurobot, Explainer, Modeler, API Keys, MCP (selected), and Settings. The main content area displays a table of sessions with columns: SESSION ID, AGENT NAME, TITLE, INPUT TOKENS, OUTPUT TOKENS, LENGTH, and CREATED AT (UTC). Below the table, there are tabs for INFO, TITLE, and TOKENS. The INFO tab is active, showing details for a specific session.

SESSION ID	AGENT NAME	TITLE	INPUT TOKENS	OUTPUT TOKENS	LENGTH	CREATED AT (UTC)
974f3130	explainer	Explain the ILP file ran0_15_1_5.ilp	12,249	2,421	1	May 27, 2026 10:42 PM
9e3e3b11	explainer	explain the infeasibility in ran0_15_1_5.ilp	10,901	1,110	2	May 27, 2026 10:17 PM

Session ID	Length	Created At (UTC)	Updated At (UTC)
974f3130-4ccf-4e20-811e-b2885acb1b39	1	May 27, 2026 10:42 PM	May 27, 2026 10:43 PM

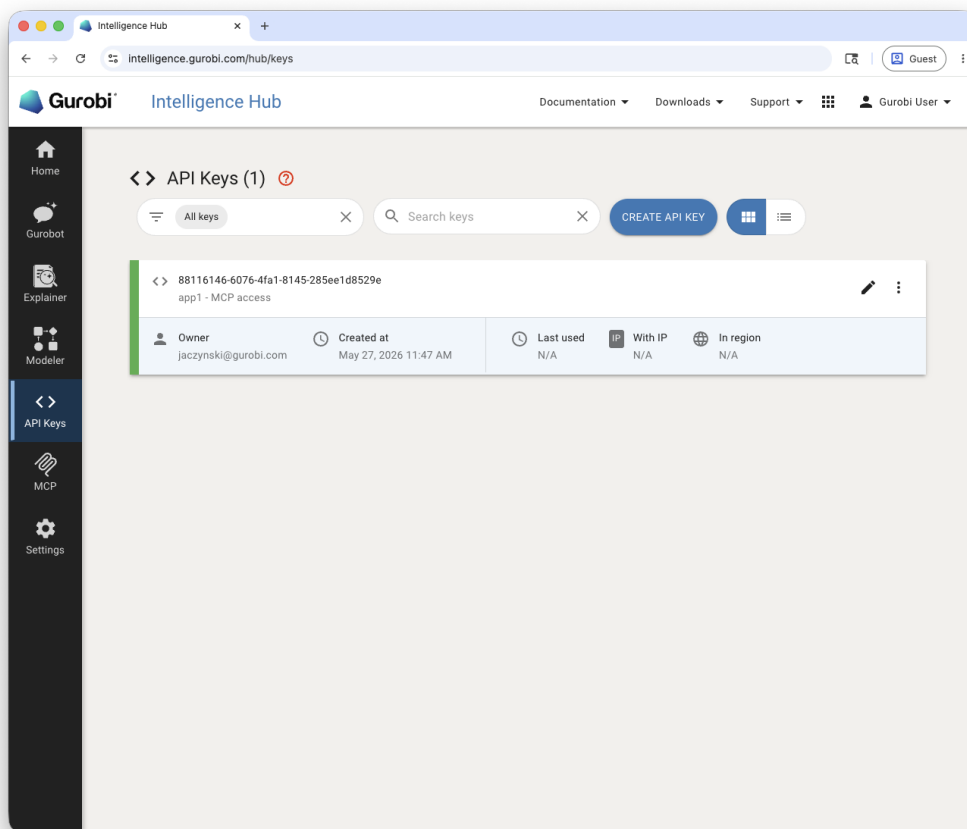
Agent Name: explainer

For each session you can:

- Open the chat view to inspect the full exchange between the MCP client and the agent, including any reports or files generated during the session.
- Review the data exchanged across the session in the **INFO**, **TITLE**, and **TOKENS** tabs at the bottom of the page.
- Delete an individual session to permanently remove its data.

