

Università degli Studi di Trento
Dipartimento di Economia e Management

Quantrix–Claude MCP bridge (2025):

server Model Context Protocol e server HTTP integrato
in Quantrix Modeler per l'analisi dei modelli finanziari
assistita dall'intelligenza artificiale

Nota tecnica sul prodotto della ricerca di tipo *Software*

Luca Erzegovesi

Versione	server MCP 3.0.0; server HTTP StartMcpHttpServer (2025)
Periodo di sviluppo	maggio – novembre 2025
DOI	10.5281/zenodo.23064178
Prodotto successivo	10.5281/zenodo.23078960 (versione 2026)
Accesso	file riservati, concessi su richiesta per ricerca e didattica

Trento, 14 ottobre 2026

Scheda del prodotto

Che cos'è. Un ponte software che permette a un assistente di intelligenza artificiale (Claude, di Anthropic) di leggere e modificare un modello finanziario aperto in *Quantrix Modeler*, un foglio di calcolo multidimensionale. Il ponte lavora sulla struttura propria del modello (matrici, categorie, elementi, formule), non su una griglia di celle.

Componenti. (1) un server HTTP scritto in Groovy ed eseguito dentro *Quantrix* come azione di script (`StartMcpHttpServer`, 1.489 righe); (2) un server Model Context Protocol (MCP) in Node.js, versione 3.0.0, con 12 strumenti (807 righe); (3) azioni di esempio e utilità di installazione.

A che cosa serve. Didattica e ricerca in finanza aziendale: spiegare come si calcola una cella attraverso il suo albero delle dipendenze, cambiare ipotesi e costruire scenari, esaminare i modelli di previsione finanziaria, di analisi di bilancio e di punto di pareggio usati nei corsi e consegnati dagli studenti.

Perché l'accesso è riservato. Gli strumenti automatizzano compiti che gli studenti dei corsi dell'autore devono svolgere a mano; la diffusione pubblica ne renderebbe possibile un uso improprio nelle prove d'esame. Il codice è depositato su Zenodo con file ad accesso riservato, concesso su richiesta per scopi di ricerca o didattica.

Sommario

La nota descrive la prima versione, sviluppata fra maggio e novembre 2025, del ponte fra l'assistente Claude e *Quantrix Modeler*. Il ponte è fatto di due programmi: un server HTTP che gira dentro *Quantrix* ed espone il modello aperto, e un server Model Context Protocol che traduce in chiamate HTTP i dodici strumenti offerti all'assistente. Il contributo principale è la ricostruzione dell'albero delle dipendenze di una cella, che permette all'assistente di spiegare un valore risalendo fino alle ipotesi. La nota presenta il problema, lo stato dell'arte nel 2025, il contributo originale e la sua cronologia, l'architettura, gli usi, i limiti, che hanno motivato la versione 2026 (Erzegovesi 2026a), e le condizioni di accesso. Poiché i file sono riservati e il software richiede una licenza di *Quantrix Modeler*, la nota sostituisce per i valutatori la prova diretta di installazione ed esecuzione.

Abstract

This note describes the first version, developed between May and November 2025, of a bridge between the Claude AI assistant and *Quantrix Modeler*, a multidimensional spreadsheet for financial modelling. The bridge has two parts: an HTTP server that runs inside *Quantrix* as a Groovy scripted action and exposes the open model, and a Model Context Protocol server (version 3.0.0) that maps twelve assistant tools onto HTTP calls. Its main contribution is the reconstruction of a cell's dependency tree, which lets the assistant explain a value down to the input assumptions. The files are restricted, because the tools automate tasks that students must carry out by hand; access is granted on request.

Indice

Scheda del prodotto	1
1 Il problema affrontato	3
1.1 Modelli multidimensionali e assistenti di intelligenza artificiale	3
1.2 Obiettivi	3
2 Stato dell'arte nel 2025	3
2.1 Il Model Context Protocol	3
2.2 Assistenti nei fogli di calcolo	3
2.3 Lo scripting di Quantrix	3
3 Contributo originale	4
3.1 Cosa è originale e cosa è ereditato	4
3.2 I contributi principali	4
3.3 Cronologia dello sviluppo	4
4 Architettura	5
4.1 Visione d'insieme	5
4.2 Il server HTTP dentro Quantrix	6
4.3 L'albero delle dipendenze	6
4.4 Il server MCP	7
5 Usi	7
5.1 Didattica e ricerca	7
5.2 Un esempio di scenario	7
5.3 Prova del funzionamento	7
6 Limiti e sviluppi	8
6.1 Limiti noti	8
6.2 Dalla versione 2025 alla versione 2026	8
7 Disponibilità, accesso e verifica delle date	8
7.1 Dove si trova	8
7.2 Accesso	8
7.3 Provenienza dei file	9
Ringraziamenti	9
Riferimenti bibliografici	9
A Gli strumenti del server MCP	10

