



# DB Gene 4.4.0 Release Notes

December 11th, 2024

Copyright © 2012-2024 DecisionBrain S.A.S. All rights reserved.

All specifications and information regarding the products in this document are subject to change without notice and should not be construed as a commitment by DecisionBrain. DecisionBrain assumes no responsibility or liability for any mistakes or inaccuracies that may appear in this document. All statements and recommendations in this document are believed to be accurate but are presented without warranty. Users must take full responsibility for their application of any product.

# DB Gene 4.4.0 Release Notes

<b>Important Notes</b>	<b>3</b>
Updates	3
Deprecations	4
<b>End-User Features</b>	<b>5</b>
New Gene Online Beta Feature	5
New Documentation Chatbot Beta Feature	6
New ChatGPT Feature in the Code Editor Widget	8
New Status Bar API	9
New Jupyter Notebook Sample	9
New Gantt Chart Widget Features	10
Improved Data Grid/Explorer Widgets	10
Improved Filter Bar and Filter Widget	10
<b>Technical Features</b>	<b>11</b>
Improved Jackson JSON Serialization Capabilities	11
Improved Security for Trivy CVEs	12
Improved Java Memory Usage Limit	12
<b>Changelog</b>	<b>13</b>
Improvements	13
Bugfixes	16

## **Note:**

DB Gene 4.4.0 introduces several infrastructure updates and deprecations. They are described in DB Gene 4.4.0 Migration Guide, available on the [DecisionBrain website](#).

The following information only focuses on the main changes in this release.

---

# Important Notes

---

*Please take into account the following important information when using the new version of the **DB Gene 4.4.0**, released on December 11th, 2024.*

---

## Updates

DB Gene 4.4.0 introduces the following infrastructure updates.

### Spring Update

DB Gene 4.4.0 now uses SpringBoot 3.3.5 and Spring Cloud 2023.0.3, Formerly, it was version 3.2.4 and version 2023.0.1, respectively.

### Angular Update

DB Gene 4.4.0 now uses Angular 18.2.10. Formerly, it was version 18.2.2.

### AG Grid Update

DB Gene 4.4.0 now uses AG Grid 32.3.1. Formerly, it was version 31.2.0.

### Keycloak Update

DB Gene 4.4.0 now uses Keycloak 26.0.5. Formerly, it was version 24.0.2.

### RabbitMQ Update

DB Gene 4.4.0 now uses RabbitMQ 4.0.2. Formerly, it was version 3.13.0.

### CPLEX Update

DB Gene 4.4.0 now uses CPLEX 22.1.2. Formerly, it was version 22.1.1.

### Python Update

DB Gene 4.4.0 now requires Python 3.12.x. Formerly, it was version 3.11.x.

## Deprecations

- *Pivot Table (Experimental)* is now renamed to *Pivot Table*. *Pivot Table - Legacy* is no longer available in the widget dropdown list.
- In the Filter widget controller, the method `GeneEntityFilter.fromSelection` is deprecated in favor of `GeneEntityFilter.fromConfig`. To know if a `GeneEntityFilter.fromConfig` corresponds to a `GeneWidgetFilter`, you can use `getFromConfigComparator` from `@gene/components`.
- In the Filter widget controller, for the `buildFilters` function in `@gene/components`, the parameter `contextSelectionKey: string` has been replaced by `filterIdentifier: GeneFilterIdentifier`. A `GeneFilterIdentifier` consists of a `typeName` and the paths leading to that `typeName`.
- In the Gantt Chart widget controller, the methods `getResourceQueryName()` and `getEventsQueryName()` in `DbGanttBuilderMethods` are not used by the `DbGanttBuilder`. They are now deprecated and will be removed in a later version.
- In the Gantt Chart widget controller, the methods `loadResources` and `loadEvents` are not used in `GanttController`. They are now deprecated and will be removed in a later version.
- In the Scenario Service GraphQL API, the property `scenarioReferenceGraph` has been removed from the definition of object Path. The REST API is still exposing this property.