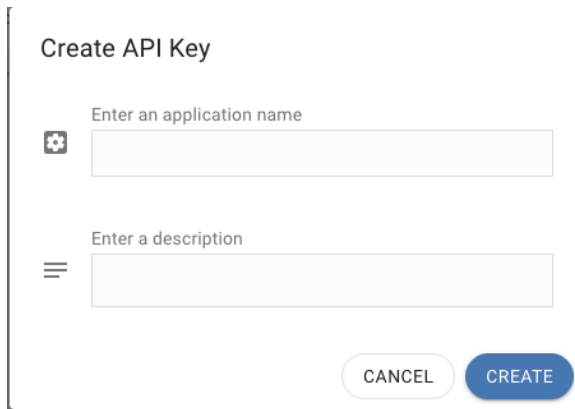
API KEYS

API keys provide programmatic access to the Intelligence Hub agents. When you create an API key, a license file is generated that contains your private API access ID and secret. This license file is used to authenticate the *MCP server* and other integrations with the Intelligence Hub backend. You can create, view, and manage API keys from the **API Keys** section of the Intelligence Hub.



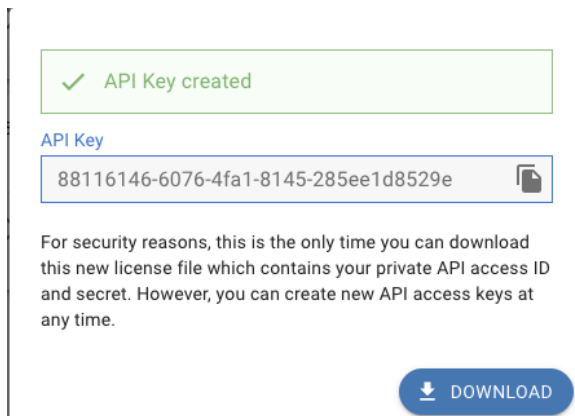
6.1 Creating an API Key

1. Click **API Keys** in the left sidebar.
2. Click the **CREATE API KEY** button.
3. In the dialog that appears, enter an application name and an optional description for the key.



The dialog box is titled "Create API Key". It contains two input fields: "Enter an application name" with a plus icon on the left, and "Enter a description" with a hamburger menu icon on the left. At the bottom right, there are two buttons: "CANCEL" and "CREATE".

4. Click **CREATE**. The new API key is displayed along with a download button.



The screen shows a green success message: "✓ API Key created". Below it, the text "API Key" is followed by a text box containing the key value "88116146-6076-4fa1-8145-285ee1d8529e" and a download icon. A warning message states: "For security reasons, this is the only time you can download this new license file which contains your private API access ID and secret. However, you can create new API access keys at any time." At the bottom right is a "DOWNLOAD" button with a download icon.

5. Copy the key or click **DOWNLOAD** to save the license file containing your private API access ID and secret.

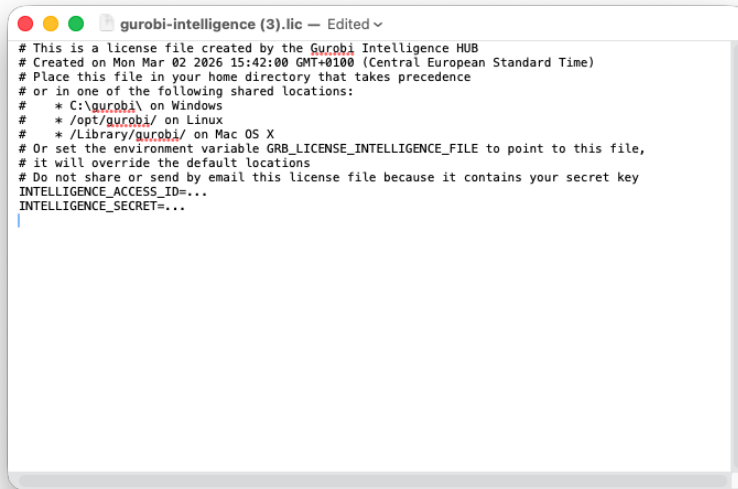
Important

For security reasons, the key value and the license file can only be downloaded at creation time. Store them securely. You can create new API keys at any time if needed.