1 Il problema affrontato

1.1 Modelli multidimensionali e assistenti di intelligenza artificiale

Nei corsi di finanza aziendale e di pianificazione finanziaria dell'autore i modelli si costruiscono in Quantrix Modeler (Quantrix, s.d.), erede di Lotus Improv. A differenza del foglio di calcolo a celle, Quantrix separa i dati, organizzati in matrici con categorie ed elementi nominati, dalle formule, scritte una volta per intere sezioni del modello con riferimenti per nome. È un'impostazione vicina a quella dei sistemi OLAP (Codd et al. 1993) che riduce gli errori tipici dei fogli a celle (Panko 1998; Hermans et al. 2015), ma rende anche più difficile, per chi legge il modello, ricostruire da dove viene un numero: la formula di una cella rimanda ad altre matrici, che a loro volta dipendono da altre formule.

All'inizio del 2025 gli assistenti di intelligenza artificiale generalisti sapevano ragionare su un modello finanziario, ma non vedevano il modello aperto sul computer dell'utente. L'interazione si riduceva a incollare testi e immagini nella conversazione: lento, soggetto a errori di trascrizione e inutilizzabile per modificare il modello.

1.2 Obiettivi

Il prodotto 2025 doveva verificare che fosse possibile, con gli strumenti allora disponibili:

1. far leggere a un assistente la struttura e i valori di un modello Quantrix aperto, per nome;
2. spiegare un valore ricostruendo l'albero delle sue dipendenze fino alle celle di input;
3. modificare ipotesi e applicare scenari dalla conversazione;
4. lanciare le azioni di script già presenti nel modello;
5. fare tutto questo senza modificare Quantrix, usando solo le sue interfacce di scripting.

2 Stato dell'arte nel 2025

2.1 Il Model Context Protocol

Il Model Context Protocol (MCP), pubblicato da Anthropic nel novembre 2024 (Anthropic 2024), standardizza il modo in cui un assistente si collega a fonti di dati e strumenti esterni (Model Context Protocol 2025). Un *server MCP* dichiara un insieme di strumenti, ciascuno con nome, descrizione e schema dei parametri; l'assistente decide quando chiamarli e riceve testo in risposta. Nella primavera del 2025 l'applicazione desktop di Claude permetteva di collegare server MCP locali, avviati come processi che comunicano sul canale standard (stdio). I server disponibili riguardavano soprattutto file, basi dati e servizi web.

2.2 Assistenti nei fogli di calcolo

I fornitori di fogli di calcolo a celle stavano integrando assistenti che operano sulla griglia di celle (per esempio Microsoft Copilot in Excel). Per i fogli multidimensionali come Quantrix, in cui la struttura per nomi è l'oggetto stesso della modellizzazione, non risultavano all'autore integrazioni comparabili.

2.3 Lo scripting di Quantrix

Quantrix Modeler permette di scrivere azioni in Groovy, salvate in librerie di script dentro il modello, che usano l'interfaccia di programmazione dell'applicazione: matrici, categorie, intervalli espressi con la stessa sintassi delle formule, celle con valori e formule, predecessori di una cella. Le azioni possono

importare solo i pacchetti Java esplicitamente autorizzati. Il prodotto sfrutta questa possibilità per far girare un server HTTP dentro l'applicazione.

3 Contributo originale

3.1 Cosa è originale e cosa è ereditato

Sono **originali** dell'autore i file della Tabella 1. Sono **ereditati**, e non inclusi nel deposito: Quantrix Modeler e le sue librerie, il server HTTP JLHTTP (Freeutils, s.d.), l'SDK MCP per Node.js e la libreria `node-fetch`, installati con `npm ci` dal file di blocco del 2025.

Tabella 1: File del deposito scritti dall'autore.