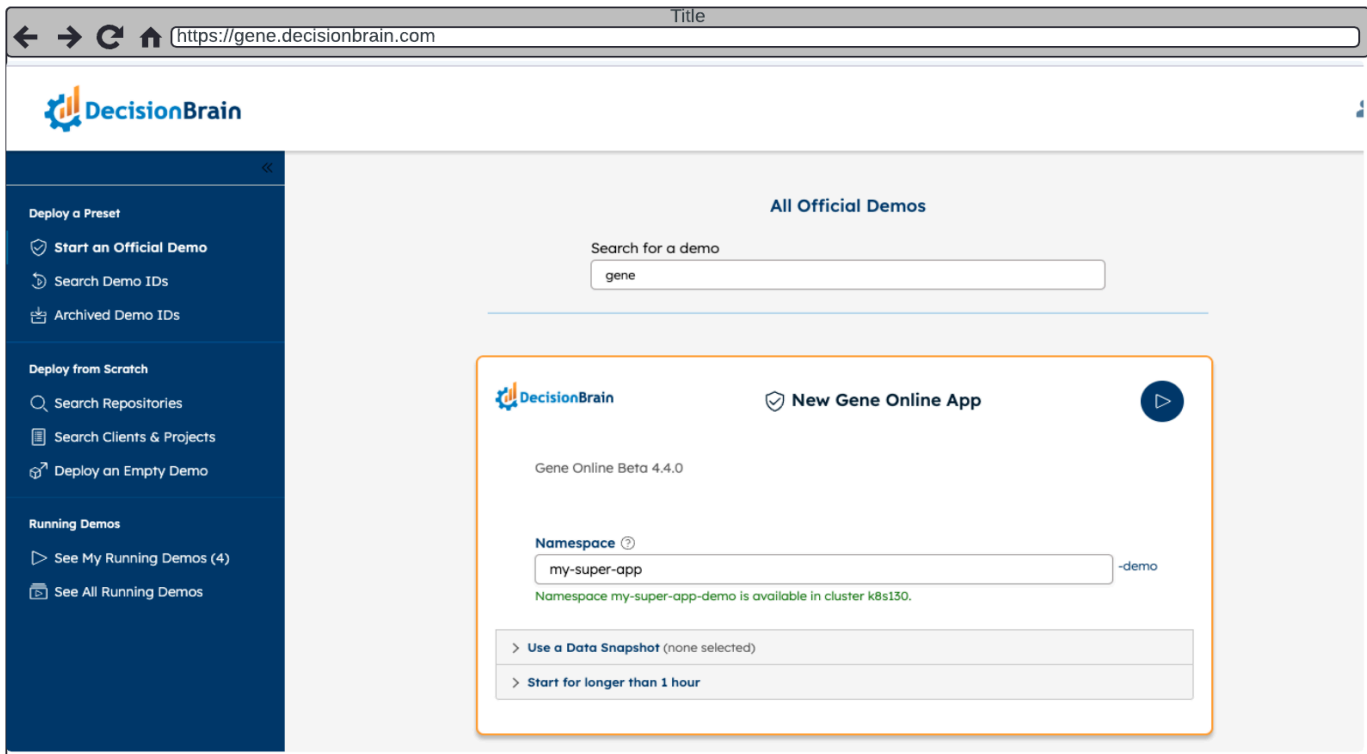
# End-User Features

DB Gene 4.4.0 introduces the new Gene Online Beta and Documentation Chatbot features, a new Jupyter Notebook sample, a new ChatGPT feature in the Code Editor widget, the new Status Bar API, new features for the Gantt Chart widget, and improved Data Grid/Explorer widgets.

