Documentation for Tellus CI-API (Serial Number API)

Thursday, October 05, 2017 12:27 AM

TellusCI-API or Tellus Continue Integration-Application Programming Interface is an endpoint running on Tellus side to provide resource for Continues Integration. As of now, Tellus CI-API or here on referred as "CI-API" is only able to set or release serial numbers on a specific pool on Tellus side

1. **Block specific or random serial number from default or mentioned pool name**:

   This end point will be responsible to blocking the serial number from the mentioned pool name. Following is the flow of operation performed by request API.

   i. Check availability for the requested serial number and other resources mentioned in the request : *CI-API accepts the requested JSON and validates the request. Validation for serial number is to check if serial number is not alphanumeric or alphabets and also check if serial number exists in the default or given pool name. For other resource mentioned like caccess,poolname,fdsav2stage and fdsav2overides will be validated by checking if they exist in the configuration file.*
   ii. Clear the FDSAV2 Stage and FDSAV2 Overrides: *After validating the request, CI-API will clear all the fdsav2session i.e., clears the fdsav2stage and fdsav2overides. This operation will be performed irrespective of either the serial number was set with fdsav2sessions or not.*
   iii. Block the serial number along with it set the fdsav2stage, fdsav2overides and caccess if mentioned: *After clearing the fdsav2session, CI-API calls internal serial number blocking call and blocks the serial number.*

Request:

Method: POST
BODY: JSON
ENDPOINT: http://tellusstgb01:8080/tellusci/ci/request
Request Field:

```
{
    "RequestJson": {
        serialNumber:
        vmsusername:
        poolname:
        caccess:
        fdsav2stage:
        fdsav2overides:
    }
}
```

Description:

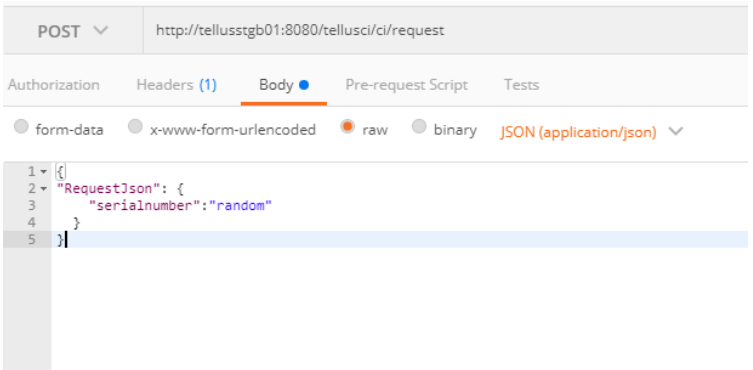
i. serialnumber: Serial number which needs to be blocked. Mandatory key with two possible value.
   - specific serialnumber: Specific serial number will be blocked else will throw bad request with error field showing error message
   - random: Key value given has "random". Any random serial number will be picked from mentioned poolname.

ii. [vmsusername]: vmsusername for which settings needs to be done. Defaults to FDSQAR_C
iii. [poolname]: name of the pool from which the serial number needs to be picked. Default CID
iv. [caccess]: caccess that needs to be set to a serial number. Default LARGESETCACCESS
v. [fdsav2stage]: Fdsav2stage settings that needs to be done for a serial number. Default null
vi. [fdsav2overides]: fdsav2overides settings that needs to be done for a serial number after fdsav2stage has been set. If fdsav2 overides are mentioned without fdsav2stage then API will reject the request and throw the error default null

Note 1: Specified keys in [] are optional.
Note 2: vmsusername,poolname,caccess even though user defined will be verified on the configuration list. If given value is available in the list then the request will be executed. List for values are as follows:
Vmsusernamae: FDSQAR_C
Caccess: LARGESETCACCESS,LARGESETCACCESS_00HCR_INCLUDED,LARGESETCACCESS_IS,LARGESETCACCESS_NO_00ORN,LARGESETCACCESS_NO_IB,LAR GESETCACCESS_QR_PA2_NOBATCHER,LARGESETCACCESS_RT,OI_AUTOMATION,RTS_DELAYED,RTS_REALTIME,largesetcaccess_admin
Poolname: CID
For now these are the values mentioned in the list. New values can be added/removed