File	Righe	Funzione
<code>quantrix-groovy/StartMcpHttpServer.groovy</code>	1.489	Server HTTP dentro Quantrix
<code>mcp-server/quantrix-mcp-server.js</code>	807	Server MCP 3.0.0, 12 strumenti
<code>mcp-server/package.json</code> , <code>package-lock.json</code>	–	Dipendenze, bloccate al 2025
<code>quantrix-groovy/examples/</code> (3 file)	95	Azioni di esempio per <code>trigger_action</code>
<code>quantrix-groovy/setup/</code> (2 file)	48	Impostazione degli import autorizzati

3.2 I contributi principali

1. **Un server HTTP dentro Quantrix.** Un'azione di script avvia, in un thread separato, un server HTTP locale (porta 9004) che espone il modello aperto: elenco e struttura delle matrici, valori per una selezione espressa con la sintassi delle formule di Quantrix, formule, sintesi del modello, scritture, scenari ed esecuzione di azioni di script. Non richiede modifiche all'applicazione: basta aggiungere la libreria JLHTTP e autorizzare gli import.
2. **L'albero delle dipendenze.** Per una cella il server risale ricorsivamente ai predecessori calcolati da Quantrix, raccogliendo per ciascuno valore, formula e coordinate, con rilevamento dei riferimenti circolari, un limite di profondità e una memoria dei risultati (Sezione 4.3).
3. **Un server MCP pensato per l'assistente.** I dodici strumenti hanno descrizioni ed esempi di sintassi; le risposte non sono JSON grezzo ma testo strutturato: per l'albero delle dipendenze, una sintesi con i livelli, le celle di input che guidano il calcolo e le celle calcolate, seguita dall'albero completo (Sezione 4.4).
4. **Scritture e scenari.** Scrittura di un valore o di una serie di valori, con restituzione del valore precedente; scenario come insieme di modifiche `Matrice::Selezione` → valore applicate in una sola chiamata; esecuzione di azioni di script nel thread grafico di Quantrix.

3.3 Cronologia dello sviluppo

La cronologia è ricostruita dai metadati delle conversazioni Claude.ai dell'autore (date registrate dai server di Anthropic) e dalle date dei file nel Google Drive istituzionale.

Tabella 2: Cronologia del prodotto 2025.

Data	Passaggio
23/05/2025	Prime versioni dei due server, condivise in conversazione (Groovy 208 righe, MCP 395 righe)
27–29/05/2025	Server Groovy esteso (1.292 righe); progettazione dell'albero delle dipendenze e dell'analisi delle formule
30/05/2025	Server MCP 3.0.0: scritture, scenari, azioni di script
02/06/2025	Versione del server Groovy salvata nel modello <i>Financial Projections MCP server</i> ; file del server MCP salvato in Drive
04–05/06/2025	Prove di registrazione delle richieste nel server Groovy
17/09/2025	Server MCP 3.0.0 usato per l'analisi di bilancio di imprese italiane (file identico a quello in Drive)
ott.–nov. 2025	Uso su modelli didattici: analisi dell'EBIT, punto di pareggio, revisione di consegne
27/11/2025	Ultimo salvataggio 2025 del modello <i>Analisi di bilancio MCP server</i> , da cui è estratto il server Groovy depositato
29/11/2025	Prima analisi dell'evoluzione verso la versione 2026
30/09/2026	Prova completa dei file depositati e deposito su Zenodo, DOI 10.5281/zenodo.23064178

4 Architettura

4.1 Visione d'insieme

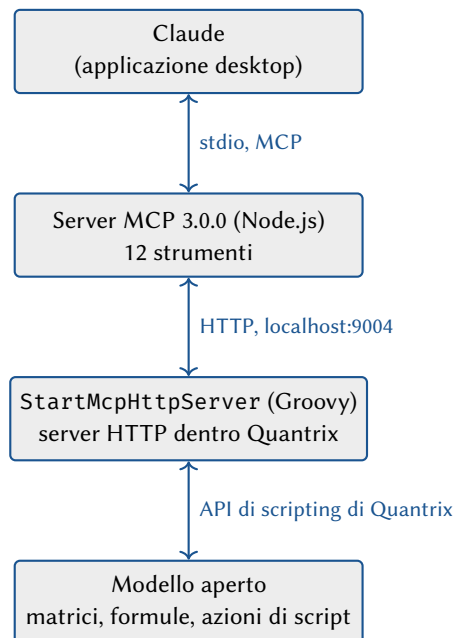


Figura 1: Architettura del ponte 2025. L'assistente chiama gli strumenti del server MCP; il server li traduce in richieste HTTP al server Groovy, che opera sul modello aperto in Quantrix.