## New Gene Online Beta Feature

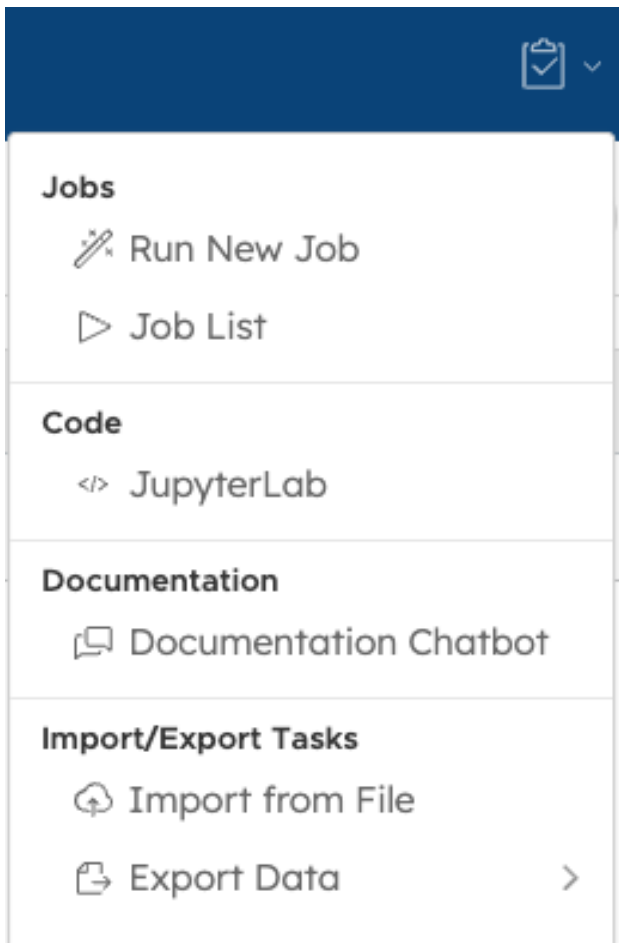
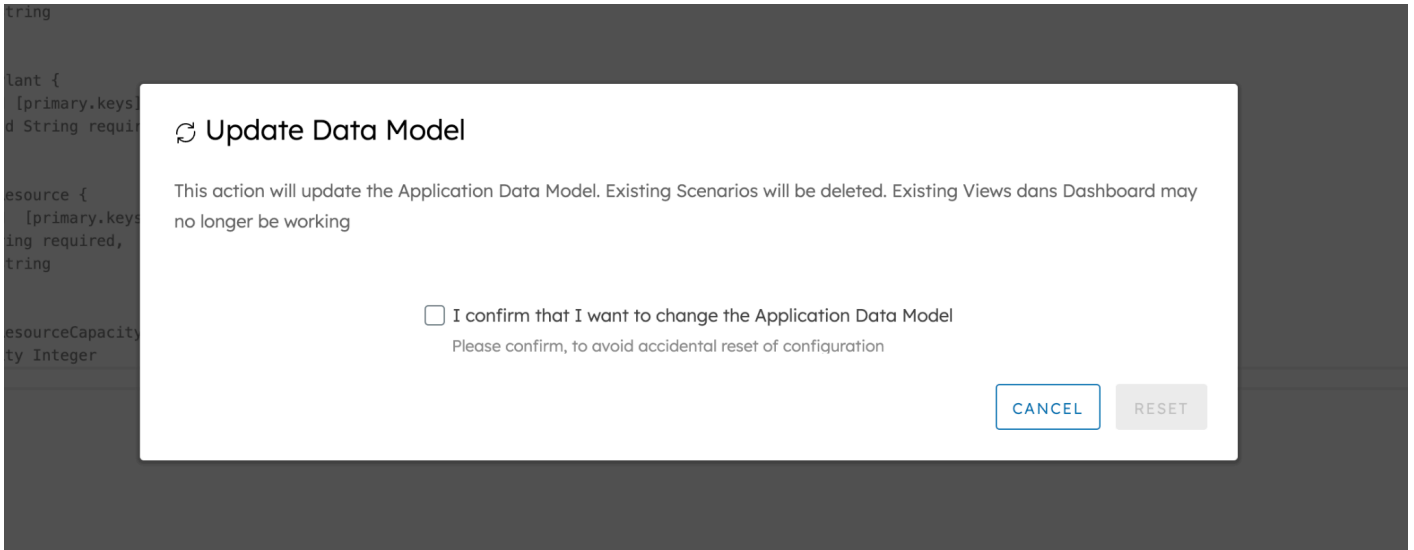
Gene Online Beta 4.4.0 is now available. It allows developers to deliver DB Gene-based applications faster, through a development cycle relying on the low-code UI and out-of-the-box configurations compared to a local installation.

There is no need for a development environment as an empty customizable application can be launched directly from the web.



The screenshot shows a web browser window with the URL `https://gene.decisionbrain.com`. The page title is "Title". The DecisionBrain logo is in the top left. A dark blue sidebar on the left contains navigation options: "Deploy a Preset" (with sub-items "Start an Official Demo", "Search Demo IDs", "Archived Demo IDs"), "Deploy from Scratch" (with sub-items "Search Repositories", "Search Clients & Projects", "Deploy an Empty Demo"), and "Running Demos" (with sub-items "See My Running Demos (4)", "See All Running Demos"). The main content area is titled "All Official Demos" and features a search bar with the text "gene". Below the search bar, a demo card is highlighted with an orange border. The card is titled "New Gene Online App" and includes the DecisionBrain logo, a play button icon, and the text "Gene Online Beta 4.4.0". It also shows a "Namespace" field with the value "my-super-app" and a "-demo" suffix. A note below the namespace field states: "Namespace my-super-app-demo is available in cluster k8s130." At the bottom of the card, there are two expandable sections: "Use a Data Snapshot (none selected)" and "Start for longer than 1 hour".

Once the web client starts, users can quickly set the application data model through the JDL Editor, configure the application as usual, and use the JupyterLab integration to edit and run optimization/ML code from an online editor.



## New Documentation Chatbot Beta Feature

The Documentation Chatbot now allows asking questions on the application features based on the Jira tickets, source code, and selected specification documents of the project.

Usage instructions are in the file `EXPERIMENTAL_FEATURES.md` generated for applications with the experimental features enabled.