Examples: All the example shown below are ran on Postman

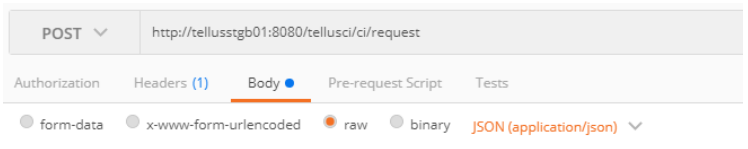
i. Request for a random serial number:



ii.

```
{
"RequestJson": {
    "serialnumber":"random"
  }
}
```

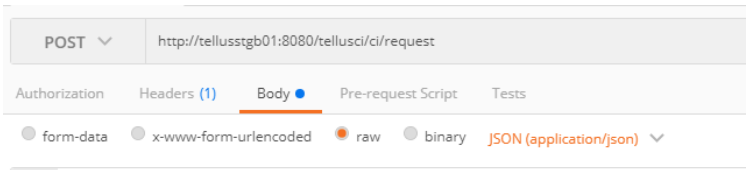
iii. Request for specific serial number:



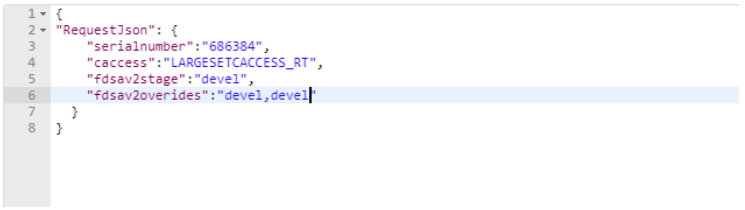
iv.

```
{
"RequestJson": {
    "serialnumber":"686384"
  }
}
```

v. Request for specific serial number with fdsav2session settings:



vi.



Response:

Response Fields:

If success: Status OK

{

"ResponseJson": {

serialnumber:

requestId:

success:

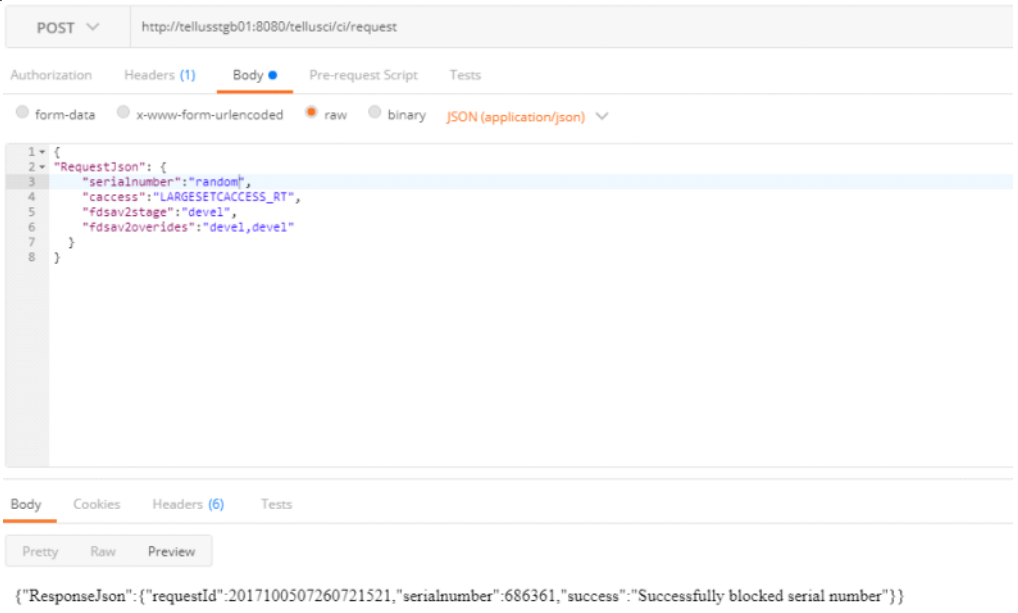
}

}

Description:

1) Serialnumber: Serialnumber which was blocked. This format is usefull if the request type if for random serialnumber. For now serialnumber field will be returned for both specific serial number and random serial number.
2) requistId: A unique 19-digit code generated for every request made. This id is useful to uniquely identify the request. This id is used at the server side to capture the log and information on who have raised the request.
3) Success: success message to show what operation was performed successfully.

