

Vppsim User Guide

Description

vppsim provides server emulation of the API's for Volume Purchase Program(VPP) documented in the Mobile Device Management Protocol Reference.

Synopsis

vppsim <command> [[<option> [<argument> ...]] ...]

Commands

start [-port port] [-output path] [configpath ...]

Start an instance of vppsim in server mode. By default, no VPP license and users are registered and server activity is logged in JSON format to stdout. The default or specified configuration is output at the start, including the token and token keys required for authentication. The process listens on port 8080. Use the option **-port** to specify an alternate port. Use the option **-output** to redirect logs to the specified file location. Specify one or more optional configuration files to override the default configuration.

stop [-port port] [-host address]

Stop an instance of vppsim in server mode. By default, the command will attempt to stop a server instance running on the local host at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host.

token [-port port] [-host address] [-output path]

Outputs the current token used by the server to authenticate all the VPP API requests. By default, the command will attempt to output the token of the server instance running on the local host at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host. If **-output** is specified the token will be written to the file at the given path.

configure [-port port] [-host address] configpath ...

Load one or more configuration files into a server instance. One or more configuration files, separated by spaces, must be specified following the command. Values in the specified configurations will immediately override settings in the running server instance. Overriding the settings of a repeating operation, such as user record association or VPP license insertion, will cause it to stop and commence its newly specified behavior. By default, this command will attempt to connect to a server instance running on the local host at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host.

reset [-port port] [-host address]

Reset a server instance to the originally specified configuration. By default, this command will attempt to connect to a server instance running on the local host at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host.

log [-port port] [-host address] [-output path]

Output the current stream of server log entries to stdout. By default, this command will attempt to connect to a server instance running on the localhost at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host. If **-output** is specified the contents of the log will be written to the file at the given path.

addvpplicense [-port port] [-host address] -adamId [adamId] [-count number]

Add an iTunes app or book specified by a valid adamId number and number of licenses. By default, this command will attempt to connect to a server instance running on the local host at port 8080 and add the VPP license record. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host. Use the option **-count** to specify the number of licenses to create.

associateusers [-port port] [-host address] {clientIdStr... | [-input path] | -all}

Changes the state of the specified user to Associated with randomly generated itsIdHash value for the user. One or more user clientIdStr's separated by spaces or the option **-input** followed by a path for JSON formatted file containing user clientIdStr's must be specified or the option **-all** must be specified. Only users with status *Registered* can be changed status to *Associated*. By default, this command will attempt to connect to a server instance running on the local host at port 8080. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host.

dumplenses [-port port] [-host address] [-output path]

Output the current list of licenses loaded in a server instance. The license records are output in JSON format as an array of dictionaries. There is one dictionary entry per license. By default, this

command will attempt to connect to a server instance running on the local host at port 8080 and write the license records to stdout. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host. Use the option **-output** to write the license records to a specified file on the local host.

dumpusers [-port port] [-host address] [-output path]

Output the current list of users loaded in a server instance. The user records are output in JSON format as an array of dictionaries. There is one dictionary entry per user. By default, this command will attempt to connect to a server instance running on the local host at port 8080 and write the user records to stdout. Use the option **-port** to specify an alternate port. Use the option **-host** to specify an alternate host. Use the option **-output** to write the user records to a specified file on the local host.

license

Display the end user software license agreement.

acknowledgements

Display third-party acknowledgments.

help

Display the complete documentation.

Options

-port <number>

(All commands) Specify the network port.

-host <hostname>

(All commands) Specify the hostname.

-output <file>

(start, token, dumplicenses, dumpusers, and log command only) Specify the output destination for the command. An optional file path on the local host can be specified. If a file path is not specified, the output is written to stdout.

-input <file>

(associateusers command only) Specify a file path on the local host whose file contents contain JSON formatted clientIdStrs of users to change the status to associated.