6.2 License File

The downloaded license file (`gurobi-intelligence.lic`) contains two parameters needed to authenticate with the Intelligence Hub backend:

```
INTELLIGENCE_ACCESS_ID=...
INTELLIGENCE_SECRET=...
```



Place the license file in your home directory, or in one of the following shared locations:

- C:\gurobi\ on Windows
- /opt/gurobi/ on Linux
- /Library/gurobi/ on macOS

Alternatively, set the environment variable `GRB_LICENSE_INTELLIGENCE_FILE` to point to the license file. This overrides the default locations.

Warning

Do not share or send the license file by email. It contains your secret key.

6.3 Viewing API Keys

The API Keys page lists all keys associated with your account. Each key card displays:

- **Key ID:** The unique identifier for the API key.
- **Application name:** The name you assigned when creating the key.
- **Owner:** The account email associated with the key.
- **Created at:** The date and time the key was created.
- **Last used:** The last time the key was used (N/A if never used).

- **With IP:** The IP address from which the key was last used.
- **In region:** The region from which the key was last used.

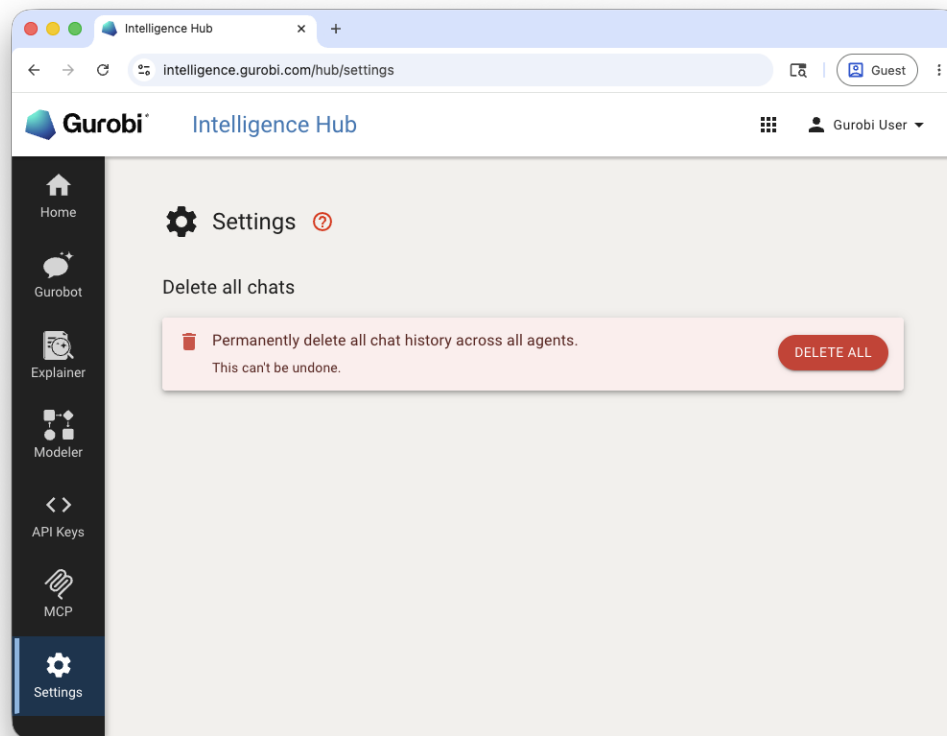
You can filter keys using the filter dropdown, search by name, and toggle between grid and list views using the view controls in the toolbar.

6.4 Managing API Keys

Each key card provides an edit button to modify the application name or description, and a menu with additional options. Use these controls to keep your keys organized and to revoke any that are no longer needed.

SETTINGS

The Settings page provides account-wide controls for managing your Intelligence Hub data. Access it by clicking **Settings** in the left sidebar. Today the page exposes a single action—deleting all chats—and additional settings may appear here in the future.



