Asset Lifecycle Dashboard API

1. Navigate to System Web Services → Scripted REST APIs
2. Click on New
3. Give name as "Asset Lifecycle Dashboard API"
4. Accept the API ID as "asset_lifecycle_dashboard_api", Application as "Global" (auto-generated read only) and API Namespace as **generated** (read only).
5. Protection policy is read-only and "None" is selected.
6. Click Submit
7. Back in the Scripted REST APIs page (sys_ws_definition), select the newly created API (Asset Lifecycle Dashboard API)
8. In the Resource Related List, select "New"
9. Type name as "IT Asset Lifecycle"
10. HTTP Method: GET
11. Relative Path: /asset/{asset_tag}
12. In the script box, write script to select details from the alm_asset table

```javascript
(function process( /*RESTAPIRequest*/ request, /*RESTAPIResponse*/ response) {

    var asset_tag = request.pathParams.asset_tag;
    var result = {};
    // get asset details
    var grAsset = new GlideRecord('alm_asset');
    grAsset.addQuery('asset_tag', asset_tag);
    grAsset.query();
    if (!grAsset.next()) {
        response.setStatus(404);
        return {
            error: "Asset not found for tag: " + asset_tag
        };
    }
    result.asset = {
        // assign values here
        tag: grAsset.asset_tag.toString(),
        name: grAsset.display_name.toString(),
        model: grAsset.model.display_value,
        ci: grAsset.ci.getDisplayValue(),
        assigned_to: grAsset.assigned_to.getDisplayValue(),
        status: grAsset.substatus.getDisplayValue(),
        warranty_expiry: grAsset.warranty_expiration ?
grAsset.warranty_expiration.toString() : "N/A"

    };
    if (!gs.nil(grAsset.ci)) {
        // get incident details
        result.incidents = [];
```

```javascript
        var grIncident = new GlideRecord('incident');
        gs.info('IT Asset Lifecycle :: grAsset.ci.sys_id:' + grAsset.ci.sys_id);
        grIncident.addQuery('cmdb_ci', grAsset.ci.sys_id);
        // in lieu of - grIncident.addQuery('active', true)
        grIncident.addActiveQuery();
        grIncident.query();
        while (grIncident.next()) {
            result.incidents.push({
                number: grIncident.number.toString(),
                short_description: grIncident.short_description.toString(),
                state: grIncident.state.getDisplayValue()
            });
        }
        // get open requests
        result.requests = [];
        var grRequest = new GlideRecord('sc_request');
        grRequest.addQuery('cmdb_ci', grAsset.ci.sys_id);
        grRequest.addActiveQuery();
        grRequest.query();
        while (grRequest.next()) {
            result.requests.push({
                number: grRequest.number.toString(),
                requested_for: grRequest.requested_for.getDisplayValue(),
                stage: grRequest.stage.toString()
            });
        }


        // get Disposal Info (if retired)
        if (grAsset.ci.install_status == 7) { // 7 = Retired
            result.disposal = {
                retired_date: grAsset.retired ? grAsset.retired.toString() :
"N/A",
                retired_by: grAsset.retired_by ?
grAsset.retired_by.getDisplayValue() : "N/A"
            };
        }
    }

    return result;
})(request, response);
```

POSTMAN URL:

GET https://<YOUR_INSTANCE>/api/< API
Namespace>/asset_lifecycle_dashboard_api/asset/LAP12345

Postman Response (As per Demo data inserted)

```json
{
    "result": {
        "asset": {
            "tag": "LAP12345",
            "name": "LAP12345 - Dell Inc. Alienware M14x",
            "model": null,
            "ci": "SN123456 - Alienware M14x",
            "assigned_to": "Abel Tuter",
            "status": "",
            "warranty_expiry": "N/A"
        },
        "incidents": [
            {
                "number": "INC0012243",
                "short_description": "Laptop not booting",
                "state": "New"
            },
            {
                "number": "INC0012244",
                "short_description": "screen flickering",
                "state": "New"
            }
        ],
        "requests": [
            {
                "number": "REQ0000001",
                "requested_for": "System Administrator",
                "stage": "requested"
            },
            {
                "number": "REQ0010002",
                "requested_for": "Adam Meyers",
                "stage": "requested"
            },
            {
                "number": "REQ0010001",
                "requested_for": "Aeron Goldtree",
                "stage": "requested"
            }
        ]
    }
}
```