-count <number>

(addvpplicense command only) Specify the number of VPP licenses to be created for the given valid iTunes adamId

-all

(associateusers command only) Change the status to associated for all users on the server

Examples

To start the server and have it listen on port 8000:

```
vppsim start -port 8000
```

To start the server, have it listen on port 9000, log activity to /tmp/vppsim.log and override the default configuration with settings from two configuration files:

```
vppsim start -port 9000 -output /tmp/vppsim.log  
vppsim_config1.json vppsim_config2.json
```

To command a server running on localhost and listening on port 8000 to output the token used to authenticate VPP requests

```
vppsim -port 8000 token
```

To command a server running on localhost and listening on port 8000 to load an additional configuration file named vppsim_config3.json:

```
vppsim -port 8000 configure vppsim_config3.json
```

To command a server running on localhost and listening on port 8000 to add 100 licenses for Podcast app:

```
vppsim -port 8000 addvpplicense -adamId 525463029 -count 100
```

To command a server running on localhost and listening on port 8000 to associate users with clientIdStr's KWYNSL-JJSNW and EJQMSA-PQJSN:

```
vppsim -port 8000 associateusers KWYNSL-JJSNW EJQMSA-PQJSN
```

To command a server running on localhost and listening on port 8000 to change the state of all users to *Associated*:

```
vppsim -port 8000 associateusers -all
```

To command a server running at myhost.example.com and listening on port 8000 to change

status of users with `clientIdStr`'s KWYNSL-JJSNW and EJQMSA-PQJSN to associated using a JSON formatted file defined in `/path/user_associations.json`:

```
vpptest -host myhost.example.com -port 8000 associateusers -  
input /path/user_associations.json
```

The `clientIdStr`'s of the users to be associated are specified in the `user_association.json` file as:

```
{  
  "clientIdStr": ["KWYNSL-JJSNW", "EJQMSA-PQJSN"]  
}
```

To command a server running at `myhost.example.com` and listening on port 9000 to output its current license records and save them to the file at path `/tmp/vptest_licenses.json`:

```
vpptest dumplicenses -host myhost.example.com -port 9000 -output /  
tmp/vptest_licenses.json
```

To command a server running at `myhost.example.com` and listening on port 9000 to output its current user records and save them to the file at path `/tmp/vptest_user.json`:

```
vpptest dumpusers -host myhost.example.com -port 9000 -output /  
tmp/vptest_users.json
```

To command a server running on `localhost` and listening on port 9000 to revert to the originally specified configuration:

```
vpptest reset -port 9000
```

Configuration Format

Configuration settings are specified in JSON format.

Configuration Categories

Account and token settings

Account fields such as organization name, secret token, and expiry date can be specified.

VPP Licenses

Dynamically generated or pre-defined VPP licenses can be added to a server on a recurring or one time basis.

User records

Dynamically generated or pre-defined registered users can be changed to an *Associated* state on a recurring or one time basis.

API responses

Responses for each API can be configured. Any sequence of valid responses for each API can be specified, including of course the expected response to an API.

Configuration Keys

"token": <dictionary>

An optional dictionary of key-values that define the SToken

Valid Keys:

"token": <string>

The default value is

"VGhpcyBpcyBhIHhbnXBsZSB0ZXh0IHdoaWN0IHdhcyB1c2VklHRvIGNyZWFOZS
B0aGUgc2ltdWxhdG9yIHJva2VuCg=="

"expDate": <string>

The default value is a year from when the simulator is started

"orgName": <string>

The default value is "Example Inc."

"license_insertions": <array>

An optional array of dictionaries specifying the rules for the creation and insertion of new VPP license or the insertion of pre-defined VPP licenses into the server. The array of dictionaries can be used to insert a list of custom VPP licenses or to dynamically generate a VPP license based on the adamId of the app or book. Additional keys allow for insertion of the records to occur on a repeating interval up to an optional iteration count.

Valid Dictionary Keys:

"license": <dictionary>

An mandatory dictionary of VPP item records

"licenses_per_event": <number>

An optional positive integer specifying the number of VPP licenses to add per iteration

"interval_in_seconds": <number>

An optional non-negative integer specifying the number of seconds to wait between each operation of adding licenses to the VPP account

"initial_delay_in_seconds": <number>

An optional non-negative integer specifying the number of seconds to wait before adding the first set of VPP licenses to the server

"max_event_count": <number>

An optional positive integer specifying the maximum number of VPP licenses

"associate_users": <dictionary>

An optional dictionary specifying the rules for associating user records from the server. The dictionary can be used to cause one of two user association behaviors. The first behavior is to associate a list of pre-defined user records specified by an array of clientUserIdStr's. The second behavior is to randomly associate user records. Additional keys allow for the association of records to occur on a repeating interval up to an optional iteration count.