7.1 Delete All Chats

All interactions with the Intelligence Hub—including inputs, outputs, and uploaded files—are stored by the service. The Settings page allows you to permanently delete all chat history across all agents (Gurobot, Explainer, and Modeler) at once.

To delete all chats:

1. Click **Settings** in the left sidebar.
2. Click the **DELETE ALL** button.

This permanently removes every conversation across all agents from your account and from our records. Only basic usage metadata is retained for reporting purposes.

To delete the chat history of a single agent, open that agent and click **Clear all history** in its sidebar panel. You can also delete an individual conversation from the ... menu next to it in the sidebar.

Warning

Deleting all chats is permanent and cannot be undone. All conversation history, analysis results, and uploaded files across all agents will be permanently deleted from our systems.

FREQUENTLY ASKED QUESTIONS

This section answers common questions about getting started with the Intelligence Hub and about how your data is handled. For the authoritative statement on data handling, always refer to the *Terms of Service* you accept before using the agents.

8.1 Getting Started

What is the Intelligence Hub, and which agent should I use?

The Intelligence Hub is a web application with three specialized agents. Use *Gurobot* for expert Gurobi support—questions, code snippets, and documentation. Use *Explainer* to diagnose infeasibility, repair feasibility, and run sensitivity analysis on a model you upload. Use *Modeler* to turn a plain-language business problem into a complete `gurobipy` project with a specification, implementation, and tests. See the *Overview* for how the agents compare and what stage each is at.

How do I access it?

The Intelligence Hub is available from the [Gurobi portal](#) and directly at intelligence.gurobi.com. The first time you sign in, you must review and accept the *Terms of Service* before you can use the agents.

Do I need to install anything?

No—the agents run on the Gurobi Intelligence Hub backend and you access them through your browser, with nothing to install. If you want to use them from an IDE or AI coding assistant instead, you can run the *MCP server* locally, which requires Docker.

How does the agent know about the latest Gurobi behavior?

All agents are grounded in a Gurobi-curated *knowledge base* that is re-indexed daily and cite their sources, so answers reflect current Gurobi guidance rather than generic web content. When writing `gurobipy` code, they also consult a dedicated *coding guide*.

What files can I work with, and where do they go?

Each conversation has its own *workspace* where uploaded and generated files are stored. Agents can also run code in a sandboxed *code execution* environment to verify what they produce. See the individual agent pages for the file types each one accepts and generates.

Is there a usage limit?

Yes. Daily usage limits apply across all agents, and a *daily usage indicator* on the home page shows how much of your limit you have consumed. Access may be temporarily suspended if you exceed the limit, and usage resets daily.

How do I give feedback?

Every agent response has thumbs up / thumbs down controls. See *Feedback* for how that feedback is used.

8.2 Security and Privacy

The answers below summarize how the Intelligence Hub handles your data. The binding terms are in the [Terms of Service](#).

Are my inputs or outputs—uploaded files, prompts, responses, or chat history—used to train, retrain, or fine-tune any LLM?

No. Your interactions, inputs, outputs, and model input files are never used to train, fine-tune, or otherwise improve any large language model. See the **Data is Not Used for Training LLMs** clause of the [Terms of Service](#).

Are my inputs or outputs used for analytics or product improvement?

Yes, in a limited way. Gurobi reserves the right to process your interactions to validate, upgrade, maintain, and otherwise develop the service—this does not include training LLMs (see above). In practice, the main driver of product improvement is your *feedback*, which Gurobi's optimization experts use to refine the knowledge base and the agents' instructions. See the **Service Improvement** clause of the [Terms of Service](#).

Is my data isolated to my account, or is any of it processed in a shared environment that other organizations could see?

Your inputs are processed in a dedicated environment with no shared state or filesystem with any other session or user. Only the portions of your input needed to fulfill a request are transmitted to a large language model running within Gurobi's secure cloud infrastructure. See the **Security** points in the [Terms of Service](#) for the full description of isolated processing and limited exposure.

How is my data protected in transit and at rest?

Inputs are encrypted in transit with HTTPS and encrypted at rest using application-layer encryption with a unique key per file. See the **Security** points in the [Terms of Service](#).