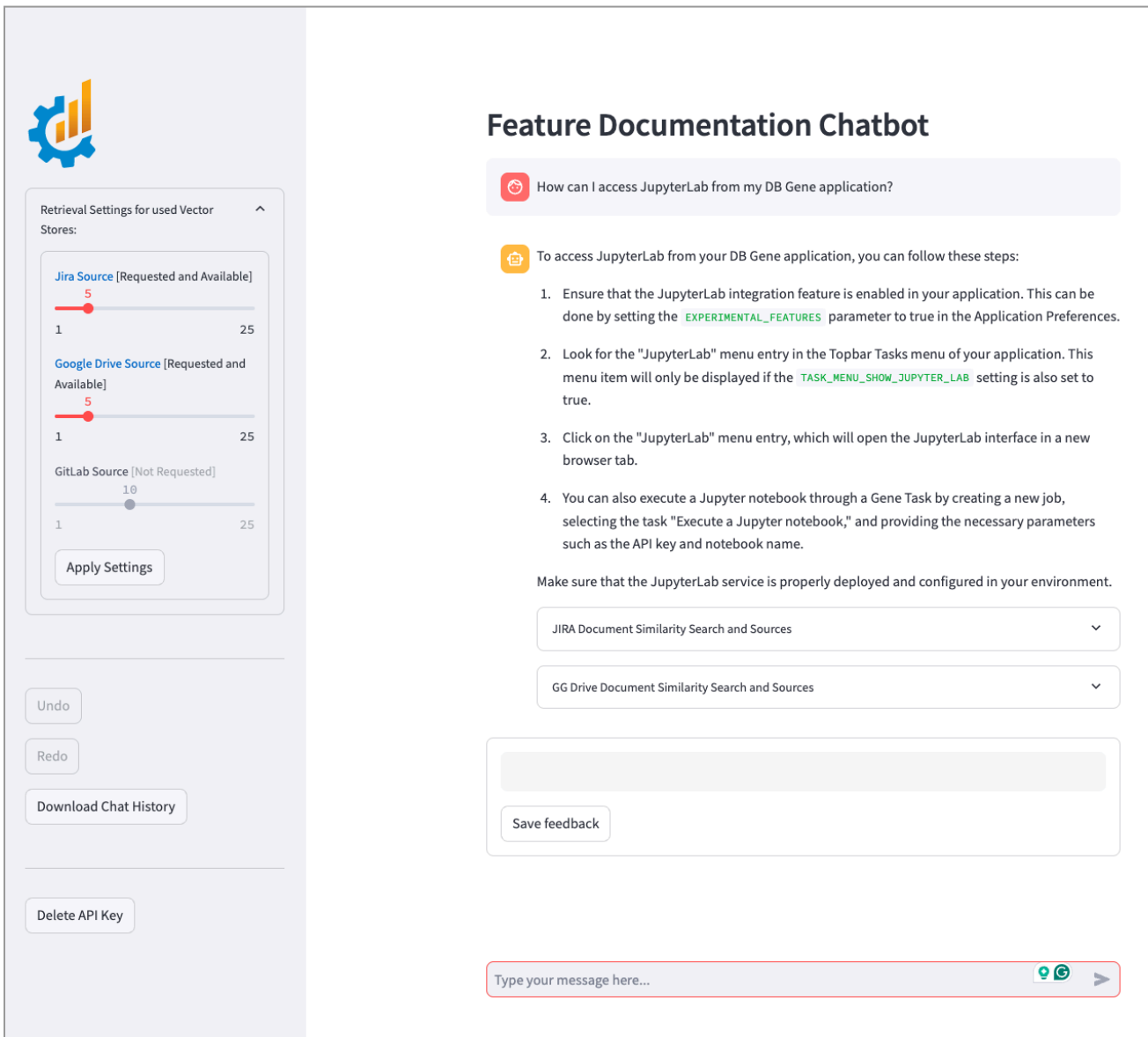
To activate this chatbot, follow the instructions in `documentation-chatbot/build.gradle`. Then, the build chain will scan the data sources (Jira, Gitlab, and/or Google Drive) to create vector stores and inject them into the Docker image for the chatbot.

From the application web client, users can open the chatbot via a command in the Task menu.

To display this menu, in the Application Preferences, the parameters `EXPERIMENTAL_FEATURES` and `TASK_MENU_SHOW_DOCUMENTATION_CHATBOT` must both be set to true.

Also, an OpenAI API key is required. It can be set as a Spring property in the `application.yml` file of the Backend Service extension, or provided at deployment time in the `app/.env` file for Docker Compose deployment, or in the values file of the deployment configuration when deploying with Kubernetes.

If no key is provided, it can be indicated using the Documentation Chatbot from the web client.



The screenshot displays the 'Feature Documentation Chatbot' interface. On the left, a sidebar contains a settings panel for 'Retrieval Settings for used Vector Stores' with sliders for 'Jira Source', 'Google Drive Source', and 'GitLab Source'. Below this are buttons for 'Undo', 'Redo', 'Download Chat History', and 'Delete API Key'. The main chat area has a title 'Feature Documentation Chatbot' and a question: 'How can I access JupyterLab from my DB Gene application?'. The response provides a list of four steps for accessing JupyterLab. Below the steps, there are two dropdown menus for 'JIRA Document Similarity Search and Sources' and 'GG Drive Document Similarity Search and Sources', a 'Save feedback' button, and a text input field with a placeholder 'Type your message here...' and a send button.

## Feature Documentation Chatbot

How can I access JupyterLab from my DB Gene application?

To access JupyterLab from your DB Gene application, you can follow these steps:

1. Ensure that the JupyterLab integration feature is enabled in your application. This can be done by setting the `EXPERIMENTAL_FEATURES` parameter to true in the Application Preferences.
2. Look for the "JupyterLab" menu entry in the Topbar Tasks menu of your application. This menu item will only be displayed if the `TASK_MENU_SHOW_JUPYTER_LAB` setting is also set to true.
3. Click on the "JupyterLab" menu entry, which will open the JupyterLab interface in a new browser tab.
4. You can also execute a Jupyter notebook through a Gene Task by creating a new job, selecting the task "Execute a Jupyter notebook," and providing the necessary parameters such as the API key and notebook name.