Example:



If error: Status Internal Server Error/Bad Request

{

"ResponseJson": {

error:

requestId:

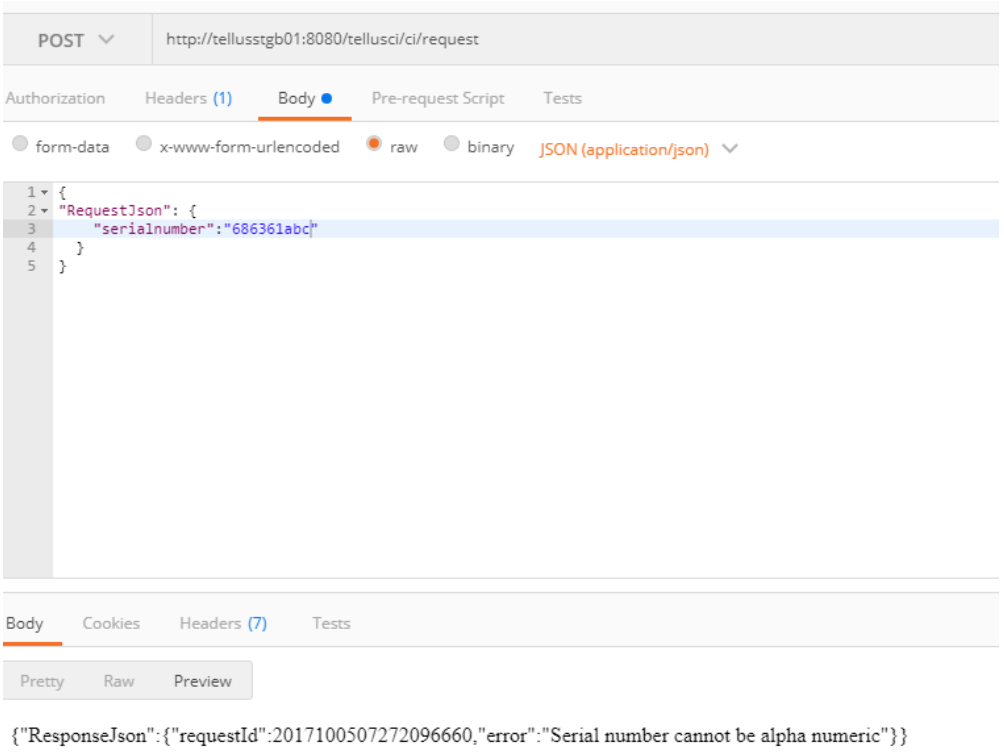
}

}

Description:

1) requistId: A unique 19-digit code generated for every request made. This id is useful to uniquely identify the request. This id is used at the server side to capture the log and information on who have raised the request.
2) Error: Error message to show what operation was not performed successfully.

Example:



2. **Releasing the serial number**:

   a. Validate the request: *CI-API accepts the requested JSON and validates the request. Validation for serial number is to check if serial number is not alphanumeric or alphabets and also check if serial number was blocked in the first place to release*
   b. Clear fdsav2session: *After validating, API clears the fdsav2sessions*
   c. Release the serial number: *After completing the session clearing, API calls the internal call to release the serial number.*

Method: POST
BODY: JSON
ENDPOINT: http://tellusstgb01:8080/tellusci/ci/release
Request Field:

```
{
    "RequestJson": {
        serialNumber:
        [vmsusername]:
        [poolname]:
        [caccess]:
        [fdsav2stage]:
        [ fdsav2overides]:
    }
}
```