L'assistente non tocca mai direttamente il modello: ogni operazione passa per il server MCP e per il server dentro Quantrix (figura 1).

4.2 Il server HTTP dentro Quantrix

StartMcpHttpServer è un'azione di script da aggiungere a una libreria del modello. Quando la si esegue crea un thread con un server JLHTTP sulla porta 9004 e registra un gestore per ogni percorso. I gestori usano le classi del motore di calcolo di Quantrix: il modello, le matrici cercate per nome, gli intervalli (QNodeRange) costruiti dalla stessa sintassi delle formule (Ricavi:2024:consuntivo) e le celle, con valore, formula e predecessori. Le scritture sono eseguite in un blocco con privilegi elevati, come richiesto dall'ambiente di scripting; le azioni di script sono eseguite nel thread grafico dell'applicazione (SwingUtilities), in modalità asincrona o sincrona.

Tabella 3: Percorsi del server HTTP usati dal server MCP.

Percorso	Metodo	Funzione
/matrices	GET	Elenco delle matrici con le loro categorie
/matrix	GET	Struttura di una matrice (categorie ed elementi) o valore di una cella
/selection, /get-values	GET	Valori di una selezione
/cell-analysis	GET	Formula di una cella e predecessori immediati
/dependency-tree	GET	Albero delle dipendenze, con profondità massima
/model-summary	GET	Matrici, categorie, numero di elementi e di celle
/clear-cache	GET	Svuota la memoria degli alberi delle dipendenze
/set-value, /set-values	GET	Scrittura di uno o più valori
/create-scenario	POST	Insieme di scritture descritto in JSON
/trigger-action	GET	Esecuzione di un'azione di una libreria di script

Il file contiene anche alcuni percorsi sperimentali delle prime prove (messaggi di una chat memorizzati in una matrice, creazione di una matrice vuota, un suggeritore di formule segnaposto) che il server MCP non usa.

4.3 L'albero delle dipendenze

Data una matrice e una cella, il server costruisce l'albero ricorsivamente (buildDependencyTree). Per ogni nodo registra nome della cella, valore, tipo, formula (testo del lato destro) e coordinate per categoria; poi chiede a Quantrix i predecessori della cella, cioè gli intervalli da cui dipende la sua formula, e scende nel primo elemento di ciascuno. La discesa si ferma:

- sulle celle senza predecessori, che diventano *foglie* (valori di input);
- su una cella già presente nel cammino corrente, segnalata come riferimento circolare;
- alla profondità massima, dieci livelli se non diversamente indicato.

L'insieme delle celle visitate è copiato per ogni ramo, così una cella raggiunta per due percorsi diversi compare in entrambi senza essere scambiata per un ciclo. L'albero è conservato in una memoria indicizzata da matrice, cella e profondità, che si svuota con `clear_cache`. Una semplice analisi del testo della formula ne classifica le operazioni (somme, prodotti, condizioni, aggregazioni).

4.4 Il server MCP

Il server è scritto in JavaScript per Node.js con l'SDK MCP ufficiale e comunica con l'applicazione Claude sul canale standard. Dichiaro dodici strumenti (appendice A) e, per ciascuno, chiamo il percorso corrispondente del server HTTP. Le risposte sono riformulate per l'assistente. Per `explain_cell` il server calcola dall'albero una sintesi e restituisce un testo di questa forma:

```
# Analysis of <cella>
Matrix · Current Value · Value Type · Dimensional Context
Formula: <formula> Formula Analysis: complexity, operations
Dependency Summary: total dependency levels · input cells · calculated cells · circular references
Input Values (driving this calculation): <foglia>: <valore> ...
Complete Dependency Tree: <albero indentato, con formula o "Input value" per ogni nodo>
```

Le scritture riportano matrice, selezione, nuovo valore e valore precedente, e suggeriscono di verificare l'effetto con `explain_cell`.