Make sure that the JupyterLab service is properly deployed and configured in your environment.

JIRA Document Similarity Search and Sources

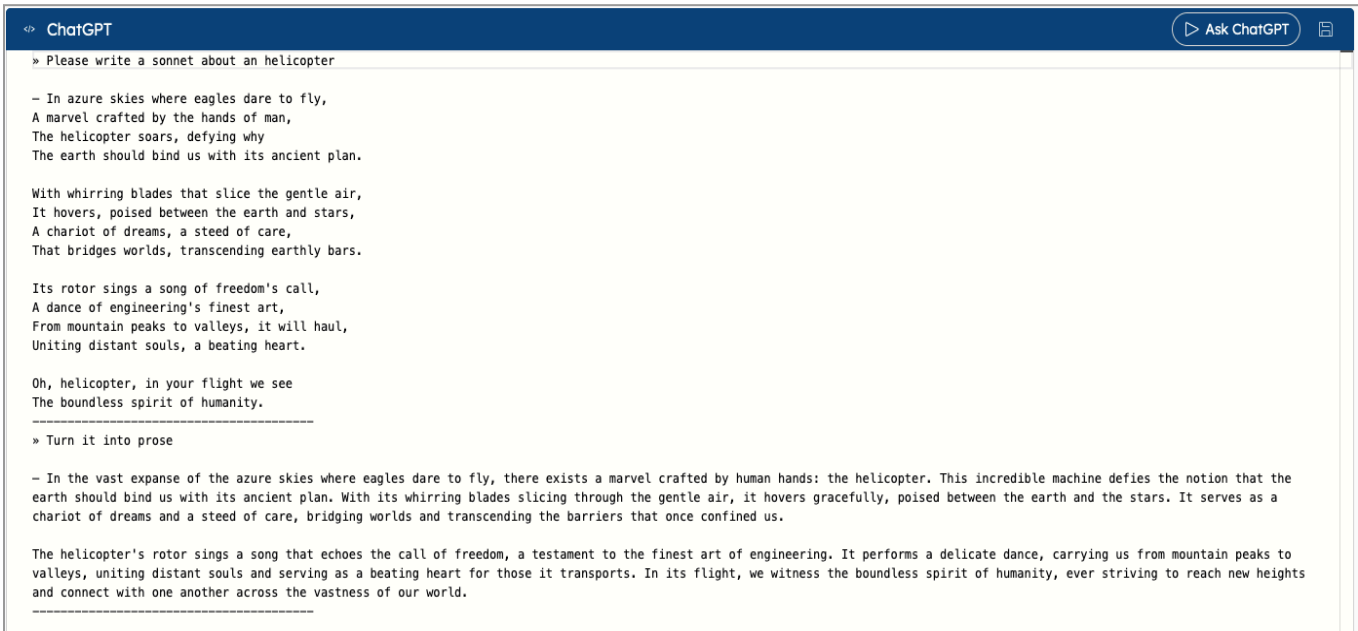
GG Drive Document Similarity Search and Sources

Save feedback

Type your message here...

## New ChatGPT Feature in the Code Editor Widget

A scripted task now allows using the Code Editor widget as a ChatGPT discussion terminal.



```

<> ChatGPT
▶ Ask ChatGPT
» Please write a sonnet about an helicopter

- In azure skies where eagles dare to fly,
A marvel crafted by the hands of man,
The helicopter soars, defying why
The earth should bind us with its ancient plan.

With whirring blades that slice the gentle air,
It hovers, poised between the earth and stars,
A chariot of dreams, a steed of care,
That bridges worlds, transcending earthly bars.

Its rotor sings a song of freedom's call,
A dance of engineering's finest art,
From mountain peaks to valleys, it will haul,
Uniting distant souls, a beating heart.

Oh, helicopter, in your flight we see
The boundless spirit of humanity.
-----
» Turn it into prose

- In the vast expanse of the azure skies where eagles dare to fly, there exists a marvel crafted by human hands: the helicopter. This incredible machine defies the notion that the earth should bind us with its ancient plan. With its whirring blades slicing through the gentle air, it hovers gracefully, poised between the earth and the stars. It serves as a chariot of dreams and a steed of care, bridging worlds and transcending the barriers that once confined us.

The helicopter's rotor sings a song that echoes the call of freedom, a testament to the finest art of engineering. It performs a delicate dance, carrying us from mountain peaks to valleys, uniting distant souls and serving as a beating heart for those it transports. In its flight, we witness the boundless spirit of humanity, ever striving to reach new heights and connect with one another across the vastness of our world.
-----

```

This experimental feature provides access to OpenAI's GPT models. Its simplest usage involves equipping a Code Editor widget with a button that launches the `ProcessChatGptConversationTask`. This scripted task requires the following inputs:

- The current scenario;
- The name of the parameter associated with the Code Editor widget;
- An optional API key for OpenAI (useful if none is provided in the Spring properties);
- An optional model name (to override the default);
- An optional model temperature (to override the default).