Can I delete my data?

Yes. You can delete an individual conversation, all chats for an agent, or all chats across every agent at any time from the [Settings](#) page. Your inputs are permanently deleted when you do so, when you delete your account, or when you send a written deletion request to operations@gurobi.com, subject to applicable legal retention requirements. See the **Deletion** point in the [Terms of Service](#).

Should I include personal or confidential data in my prompts?

No. Do not include data that identifies or could reasonably be linked to a natural person—names, addresses, social security numbers, payment card information, and the like. See the **Personal data** point in the [Terms of Service](#).

How is the MCP server kept secure?

The local [MCP server](#) is sandboxed to the single root directory you mount, passes your credentials only as environment variables rather than storing them in a project config, and exposes its activity through standard Docker logs. See the **Design Rationale** in the [MCP server](#) page.

Is the agent's output guaranteed to be correct?

No. The service uses generative AI and may produce inaccuracies. Independently verify any output before relying on it. See the **Generative AI disclaimer** in the [Terms of Service](#).

TERMS OF SERVICE

Last updated: June 10, 2026

9.1 Introduction

Welcome to the artificial intelligence service provided by Gurobi Optimization, LLC (“we,” “our,” or “us”) for use in connection with the Gurobi software product and mathematical optimization (“Service”). These Terms of Service (“Terms”) govern your, including your authorized users (collectively referred to herein as “you” and “your”), access to and use of the Service. These Terms do not and are not intended to replace or supersede the agreement between you and us governing your access to and use of our software product (“EULA”). Notwithstanding the foregoing, to the extent these Terms directly conflict with the EULA, and such conflicting provisions relate solely to your access to and the use of the Service, these Terms will govern and control solely with respect to the Service.

9.2 Acceptance of Terms

By using the Service, you hereby acknowledge that you have read, understood, and agree to be bound by these Terms and our Privacy Policy. If you do not agree to be bound by these Terms or our Privacy Policy, you may not use the Service.

9.3 Scope of Service

Focus of Service: The Service is designed solely for discussions and inquiries between you and the Service through text or chat features related to mathematical optimization and the usage of the Gurobi products (“Interactions”). Any attempt to use the Service for other purposes is strictly prohibited. Subject to these Terms, you may input, provide, upload, or otherwise disclose information, data, content, or other materials into the Service (collectively “Inputs”) including any MPS, LP, API recording or other model input files used as input to the Service (collectively, “Model Input Files”). The Service may generate information, data, content, or other materials in connection with your Interactions with the Service (collectively “Outputs”).

Usage Limits: Daily usage limits may apply to your Interactions. We reserve the right to temporarily suspend your access to and use of the Service if you exceed those usage limits, without prejudice to any other rights or remedies we may have under these Terms.

9.4 User Responsibilities

Accuracy of Information: You are responsible for ensuring that all Inputs are complete, accurate, and do not violate any applicable law, regulation, or third-party right, or these Terms. You represent and warrant that you have all requisite rights, licenses, and authority to input, provide, upload, or otherwise disclose such Inputs in connection with your use of the Service.

9.4.1 Prohibited Activities

You will not, and will not permit any third party to, directly or indirectly:

- Exceed the daily usage limits as set forth in these Terms;
- Use the Service for any purpose other than in connection with the Gurobi product for purposes of mathematical optimization;
- Input, provide, upload, or otherwise disclose any data or information that identifies or could reasonably be linked with a natural person or any other sensitive data (including, without limitation, names, addresses, social security numbers, payment card information, etc.) (collectively, “Personal Data”). In the event that you choose to provide Personal Data in your Interactions or Inputs or otherwise through your use of the Service, such Personal Data shall be processed as set forth in “Data Collection and Privacy” below;
- Use foul, abusive, harassing, or otherwise harmful language, or engage in any behavior that is disruptive, abusive, or otherwise harmful to the Service or other users;
- Input, provide, upload, or otherwise disclose or attempt to execute or run on or through the Service any worm, virus, trojan horse, spyware, spam, or other malicious or harmful code; or
- Engage in any activity that could or is intended to harm, disrupt, or degrade the Service, any other users use of the Service, or our network, servers, infrastructure, or data.
- Attempt to reverse engineer, extract, discover, or otherwise derive the underlying models, prompts, algorithms, or security mechanisms used by the Service.

