



# Securing extensions

Neo4j can be extended by writing custom code which can be invoked directly from Cypher, as described in [Java Reference → User-defined procedures](#) and [Java Reference → User-defined functions](#). This page describes how to ensure the security of these additions.

## Allow listing

Allow listing can be used to allow the loading of only a few extensions from a larger library. It is recommended to load extensions using the principle of least privilege. This principle dictates that you only load the procedures and functions necessary to execute your queries.

The configuration setting `dbms.security.procedures.allowlist` is used to name certain procedures and functions that should be available from a library. It defines a comma-separated list of procedures and functions that are to be loaded. The list may contain both fully qualified procedure names, and partial names with the wildcard `*`.

### *Example 1. Allow listing*

In this example, we need to allow the use of the method `apoc.load.json` as well as all the methods under `apoc.coll`. We do not want to make available any additional extensions from the `apoc` library, other than the ones matching these criteria.

```
# Example allow listing
dbms.security.procedures.allowlist=apoc.coll.*,apoc.load.json
```

There are a few things that should be noted about `dbms.security.procedures.allowlist`:

- If using this setting, no extensions other than those listed will be loaded. In particular, if it is set to the empty string, no extensions will be loaded.

[AI search](#)

**WARNING**

- The default of the setting is `*`. This means that if you do not explicitly give it a value (or no value), all libraries in the *plugins* directory will be loaded.

## Unrestricting

For security reasons, procedures and functions that use internal APIs are disabled by default. In this case, it is also recommended to use the principle of least privilege and only unrestrict those procedures and functions which you are certain to use.

Procedures and functions can be unrestricted using the configuration setting `dbms.security.procedures.unrestricted`. It defines a comma-separated list of procedures and functions that are to be unrestricted. The list may contain both fully qualified procedure and function names, and partial names with the wildcard (`*`) expression.

### *Example 2. Unrestricting*

In this example, we need to unrestrict the use of the procedures `apoc.cypher.runFirstColumn` and `apoc.cypher.doIt`.

```
# Example unrestricting
dbms.security.procedures.unrestricted=apoc.cypher.runFirstColumn,apoc.cypher.doIt
```

[Prev](#)[Security](#)[Next](#)[SSL framework](#)

## Contents







[Allow listing](#)[Unrestricting](#)

**Nov 6 2025**





The Call for Papers is now open and we want to hear about your graph-related projects. Submit your talks by June 15

[Submit your talk](#)

#### LEARN

-  [Sandbox](#)
-  [Neo4j Community](#)
- [Site](#)
-  [Neo4j Developer Blog](#)
-  [Neo4j Videos](#)
-  [GraphAcademy](#)
-  [Neo4j Labs](#)

#### SOCIAL

-  [Twitter](#)
-  [Meetups](#)
-  [Github](#)
-  [Stack Overflow](#)
- [Want to Speak?](#)

#### CONTACT US →

US: 1-855-636-4532  
Sweden +46 171 480 113  
UK: +44 20 3868 3223  
France: +33 (0) 1 88 46 13 20

© 2025 Neo4j, Inc.

[Terms](#) | [Privacy](#) | [Sitemap](#)

Neo4j®, Neo Technology®, Cypher®, Neo4j® Bloom™ and Neo4j® Aura™ are registered trademarks of Neo4j, Inc. All other marks are owned by their respective companies.