Valid Keys:

"user_list": <array of dictionaries>

An optional array of user records

"users_per_event": <number>

An optional positive integer specifying the number of users to be marked as associated per iteration

"interval_in_seconds": <number>

An optional non-negative integer specifying the number of seconds to wait between each operation of marking users as associated

"initial_delay_in_seconds": <number>

An optional non-negative integer specifying the number of seconds to wait before changing user records state

"max_event_count": <number>

An optional positive integer specifying the maximum number of users

"registerVPPUserSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the registerVPPUserSrv API.

"getVPPUserSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the getVPPUserSrv API.

"getVPPUsersSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the getVPPUsersSrv API.

"getVPPLicensesSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the getVPPLicensesSrv API.

"getVPPAssetsSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the getVPPAssetsSrv API.

"retireVPPUserSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the retireVPPUserSrv API.

"associateVPPLicenseSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the associateVPPLicenseSrv API.

"disassociateVPPLicenseSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the disassociateVPPLicenseSrv API.

"editVPPUserSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the editVPPUserSrv API.

"manageVPPLicensesByAdamIdSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the manageVPPLicensesByAdamIdSrv API.

"VPPClientConfigSrv_responses": <array>

An optional array of response key records specifying the rule to return one of the valid responses for the VPPClientConfigSrv API.

VPP Item Record Keys

"adamId" : <number>

AdamId of the VPP app or book (required)

"productName" : <string>

Name of the app or book (optional)

"productTypeId" : <number>

Type can be either 8 for applications or 10 for publications (optional)

"isIrrevocable" : <boolean>

Flag to specify if the license can be revoked. Defaults to false (optional)

"pricingParam" : <string>

Pricing param can be either STDQ for standard quality or PLUS for high quality. Defaults to STDQ (optional)

"deviceAssignable" : <boolean>

Flag to specify if the license can be assigned to a device directly. Defaults to false (optional)

User Record Keys

"clientIdStr": <array>

The list of user record clientIdStr's

RESPONSE RECORD KEYS

"count": <number>

An optional positive number to specify the number of times the response should be returned. Defaults to 1

"response": <string>

A string of the supported valid response for the API

"retry_after": <string>

An optional string for the service_unavailable response for all the API's. The string value for this field can be either an HTTP-date or a positive integer number of seconds (in decimal) specifying the retry_after time returned in the response header. If no value is provided then a randomly generated retry_after value will be returned in the service_unavailable response.

Valid Responses Per API

"registerVPPUserSrv_responses"

service_unavailable

login_required

internal_error

url_has_moved

token_has_expired

- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- clientuseridstr_required
- appleid_cant_be_associated

"getVPPUserSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- result_not_found
- userid_or_clientuseridstr_required
- invalid_argument

"getVPPUsersSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- invalid_argument
- result_not_found
- invalid_date_format

"getVPPLicensesSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- invalid_argument
- result_not_found

invalid_date_format

"getVPPAssetsSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- invalid_argument
- result_not_found

"retireVPPUserSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- userid_or_clientuseridstr_required
- invalid_argument
- result_not_found
- registered_user_not_found
- admin_user_not_found
- failed_creating_unclaim_job
- user_already_retired

"associateVPPLicenseSrv_responses"

- service_unavailable
- login_required
- internal_error
- url_has_moved
- token_has_expired
- invalid_authentication_token
- token_revoked
- empty_shared_data_response
- userid_or_clientuseridstr_required
- licenseid_or_adamid_required
- invalid_argument
- result_not_found

account_storefront_mismatch
registered_user_not_found
license_not_found
admin_user_not_found
failed_creating_claim_job
license_already_assigned
user_already_retired
user_already_deleted
license_refunded
license_already_assigned_to_a_different_user