9.5 Data Collection and Privacy

Data Storage: We store all Interactions with the Service, including Inputs, Outputs, including, without limitation, any files generated during any Interaction.

User Control: Subject to our obligation to comply with applicable law, you have the ability to delete individual Interactions or all Interactions from your account. Should you decide to delete your account, all associated Interactions will be permanently deleted.

Privacy Policy: The Gurobi Privacy Policy applies to all data collected and processed in connection with the Services. By using this Service, you consent to the collection, storage, and use of your data (including any Personal Data you choose to provide) as described in our Privacy Policy.

Data is Not Used for Training LLMs: Your Interactions, Inputs, Outputs, and Model Input Files will not be used to train, fine-tune, or otherwise improve any large language models (often referred to as “LLMs”).

Service Improvement: Subject to your right to delete your Interactions as set forth above, you hereby grant us a right to copy, reproduce, display, modify, store, use, and otherwise process your Interactions (including any Personal Data you choose to provide) with the Service in order to validate, upgrade, update, improve, maintain, modify, and otherwise develop the Service (collectively “Improvements”). We may also use your questions, comments, suggestions, or other feedback to make Improvements, and you hereby permit us to do so without attribution, restriction, or compensation.

You acknowledge and agree that we have and retain all right, title, and interest in and to the Service, including any and all Improvements thereto and any and all intellectual property rights therein.

9.6 Data Processing and Security

9.6.1 Service Delivery

In order to operate and deliver the Service, we must and you hereby authorize us to store, copy, transmit, display, and process your Inputs, including any Model Input Files, solely as necessary to respond to and fulfill your requested Interactions. This includes transferring such Inputs to isolated compute environments for processing. This grant is limited to what is strictly necessary for service delivery and the improvement of the Service.

9.6.2 Security Measures

We take the security of your Inputs, including any Model Input Files, seriously and apply the following measures:

- **Encryption in Transit.** All Inputs transmitted between your device and our Service are encrypted using industry-standard TLS over HTTPS.
- **Encryption at Rest.** All Inputs are encrypted at rest using application-layer encryption, where each file is encrypted with a unique key.
- **Isolated Processing.** While being processed, your Inputs are handled within a dedicated, isolated environment with no shared state or shared filesystem with any other session or user.
- **Limited Exposure.** Portions of your Inputs may be transmitted to a large language model operating within our secure cloud infrastructure solely to fulfill your requested Interaction.

9.6.3 Model Input Files and Support

Model Input Files submitted as Inputs to the Service are processed solely to deliver your requested Interaction and are not placed in Gurobi's model library or used for solver benchmarking. However, if you choose to escalate an Interaction to a Gurobi support ticket, any Model Input Files associated with that Interaction may be transferred to Gurobi's support team and, from that point, shall be governed by Section 7.2 of the EULA, including its provisions regarding the model library, benchmarking, and deletion upon written request.

9.6.4 Deletion

Your Inputs, including any Model Input Files, will be permanently deleted upon the earliest of: (a) your deletion of the related chat session; (b) your deletion of all chats associated with an agent; (c) your deletion of all chats; (d) your deletion of your account; or (e) receipt of your written deletion request submitted to operations@gurobi.com, subject to applicable legal retention requirements.

9.7 Termination of Access

The Service is provided for convenience only at our sole discretion. We reserve the right to modify, suspend or terminate your access to, or otherwise cease offering, the Service at any point for no reason or any lawful reason, including, without limitation, for any violation of these Terms or any other misuse of the Service.

9.8 Generative AI Disclaimer and Accuracy Notice

THE SERVICE IS PROVIDED “AS IS”, “AS AVAILABLE”, AND WITHOUT ANY WARRANTIES OR REPRESENTATIONS OF ANY KIND. TO THE GREATEST EXTENT PERMITTED BY LAW, WE HEREBY DISCLAIM ANY AND ALL WARRANTIES AND REPRESENTATIONS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, AND ANY WARRANTIES ARISING OUT OF ANY COURSE OF DEALING OR USAGE OF TRADE.