Description:

i. serialnumber: Serial number which needs to be blocked. Mandatory key with two possible value.
   - specific serialnumber: Specific serial number will be blocked else will throw bad request with error field showing error mess age
   - random: Key value given has "random". **During the release serial number cannot be random**. If encountered then API will reject the request with error.

ii. [vmsusername]: vmsusername for which settings needs to be done. Defaults to FDSQAR_C. **During the release request vmsusername will be ignored**
iii. [poolname]: name of the pool from which the serial number needs to be picked. Default CID. **During the release request poolname will be ignored**
iv. [caccess]: caccess that needs to be set to a serial number. Default LARGESETCACCESS. **During the release request caccess will be ignored**
v. [fdsav2stage]: Fdsav2stage settings that needs to be done for a serial number. Default null. **During the release request fdsav2stage will be ignored**
vi. [fdsav2overides]: fdsav2overides settings that needs to be done for a serial number after fdsav2stage has been set. default null. **During the release request fdsav2overides will be ignored**

Note 1: Specified keys in [] are optional.
Note 2: vmsusername,poolname,caccess even though user defined will be verified on the configuration list. If given value is available in the list then the request will be executed. List for values are as follows:
Vmsusernamae: FDSQAR_C
Caccess: LARGESETCACCESS,LARGESETCACCESS_00HCR_INCLUDED,LARGESETCACCESS_IS,LARGESETCACCESS_NO_00ORN,LARGESETCACCESS_NO_IB,LARGESETCACCESS_QR_PA2_NOBATCHER,LARGESETCACCESS_RT,OI_AUTOMATION,RTS_DELAYED,RTS_REALTIME,largesetcaccess_admin
Poolname: CID
For now these are the values mentioned in the list. New values can be added/removed

Response for the release will be

i. For positive response:
   Status code: 200
   Response type: JSON

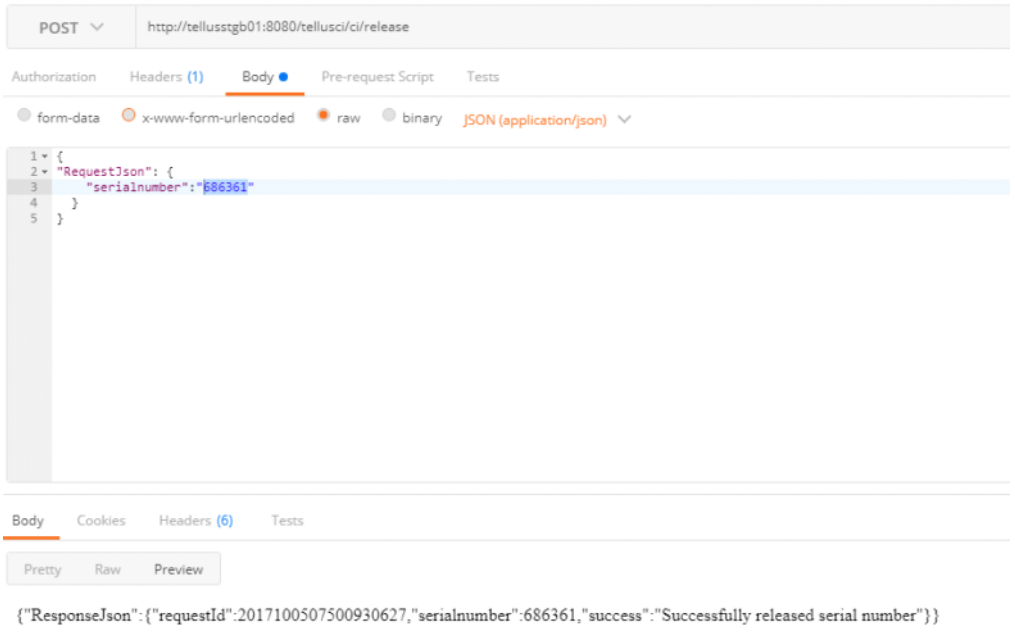
```
{
    "ResponseJson": {
        serialnumber:
        requestId:
        success:
    }
}
```

where
1. serialnumber : serialnumber which was released.