"disassociateVPPLicenseSrv_responses"

service_unavailable
login_required
internal_error
url_has_moved
token_has_expired
invalid_authentication_token
token_revoked
empty_shared_data_response
missing_licenseid_argument
invalid_argument
result_not_found
license_is_irrevocable
registered_user_not_found
license_not_found
admin_user_not_found
failed_creating_unclaim_job
license_not_associated

"editVPPUserSrv_responses"

service_unavailable
login_required
internal_error
url_has_moved
token_has_expired
invalid_authentication_token
token_revoked
empty_shared_data_response
userid_or_clientuseridstr_required
invalid_argument
nothing_to_change
registered_user_not_found
failed_creating_claim_job

failed_creating_unclaim_job
appleid_cant_be_associated
no_registered_user_found
appleid_already_associated

"manageVPPLicensesByAdamIdSrv_responses"

service_unavailable
login_required
internal_error
url_has_moved
token_has_expired
invalid_authentication_token
token_revoked
empty_shared_data_response
adamidstr_required
adamidstr_invalid
pricing_param_invalid
association_disassociation_array_required
one_associate_array_required
one_disassociate_array_required
associate_size_exceeded
disassociate_size_exceeded
no_license_to_associate
not_enough_licenses_to_associate
no_license_to_disassociate
internal_error
license_is_irrevocable
registered_user_not_found
license_not_found
license_already_assigned
user_already_retired
license_not_associated
user_already_deleted
frequent_reassociation
license_not_elligible

"VPPClientConfigSrv_responses"

service_unavailable
login_required
internal_error
url_has_moved
token_has_expired
invalid_authentication_token
token_revoked

empty_shared_data_response
clientcontext_parameter_exceeded_max_width
invalid_apn_token
org_country_not_found

Configuration Examples

Example One:

This configuration specifies the following:

- Alternate token keys values for expDate, orgName, token
- A license insertion rule to dynamically create and add 5 new license records of the podcast app every 2 seconds with an initial delay of 5 seconds and perform this operation 10 times.
- A custom response configuration to the registerVPPUserSrv service request which returns a 503 Service Unavailable HTTP error with a Retry-After header of 20 seconds for the first call and any subsequent requests within the 20 seconds will result in a 503 Service Unavailable error with a Retry-After value less than 20 seconds (this value can be an integer string or a date string). After 20 seconds, the server returns '9601, Login required' error for the first two calls to the API, then returns the expected response once, and then returns '9600, \"userId\" or \"clientIdStr\" is required argument.' error for the next three calls before returning expected responses for the subsequent calls.

```
{
  "stoken": {
    "expDate": "2015-10-21T15:11:11-07:00",
    "orgName": "VPPSim",
    "token": "VPPSimulatorToken"
  },
  "license_insertions": [
    {
      "initial_delay_in_seconds": 5,
      "interval_in_seconds": 2,
      "license": {
        "adamId": 525463029
      },
      "licenses_per_event": 5,
      "max_event_count": 10
    }
  ],
}
```

```

"registerVPPUserSrv_responses": [
  {
    "count": 1,
    "response": "service_unavailable",
    "retry_after": "20"
  },
  {
    "count": 2,
    "response": "login_required"
  },
  {
    "count": 1,
    "response": "expected_response"
  },
  {
    "count": 3,
    "response": "clientuseridstr_required"
  }
]
}

```

Example Two:

This configuration specifies the following:

- A custom response to the associateVPPLicenseWithVPPUserSrv service request to send a '9617, URL has moved' error for the first call, '9608, Empty response from SharedData service' response for the second request, '9604, Result not found' for the third request, '9610, License not found' for the fourth request, '9616, License already assigned' with a predefined license for the fifth request and '9616, License already assigned' with a predefined user for the sixth request. It then returns the expected response for the subsequent calls.
- A user association rule to change the status of a user from the user_list to 'associated' every 5 seconds and to do this operation 2 times in total. The users in the user_list are specified using the clientIdStr used to register the user.
- A license insertion rule to dynamically create and add 10 new license records of a custom app with adamId, pricingParam and productTypeName specified every 2 seconds and perform this operation 10 times.
- A custom response to the retireVPPUserSrv service request to send a '9618, The user has already been retired.' error for the first two calls, then returns the expected response for the next two calls,