In this setup, the content of the Code Editor widget serves as the input question, which is passed to the GPT model. The model's response is appended to the content of the Code Editor. Markers (» and -) are used to differentiate the question and the answer, with question-answer pairs separated by lines of dashes. Subsequent questions can be added below the last line of dashes.

The list of questions and answers above the most recent question constitutes the conversation history with the GPT model. This history can be referenced in subsequent questions. If maintaining the history is unnecessary, or if it is not needed, the content of the Code Editor widget can be cleared before entering a new question. A pre-prompt can also be provided as instructions for the GPT model. To achieve this, if



the parameter `conversation` is associated with the Code Editor containing the questions and answers, the pre-prompt should be stored in the parameter `conversation-instructions`.

The defaults for the LLM model and its temperature are configured as Spring properties in the `application.yml` file of the backend service extension. An OpenAI API key is required. This can be specified in one of the following ways:

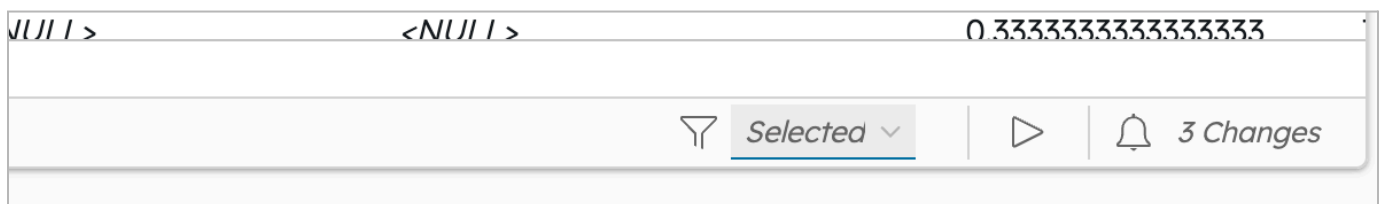
- As a Spring property in the `application.yml` file of the backend service extension (not the preferred option);
- Locally, when running from an IDE or through Docker Compose, as an environment variable;
- In the `deployment/docker/app/.env` file when deploying with Docker Compose;
- In the values file of the deployment configuration when deploying with Kubernetes.

The cost of each execution of the `ProcessChatGptConversationTask` is calculated and included as an output of the task execution. This cost is determined based on official pricing and is configured via Spring properties in the `application.yml` file of the backend service extension.

More advanced use cases may involve programmatically computing the content of the conversation and instruction parameters, instead of relying on the Code Editor widget.

## New Status Bar API

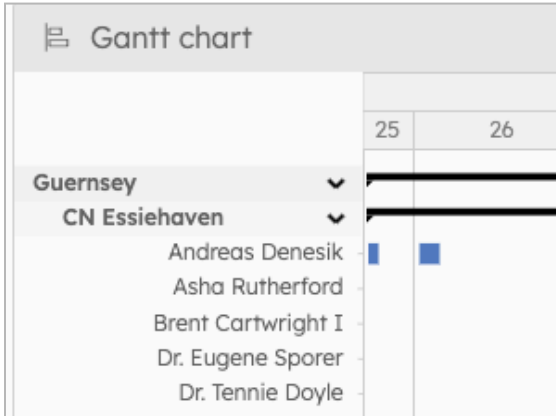
The Gene Status Bar API, which allows adding a status bar at the bottom of some widgets, is now available. The `SampleStatusBarController` sample is available by default.



## New Jupyter Notebook Sample

The application now provides users with a Jupyter Notebook sample to import scenario data into a Pandas data frame.





## New Gantt Chart Widget Features

The Gantt Chart widget now allows:

- Setting a custom context menu.
- Loading events dynamically.
- Grouping resources.
- Identifying resources and groups by `internalId`.

## Improved Data Grid/Explorer Widgets

In the Data Grid/Explorer widgets, `DateTime` columns can now be filtered only by `Date`.

Also, undefined Boolean values are now shown as `<NULL>`.

## Improved Filter Bar and Filter Widget

The Filter widget and the Filter Bar now allow filtering Booleans.

---

# Technical Features

---

*DB Gene 4.4.0 introduces several technical improvements regarding JSON object size limit, security, and memory usage.*

---

## Improved Jackson JSON Serialization Capabilities

The `jackson-core` library in version `>=2.15` introduced a configurable size limit for the serialized JSON object with a default value of `20MB`. This change in the `jackson-core` library used by DB Gene produced issues in some applications while transferring large scenarios between the Execution Service and Backend Service.

An application property now allows configuring the `jackson-core` size limit. This setting can be overridden and increased for both services when dealing with large scenarios.

Therefore, the value for both the Execution Service and Backend Service is now set to `250MB` by default.