5 Usi

5.1 Didattica e ricerca

Nel 2025 il ponte è stato usato dall'autore su modelli didattici e di ricerca:

- **Spiegare i modelli.** Ricostruire come si calcola una voce di un conto economico previsionale o un indice di bilancio, fino alle ipotesi da cui dipende, e farne una spiegazione in linguaggio naturale.
- **Analisi di bilancio di imprese italiane.** Lettura di conto economico e stato patrimoniale riclassificati in un modello di analisi di bilancio (settembre e novembre 2025).
- **Analisi *what-if*.** Modifica delle ipotesi di un modello di previsione o di punto di pareggio e lettura dei risultati; scenari composti da più modifiche.
- **Revisione delle consegne.** Lettura semiassistita di modelli *Quantrix* consegnati dagli studenti, come ausilio al docente, che resta responsabile della valutazione.

5.2 Un esempio di scenario

Uno scenario è un insieme di modifiche indicate come `Matrice::Selezione`. L'assistente invia allo strumento `create_scenario` un oggetto come il seguente (i nomi sono illustrativi):

```
{ "scenario_name": "Crescita ridotta",
  "changes": { "Ipotesi::Crescita ricavi:2026": 0.02,
               "Ipotesi::Incidenza costi variabili:2026": 0.62 } }
```

Il server Groovy applica le modifiche una per una e restituisce, per ciascuna, l'esito e il valore precedente, con l'elenco delle modifiche non riuscite.

5.3 Prova del funzionamento

Il 30 settembre 2026 i file depositati sono stati eseguiti insieme: server MCP 3.0.0, con le dipendenze installate dal file di blocco del 2025, collegato a `StartMcpHttpServer` in esecuzione in una copia del modello didattico di analisi di bilancio del 27 novembre 2025 (25 matrici). Hanno funzionato `matrices`, `matrix`, `model_summary`, `selection`, `get_values`, `analyze_formula`, `explain_cell` (albero

delle dipendenze), `set_value`, `set_values` e `clear_cache`; il modello è stato poi riportato ai valori originali. `create_scenario` e `trigger_action` non sono stati provati in quell'occasione. Che il server in esecuzione fosse quello del 2025 è confermato dall'assenza del percorso `/matrix-formulas`, aggiunto solo nel gennaio 2026.

6 Limiti e sviluppi

6.1 Limiti noti

- **Un solo modello.** Il server opera sul modello attivo, e l'azione va avviata dentro un modello; non c'è modo di scegliere il modello dalla conversazione.
- **Scritture non controllate.** `set_value` scrive nella prima cella della selezione, anche se la selezione ne comprende più d'una; `set_values` scrive tanti valori quanti ne trova, senza rifiutare una selezione di dimensione diversa. Una selezione imprecisa può quindi modificare celle non volute.
- **Ricalcolo e memoria.** Dopo una scrittura le celle dipendenti mostrano il nuovo valore solo alla lettura successiva, perché il modello ricalcola dopo la risposta; gli alberi delle dipendenze in memoria non si aggiornano da soli e vanno svuotati con `clear_cache`.
- **Selezioni e nomi.** I nomi di elementi con caratteri speciali, come i punti, vanno scritti tra apici come nelle formule di *Quantrix*; per un intervallo l'albero delle dipendenze segue solo la prima cella.
- **Esito degli script.** In modalità asincrona `trigger_action` segnala l'avvio dell'azione, non il suo risultato.
- **Licenza.** Il ponte richiede una licenza di *Quantrix Modeler* e una sottoscrizione a Claude.

6.2 Dalla versione 2025 alla versione 2026

Questi limiti hanno guidato la versione 2026 (Erzegovesi 2026a), depositata come prodotto distinto: il server Groovy è stato riscritto come plugin Java che serve tutti i modelli aperti, il server MCP è passato da 12 a 40 strumenti con verifiche del modello attivo e rifiuto delle scritture che non corrispondono alla selezione, e il metodo di lavoro è stato codificato in una skill di Claude.

7 Disponibilità, accesso e verifica delle date

7.1 Dove si trova

Il prodotto è depositato su Zenodo (Erzegovesi 2026b) come record pubblico con file riservati, licenza chiusa, versione «3.0.0 (2025)» e date di creazione 2025-05-23/2025-11-27 nei metadati. Il deposito contiene 11 file: i due server, il file di blocco delle dipendenze, le azioni di esempio, le utilità di installazione, un README e un file `SOURCES.md` con origine, data e impronta SHA-256 di ogni file. Il record 2026 è collegato a questo con la relazione *Continues*.