then a '9609, Registered user not found.' for the next request and then the expected response for subsequent calls.

```
{
  "associateVPPLicenseSrv_responses": [
    {
      "count": 1,
      "response": "url_has_moved"
    },
    {
      "count": 1,
      "response": "empty_shared_data_response"
    },
    {
      "count": 1,
      "response": "result_not_found"
    },
    {
      "count": 1,
      "response": "license_not_found"
    },
    {
      "count": 1,
      "response": "license_already_assigned",
      "license": {
        "licenseId": 5,
        "adamId": 5,
        "productTypeId": 5,
        "pricingParam": "STDQ",
        "productTypeName": "Software",
        "isIrrevocable": false,
        "userId": 100,
        "clientUserIdStr": "ABC-XYZ",
        "itsIdHash": "ASNWHD12DMDJEAPQM3N35AMDK",
        "status": "Registered"
      }
    }
  ]
}
```



```

    }
  },
  {
    "count": 1,
    "response": "license_already_assigned",
    "users": [
      {
        "clientUserIdStr": "user1",
        "email": "user1@apple.com",
        "itsIdHash": "ASNWHHD12DMDJEAPQM3N35AMDK",
        "status": "Associated",
        "userId": 100261
      }
    ]
  }
],
"associate_users": {
  "interval_in_seconds": 5,
  "max_event_count": 2,
  "user_list": [
    {
      "clientUserIdStr": "abcd"
    },
    {
      "clientUserIdStr": "xyz"
    }
  ],
  "users_per_event": 1
},
"license_insertions": [
  {
    "interval_in_seconds": 2,
    "license": {
      "adamId": 123445,
      "pricingParam": "PLUS",

```

```

        "productTypeName": "Custom B2B"
    },
    "licenses_per_event": 10,
    "max_event_count": 10
}
],
"retireVPPUserSrv_responses": [
    {
        "count": 2,
        "response": "user_already_retired"
    },
    {
        "count": 2,
        "response": "expected_response"
    },
    {
        "count": 1,
        "response": "registered_user_not_found"
    }
]
}

```

Example Three:

This configuration specifies settings for all supported keys and illustrates the complete structure of the configuration format. Note that not all settings specified in this example are intended to be used together. Refer to the documentation for each setting for details.

```

{
    "token": {
        "expDate": "2015-10-21T15:11:11-07:00",
        "orgName": "VPPSim",
        "token": "VPPSimulatorToken"
    },
    "registerVPPUserSrv_responses": [
        {

```

```

    "count": 1,
    "response": "service_unavailable",
    "retry_after": "20"
  },
  {
    "count": 1,
    "response": "login_required"
  },
  {
    "count": 1,
    "response": "internal_error"
  },
  {
    "count": 1,
    "response": "url_has_moved"
  },
  {
    "count": 1,
    "response": "token_has_expired"
  },
  {
    "count": 1,
    "response": "invalid_authentication_token"
  },
  {
    "count": 1,
    "response": "token_revoked"
  },
  {
    "count": 1,
    "response": "empty_shared_data_response"
  },
  {
    "count": 1,
    "response": "clientuseridstr_required"
  }

```

```

    },
    {
        "count": 1,
        "response": "expected_response"
    },
    {
        "count": 1,
        "response": "appleid_cant_be_associated"
    }
],
"getVPPAssetsSrv_responses":[
    {
        "count": 1,
        "response": "service_unavailable",
        "retry_after": "2015-09-31T15:59:29-07:00"
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    }
]

```

```

    },
    {
        "count": 1,
        "response": "token_revoked"
    },
    {
        "count": 1,
        "response": "empty_shared_data_response"
    },
    {
        "count": 1,
        "response": "invalid_argument"
    },
    {
        "count": 1,
        "response": "result_not_found"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"getVPPUserSrv_responses":[
    {
        "count": 1,
        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    },

```

```

{
    "count": 1,
    "response": "url_has_moved"
},
{
    "count": 1,
    "response": "token_has_expired"
},
{
    "count": 1,
    "response": "invalid_authentication_token"
},
{
    "count": 1,
    "response": "token_revoked"
},
{
    "count": 1,
    "response": "empty_shared_data_response"
},
{
    "count": 1,
    "response": "result_not_found"
},
{
    "count": 1,
    "response": "userid_or_clientuseridstr_required"
},
{
    "count": 1,
    "response": "invalid_argument"
},
{
    "count": 1,
    "response": "expected_response"
}