The value can be edited but should be the same in both the following files:

- `extensions/execution-service-extension/src/main/resources/application.yml`
- `extensions/backend-service-extension/src/main/resources/application.yml`

Unset

```
# Configure the maximum size of a JSON payload. Used to send and receive
input/outputs to/from a routine.
gene:
  object-mapper:
    stream-max-string-length: 250000000 # 250MB
```

For more details, please refer to the [Jackson Core Github](#) page.

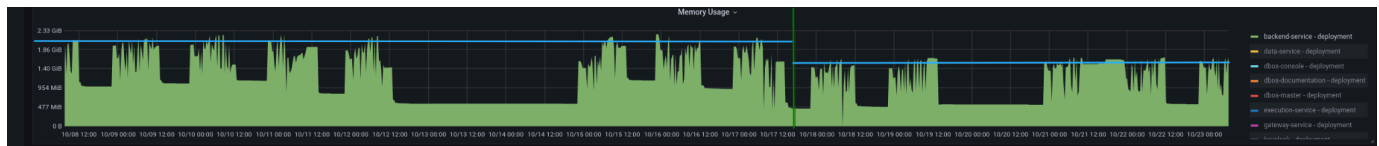
## Improved Security for Trivy CVEs

Trivy CVEs are now fixed using RabbitMQ 4.0.2.



## Improved Java Memory Usage Limit

Java native memory usage is now limited using the `MALLOC_ARENA_MAX` parameter, with negligible to no loss of performance.



# Changelog

DB Gene 4.4.0 introduces several improvements and bugfixes listed in detail below.

## Improvements

DB Gene 4.4.0 introduces the following improvements:

<b>Data Model</b>	DBPF-6544	Internal Types are now exported in Excel for Composite Data Model applications
<b>Data Scenario Service</b>	DBPF-7096	In the Scenario Service GraphQL API, the "scenarioReferenceGraph" property has now been removed from the "Path" object
<b>Dev 3rd-party Components</b>	DOC-842	Jackson "streamMaxStringLength" property is now configurable and set with 250MB default value
	DBPF-7349	platform-common-lib now relies on Spring Boot 3.3.5
	DBPF-7153	DB Gene now relies on Keycloak 25.0.5
	DOC-328	DB Gene now relies on AG Grid 32.3.1
	DBPF-7150	DB Gene now relies on Angular 18.2.10
	DBPF-6850	DB Gene is now compatible with CPLEX 22.1.2
	DBPF-7098	DB Gene now relies on Keycloak 26.0.5
	DBPF-7162	DB Gene is now compatible with Python 3.12.0
	DBPF-7099	DB Gene now relies on RabbitMQ 4.0.2
	DBPF-7097	DB Gene now relies on SpringBoot 3.3.5, Spring Cloud 2023.0.3, and Spring Framework 6.1.14

<b>Dev</b> Deployment	DOC-967	Java native memory usage is now limited using the MALLOC_ARENA_MAX parameter
<b>Dev</b> Documentation Chatbot	DBPF-7165	The Documentation Chatbot is now available
<b>Dev</b> Gene Online	DBPF-7101	Gene Online Beta 4.4.0 is now available
<b>Dev</b> JupyterLab	DBPF-7137	The application now provides users with a Jupyter Notebook example to import scenario data into Pandas Data Frames
<b>Dev</b> Security	DBPF-7183	Trivy CVEs are now fixed using RabbitMQ 4.0.2
<b>UI</b> Extensibility	DBPF-7254	The Gene Status Bar API, which allows adding a status bar at the bottom of some widgets, is now available
<b>UI</b> Code Editor	DBPF-7189	A scripted task now allows using the Code Editor widget as a ChatGPT discussion terminal
<b>UI</b> Data Grid/Explorer	DOC-938	In the Data Grid/Explorer widgets, "DateTime" columns can now be filtered only by "Date"
	DBPF-7266	In the Data Grid/Explorer widgets, undefined Boolean values are now shown as <NULL>
<b>UI</b> Filter	DOC-894	The Filter widget and the Filter Bar now allow filtering Booleans

<b>UI</b> Gantt	DOC-492	<i>The Gantt Chart widget now allows grouping resources</i>
	DBPF-7156	<i>The Gantt Chart widget now identifies resources and groups by "internalId"</i>
	DBPF-7234	<i>The Gantt Chart widget now supports context menu customization</i>
	DOC-606	<i>The Gantt Chart widget now supports dynamic loading of events</i>
<b>UI</b> Pivot Table	DBPF-7295	<i>Pivot Table (Experimental) is now renamed to "Pivot Table" and "Pivot Table - Legacy" is no longer available in the widget dropdown list</i>

## Bugfixes

DB Gene 4.4.0 introduces the following bugfixes:

