Mission Upload Bash Script Sample

Below is a sample script to upload missions defined in JSON. By default it's configured to upload missions defined in a JSON file (missions.json) in the same directory as the script, to your local Elements instance.

You can format the missions in the JSON in the same way you see in the example earlier in this document.

This script will also update missions if they already exist in the database.

```
#!/usr/bin/env bash

function post_item() {

    url=$1
    secret=$2
    definition=$3
    name=$(jq -r  '.name' <<< "${definition}")
    id=$(curl -k -X GET "${url}/mission/${name}" | jq -r '.id')
    if [ -z "$id" ]
    then
          echo "${id} is NULL, Creating Item"
          curl -k -X POST \
          "${url}/mission" \
          -H 'Cache-Control: no-cache' \
          -H 'Content-Type: application/json' \
          -H "Elements-SessionSecret: ${secret}" \
          -d "${definition}"
    else
          echo "${id} is NOT NULL, Updating Item"
          item=$(jq --arg id ${id} '{id: $id} + .' <<< "${definition}")
          curl -k -X PUT \
          "${url}/mission/${id}" \
          -H 'Cache-Control: no-cache' \
```

```
          -H 'Content-Type: application/json' \
          -H "Elements-SessionSecret: ${secret}" \
          -d "${item}"
    fi
    return $?

}

echo -n "Definitions: (missions.json): "
read definitions

echo -n "API (http://localhost:8080/api/rest): "
read url

echo -n "Username:  "
read username

echo -n "Password:  "
read -s password

definitions=${definitions:-"missions.json"}

if [ ! -f ${definitions} ]
then
    echo "Definitions not found: ${definitions}"
    exit 1
fi

url=${url:-"http://localhost:8080/api/rest"}

session=$(curl -k -X POST \
  "${url}/session" \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json' \
  -d "{ \"userId\" : \"${username}\", \"password\" : \"${password}\" }")

code=$?

if [ ${code} -ne 0 ]
then
    echo "Failed to create session."
```

```
    exit 1
else
    echo "Successfully created session."
    secret=$(echo ${session} | jq -r ".sessionSecret")
fi

jq -c '.[]' $definitions | while read item;
do
    post_item "${url}" "${secret}" "${item}"
done
```