system.db.runPrepQuery

This function is used in **Python Scripting.**

Description

Runs a prepared statement against the database, returning the results in a PyDataSet.. Prepared statements differ from regular queries in that they can use a special placeholder, the question-mark character (?) in the query where any dynamic arguments would go, and then use an array of values to provide real information for those arguments. Make sure that the length of your argument array matches the number of question-mark placeholders in your query.

This call should be used for SELECT queries. This is a useful alternative to system.db.runQuery because it allows values in the WHERE clause, JOIN clause, and other clauses to be specified without having to turn those values into strings . This is safer because it protects against a problem known as a SQL injection attack, where a user can input data that affects the query's semantics.

> The "?" placeholder refers to variables of the query statement that help the statement return the correct information. The "?" placeholder cannot reference column names, table names, or the underlying syntax of the query. This is because the SQL standard for handling the "?" placeholder excludes these items.

Client Permission Restrictions

Permission Type: Legacy Database Access

Client access to this scripting function is blocked to users that do not meet the role/zone requirements for the above permission type. This function is unaffected when run in the Gateway scope.

Syntax

system.db. runPrepQuery(query, args, database, tx)

- Parameters

  String query - A query (typically a SELECT) to run as a prepared statement with placeholders (?) denoting where the arguments go.

  Object[] args - A list of arguments. Will be used in order to match each placeholder (?) found in the query.

  String database - The name of the database connection to execute against. If omitted or "", the project's default database connection will be used.

  String tx - A transaction identifier. If omitted, the query will be executed in its own transaction.

- Returns

  PyDataSet - The results of the query as a PyDataSet

- Scope

  All

Code Examples

Code Snippet

```
# This example would search for all records in a LogEntry table where the message contained
a user-entered search term.

search = event.source.parent.getComponent("SearchFor").text
# Wrap the term in % signs for LIKE-style matching
search = '%' + search + '%'

results= system.db.runPrepQuery("SELECT * FROM LogEntry WHERE EntryText LIKE ?", [search])
event.source.parent.getComponent("Table").data = results
```