<b>Data</b> Built-in Import/Export	DOC-996	<i>In some cases, the application was not importing DBRF/ZIP scenario files</i>
	DBPF-7324	<i>When a scenario was being imported in the background, using the option "Create a new scenario from the same file" was resetting the dialog</i>
	DBPF-7411	<i>When adding a scenario, the dialog was allowing to type in the "Scenario type" input field</i>
<b>Data</b> Database	DOC-947	<i>After a deadlock, the application was not properly retrying to fetch some data</i>
	DOC-944	<i>Cancelling a schema checker job was producing a large amount of logs</i>
	DOC-962	<i>During a transactional query, the application was not notifying changes if all creation, update and deletion parts were not present</i>
	DBPF-7274	<i>In some cases, the schema checker was triggering an NPE</i>
	DBPF-7390	<i>When saving a CSV scenario file into a Composite Data Model application, the file was incorrectly processed as a standard CSV file, resulting in the loss of relationships</i>
<b>Data</b> Scenario Service	DOC-961	<i>Scenario duplication was failing after resetting workspaces and folders</i>
	DBPF-7389	<i>The endpoint to retrieve a reference graph from the Scenario Service was not working properly</i>
	DBPF-7404	<i>The Execution Service was receiving 403 errors from the Scenario Service when creating scenario events due to missing job ID</i>



<b>DBOS</b> Master	DBPF-3102	<i>If RabbitMQ restarted during a DBOS job execution, the Execution Service was continuously stacking due to missing cleanup of anonymous queues</i>
	DBPF-6409	<i>Workers were failing to finish on Windows due to incorrect handling of file paths in URI resolution</i>
<b>Dev</b> 3rd-party Components	DBPF-7365	<i>Spring BOM was missing in "test-fixtures" dependencies, causing issues with version resolution</i>
	DBPF-7211	<i>The "npm-login.sh" script was not working if an old version of Node JS was installed</i>
	DBPF-7071	<i>The Keycloak login was not displaying properly due to a CSS error</i>
<b>Dev</b> REST API	DBPF-7094	<i>Some files were missing from "public-api" exports</i>
<b>Dev</b> Security	DOC-932	<i>Prometheus endpoint 'scenario_sessions_actives' was displaying inaccurate session counts when messages were lost</i>
<b>Scripted Tasks</b> Jobs	DBPF-7028	<i>The routine allowing to execute rules was not working properly</i>
<b>UI</b> Application Preferences	DBPF-7221	<i>The application was not taking some Application Preferences settings into account</i>
<b>UI</b> Code Editor	DBPF-7109	<i>When using the "Autosave" option, the Code Editor widget was not working properly</i>

<b>UI</b> Data Grid/Explorer	DBPF-1767	<i>In some cases in a Data Grid/Explorer widget based on a custom entity, the configurator was incorrectly displaying the "Is editable" checkbox</i>
	DBPF-7058	<i>Editing a Data Grid was triggering two refreshes of the dashboard</i>
	DOC-973	<i>The application was using incorrect color hints for nested fields</i>
	DBPF-7072	<i>Using a space character as a filter for text columns was causing the Data Grid/Explorer widget to crash</i>
	DOC-971	<i>When using a Data Grid in "Selection" mode, enabling scenario comparison was triggering a JS error</i>
<b>UI</b> Filter	DBPF-7041	<i>"Selection" mode was not properly applying to other dashboards and views despite the filter scope being set to "VIEW"</i>
	DOC-803	<i>Having multiple filters on the same Entity within a single Filter widget or Filter Bar was only using the value of the last one in the array</i>
	DBPF-7042	<i>In the Filter Bar, the drop-down list "Other filters" was not displaying "Selection" values properly</i>
	DBPF-6942	<i>In the Filter Bar, filter drop-down lists were not displaying values properly in the configurator preview</i>
	DBPF-7040	<i>When deleting the last filter of the Filter Bar, the configurator was still displaying it in the preview</i>

<b>UI</b> Gantt	DBPF-6911	<i>In some cases in a Gantt Chart widget with overlapping activities, the longer-lasting ones were hiding the shorter ones</i>
	DBPF-7080	<i>In some cases when using an existing Gantt Chart widget configuration, the application was displaying an empty chart</i>
	DBPF-7335	<i>In the Gantt Chart widget, the resource Selection option was available for a primitive grouping field</i>
	DOC-991	<i>In the Gantt Chart widget, the time axis was not aligning properly with the event timestamp</i>
	DBPF-7100	<i>The Gantt Chart widget was not properly loading filters when the scope was set to "VIEW"</i>
<b>UI</b> Job	DBPF-7086	<i>In the Job Details widget, the loading spinner was not displaying properly</i>
<b>UI</b> Pivot Table	DBPF-7012	<i>In the Pivot Table widget, the value "Aggregation" option was enabled but was not working properly</i>
	DBPF-7075	<i>In the Pivot Table widget, the value "Aggregation" option was not available for "internal_scenario_id"</i>
	DBPF-7013	<i>The Pivot Table widget configurator was allowing to save an incomplete configuration, causing the widget to not display properly</i>
	DOC-941	<i>The Pivot Table widget was not allowing AG Grid localization</i>