WITHOUT LIMITING THE GENERALITY OF THE FOREGOING, WE DO NOT WARRANT THAT THE SERVICE WILL OPERATE WITHOUT INTERRUPTION, BE ERROR FREE OR GENERALLY AVAILABLE, OR THAT ANY OUTPUTS BY THE SERVICE ARE SUITABLE, ACCURATE, ERROR FREE, OR CAN OR SHOULD BE RELIED ON FOR ANY PURPOSE WHATSOEVER.

You are hereby advised to independently verify any and every Output provided by the Service prior to relying on, or permitting or inducing any third party to rely upon, any such Output. You are solely responsible for assessing the suitability, accuracy, reliability, propriety, and security of any Output, and for ensuring that your use of and reliance on any Output complies with all applicable laws, regulations, and best practices.

9.9 Limitation of Liability and Indemnification

TO THE GREATEST EXTENT PERMITTED BY LAW, IN NO EVENT SHALL WE, INCLUDING OUR AFFILIATES AND OUR AND THEIR OFFICERS, DIRECTORS, SHAREHOLDERS, EMPLOYEES, AGENTS, AND SUCCESSORS AND ASSIGNS, BE LIABLE FOR ANY INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, PUNITIVE DAMAGES, OR FOR ANY LOSS OF PROFITS, GOODWILL, LOSS OF DATA, OR OTHER LOSSES, ARISING OUT OF OR RELATED TO THESE TERMS AND YOUR USE OF THE SERVICE, REGARDLESS OF WHETHER WE WERE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND REGARDLESS OF THE THEORY OF LIABILITY, WHETHER ARISING IN CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE. TO THE GREATEST EXTENT PERMITTED BY LAW, IN NO EVENT SHALL WE, INCLUDING OUR AFFILIATES AND OUR AND THEIR OFFICERS, DIRECTORS, SHAREHOLDERS, EMPLOYEES, AGENTS, AND SUCCESSORS AND ASSIGNS, TOTAL AGGREGATE LIABILITY ARISING OUT OF OR RELATED TO THESE TERMS AND YOUR USE OF THE SERVICE EXCEED THE GREATER OF (I) FEES PAID BY YOU IN THE SIX MONTHS IMMEDIATELY PRECEDING THE FIRST EVENT GIVING RISE TO THE FIRST LIABILITY, AND (II) TWO HUNDRED DOLLARS (\$200). THE FOREGOING IS AN AGGREGATE LIABILITY CAP, NOT A PER CLAIM OR PER INCIDENT LIABILITY CAP.

To the greatest extent permitted by law, you will defend, indemnify, and hold us, including our affiliates and our and their officers, directors, shareholders, employees, agents, and successors and assigns, harmless from and against any and all third party allegations, claims, demands, lawsuits, proceedings, investigations, and any and all damages, liabilities, judgments, fines, penalties, and costs and expenses (including reasonable attorney’s fees) in connection therewith, that arise from or relate to: (i) your infringement upon or misappropriation of any third-party intellectual property right; (ii) your violation of any third-party right of privacy; (iii) your violation of applicable law; and (iv) your use of the Service other than as expressly permitted by these Terms.

9.10 Modifications to the Terms

We reserve the right to amend these Terms at any time by posting the amended terms online. It is your responsibility to regularly review the Terms for any amendments. Any amended Terms will be deemed accepted and become effective seven (7) days from when such Terms are posted online (the “Notice Period”), and your continued use of the Service after the Notice Period constitutes your acceptance of such amendment. If you do not agree to the amended Terms, you must cease use of and access to the Service prior to the conclusion of the Notice Period.

9.11 Contact Information

For any questions about these Terms, please contact us at:

Email: operations@gurobi.com

Address: 9450 SW Gemini Dr #90729, Beaverton, OR 97008