7.2 Accesso

L'accesso ai file è concesso su richiesta, tramite Zenodo, per scopi di ricerca o didattica. Non sono inclusi *Quantrix Modeler*, le sue librerie, *JLHTTP* e le dipendenze npm, né modelli con dati di studenti.

7.3 Provenienza dei file

Ogni file è una copia senza modifiche, o un'azione estratta senza modifiche, di un file salvato dall'autore nel 2025 nel Google Drive istituzionale (cartella del laboratorio di pianificazione finanziaria 2025–26). Due fonti indipendenti attestano il server MCP: il file in Drive, modificato il 02/06/2025, è identico byte per byte al codice allegato alla conversazione Claude.ai del 17/09/2025 (SHA-256 3ed5c54de9815786...). Il server Groovy depositato è quello del modello del 27/11/2025 e coincide, salvo una riga vuota, con quello salvato nel modello del 02/06/2025. Gli elenchi delle conversazioni e il codice allegato, tratti dall'esportazione dei dati di Claude.ai, sono conservati dall'autore.

Ringraziamenti

Uso di assistenti di intelligenza artificiale. Il codice del 2025 è stato scritto con l'aiuto di Claude (Anthropic), nelle conversazioni elencate nella cronologia. La preparazione del deposito e la stesura di questa nota sono state assistite da Claude. Scelte progettuali, contenuti e verifiche sono dell'autore.

Riferimenti bibliografici

Anthropic. 2024. «Introducing the Model Context Protocol». novembre. <https://www.anthropic.com/news/model-context-protocol>.

Codd, E. F., S. B. Codd, e C. T. Salley. 1993. *Providing OLAP to User-Analysts: An IT Mandate*. E. F. Codd & Associates.

Erzegovesi, Luca. 2026a. *Quantrix–Claude MCP bridge (2026): Quantrix Modeler plugin, Model Context Protocol server and Claude skill for AI-assisted financial modelling*. Zenodo, released. <https://doi.org/10.5281/zenodo.23078960>.

Erzegovesi, Luca. 2026b. *Quantrix–Claude MCP bridge: Model Context Protocol server and embedded HTTP server for AI-assisted analysis of Quantrix Modeler financial models (2025)*. Zenodo, released. <https://doi.org/10.5281/zenodo.23064178>.

Freeutils. s.d. *JLHTTP: Java Lightweight HTTP Server*. <https://www.freeutils.net/source/jlhttp/>.

Hermans, Felienne, Martin Pinzger, e Arie van Deursen. 2015. «Detecting and Refactoring Code Smells in Spreadsheet Formulas». *Empirical Software Engineering* 20 (2): 549–75.

Model Context Protocol. 2025. «Model Context Protocol: Specification and documentation». <https://modelcontextprotocol.io>.

Panko, Raymond R. 1998. «What We Know About Spreadsheet Errors». *Journal of End User Computing* 10 (2): 15–21.

Quantrix. s.d. *Quantrix Modeler*. <https://www.quantrix.com>.

A Gli strumenti del server MCP

Tabella 4: I 12 strumenti del server MCP 3.0.0.

Strumento	Parametri	Funzione
<code>matrices</code>	–	Elenco delle matrici del modello
<code>matrix</code>	<code>name</code> , <code>coords</code>	Struttura di una matrice o valore di una cella
<code>selection</code>	<code>name</code> , <code>coords</code>	Valori di una selezione
<code>get_values</code>	<code>matrix</code> , <code>selection</code>	Valori correnti, per verifica
<code>explain_cell</code>	<code>matrix</code> , <code>range</code> , <code>max_depth</code> , <code>include_tree</code>	Spiegazione di una cella con albero delle dipendenze
<code>analyze_formula</code>	<code>matrix</code> , <code>range</code>	Formula e dipendenze immediate
<code>model_summary</code>	–	Sintesi del modello
<code>clear_cache</code>	–	Svuota la memoria degli alberi
<code>set_value</code>	<code>matrix</code> , <code>selection</code> , <code>value</code>	Scriva un valore
<code>set_values</code>	<code>matrix</code> , <code>selection</code> , <code>values</code>	Scriva una serie di valori
<code>create_scenario</code>	<code>scenario_name</code> , <code>changes</code>	Applica un insieme di modifiche
<code>trigger_action</code>	<code>library_name</code> , <code>script_name</code> , <code>mode</code>	Esegue un'azione di script