2. requestId: A unique 19-digit code generated for every request made. This id is useful to uniquely identify the request. This id is used at the server side to capture the log and information on who have raised the request. 3. success: Success message for the request made

Example:

1. For successful release

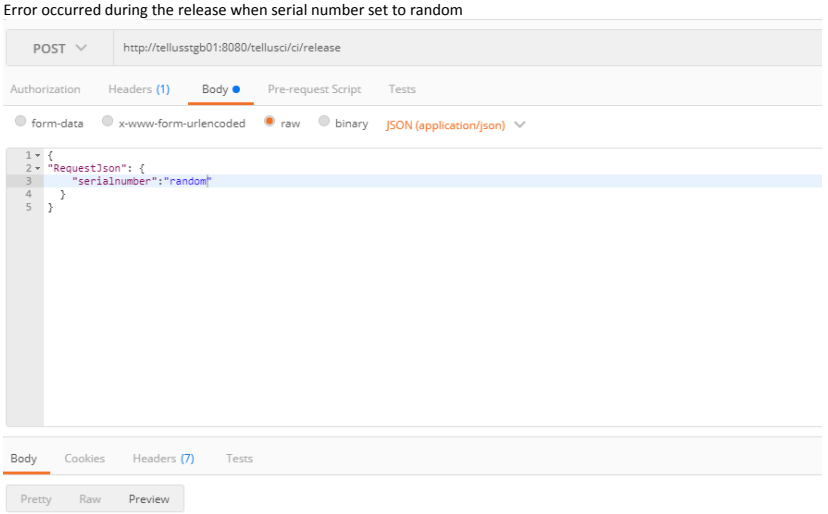


ii. For negative response:
Status code: 400/500
Response type: JSON

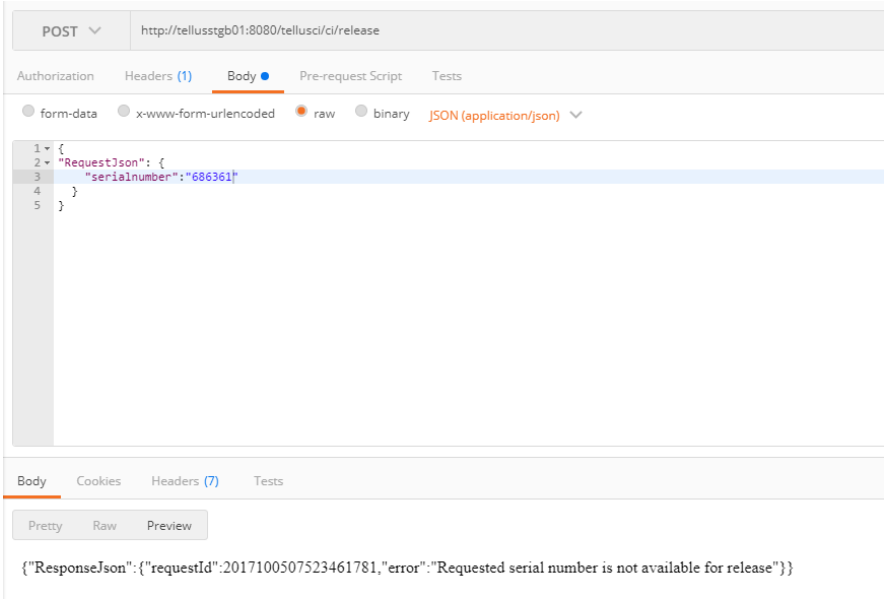
```
{
  "ResponseJson": {
        error:
       requestId:
    }
}
```

Example:

1. Error occurred during the release when serial number set to random



2. Error occurred during release when tried to release a serial number which was not blocked in the first place



Description:

i. requistId: A unique 19-digit code generated for every request made. This id is useful to uniquely identify the request. This id is used at the server side to capture the log and information on who have raised the request.
ii. Error: Error message to show what operation was not performed successfully.