**Mars Landing Media – DotWalk**

Did you know you can harness the power of dot-walking dynamically within GlideAjax scripting? Introducing **DotWalk**, a solution that brings the flexibility of dot-walking into your client-server interactions in ServiceNow. With DotWalk, you can access related fields and records across complex relationships, all from a client-side script, without writing extensive back-end code. It’s fast, it’s efficient, and it takes GlideAjax scripting to the next level. Elevate your ServiceNow development game with DotWalk!

To configure the sysparm\_variables parameter and structure the answer output correctly, it’s essential to follow a consistent **naming convention** to ensure clarity and proper mapping between client-side and server-side logic.

**1. Configuring**sysparm\_variables

The sysparm\_variables parameter should contain a comma-separated list of field names or dot-walking paths that specify the data you want to retrieve. The naming convention and formatting guidelines are as follows:

**Naming and Formatting:**

* **Single-level fields**:
  + Use the exact field name from the table (e.g., name, email, manager).
* **Dot-walking paths**:
  + Use a hyphen (-) to represent dot-walking instead of the dot (.) used in the server-side GlideRecord object.
  + Example: caller\_id-location translates to caller\_id.location in GlideRecord logic.
  + Each segment of the path corresponds to a field in the related table, ensuring dot-walking works dynamically.

**Examples of**sysparm\_variables**:**

* caller\_id-location: Fetches the location of the caller\_id referenced field.
* manager-department-name: Retrieves the name of the department referenced by manager.
* location-country,location-city: Retrieves the country and city fields of the location referenced field.

**2. Output for**answer

The answer returned by the script include will be a JSON object where:

* Each key corresponds to the fields or dot-walked paths specified in sysparm\_variables.
* The keys in the JSON object should match the dot-walked field names in sysparm\_variables, but the hyphens (-) are replaced with underscores (\_) for JSON compatibility.

**Naming and Formatting:**

* **Keys**:
  + Field paths in sysparm\_variables are transformed into JSON-safe keys by replacing hyphens (-) with underscores (\_).
  + Example:
    - Input: manager-department-name
    - Output Key: manager\_department\_name
* **Values**:
  + Field values are returned as strings unless otherwise specified (e.g., reference fields might return sys\_id unless explicitly dot-walked).

**Example Output:**

For the sysparm\_variables value:



The answer output could look like this:

A black screen with green text

Description automatically generated

**3. Error Handling**:

If a specified field or path in sysparm\_variables does not exist or fails during processing, the key should be present in the answer output with a null or empty value.