```

```

    }
  ],
  "getVPPUsersSrv_responses": [
    {
      "count": 1,
      "response": "service_unavailable",
    },
    {
      "count": 1,
      "response": "login_required"
    },
    {
      "count": 1,
      "response": "internal_error"
    },
    {
      "count": 1,
      "response": "url_has_moved"
    },
    {
      "count": 1,
      "response": "token_has_expired"
    },
    {
      "count": 1,
      "response": "invalid_authentication_token"
    },
    {
      "count": 1,
      "response": "token_revoked"
    },
    {
      "count": 1,
      "response": "empty_shared_data_response"
    },
  ],

```

```

        {
            "count": 1,
            "response": "result_not_found"
        },
        {
            "count": 1,
            "response": "invalid_date_format"
        },
        {
            "count": 1,
            "response": "invalid_argument"
        },
        {
            "count": 1,
            "response": "expected_response"
        }
    ],
    "getVPPLlicensesSrv_responses": [
        {
            "count": 1,
            "response": "service_unavailable",
        },
        {
            "count": 1,
            "response": "login_required"
        },
        {
            "count": 1,
            "response": "internal_error"
        },
        {
            "count": 1,
            "response": "url_has_moved"
        },
        {

```



```

        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    },
    {
        "count": 1,
        "response": "token_revoked"
    },
    {
        "count": 1,
        "response": "empty_shared_data_response"
    },
    {
        "count": 1,
        "response": "result_not_found"
    },
    {
        "count": 1,
        "response": "invalid_date_format"
    },
    {
        "count": 1,
        "response": "invalid_argument"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"retireVPPUserSrv_responses": [
    {
        "count": 1,

```

```

        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    },
    {
        "count": 1,
        "response": "token_revoked"
    },
    {
        "count": 1,
        "response": "empty_shared_data_response"
    },
    {
        "count": 1,
        "response": "userid_or_clientuseridstr_required"
    },
    {

```

```

        "count": 1,
        "response": "invalid_argument"
    },
    {
        "count": 1,
        "response": "result_not_found"
    },
    {
        "count": 1,
        "response": "registered_user_not_found"
    },
    {
        "count": 1,
        "response": "admin_user_not_found"
    },
    {
        "count": 1,
        "response": "failed_creating_unclaim_job"
    },
    {
        "count": 1,
        "response": "user_already_retired"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"associateVPPLicenseSrv_responses": [
    {
        "count": 1,
        "response": "service_unavailable",
    },
    {
        "count": 1,

```

```

        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    },
    {
        "count": 1,
        "response": "token_revoked"
    },
    {
        "count": 1,
        "response": "empty_shared_data_response"
    },
    {
        "count": 1,
        "response": "userid_or_clientuseridstr_required"
    },
    {
        "count": 1,
        "response": "licenseid_or_adamid_required"
    },
    {

```

```

        "count": 1,
        "response": "invalid_argument"
    },
    {
        "count": 1,
        "response": "result_not_found"
    },
    {
        "count": 1,
        "response": "account_storefront_mismatch"
    },
    {
        "count": 1,
        "response": "registered_user_not_found"
    },
    {
        "count": 1,
        "response": "license_not_found"
    },
    {
        "count": 1,
        "response": "admin_user_not_found"
    },
    {
        "count": 1,
        "response": "failed_creating_claim_job"
    },
    {
        "count": 1,
        "response": "license_already_assigned",
        "license": {
            "licenseId": 5,
            "adamId": 5,
            "productTypeId": 5,
            "pricingParam": "STDQ",

```

```

        "productTypeName": "Software",
        "isIrrevocable": false,
        "userId": 100,
        "clientUserIdStr": "ABC-XYZ",
        "itsIdHash": "ASNWH12DMDJEAPQM3N35AMDK",
        "status": "Registered"
    }
},
{
    "count": 1,
    "response": "license_already_assigned",
    "users": [
        {
            "clientUserIdStr": "user1",
            "email": "user1@apple.com",
            "itsIdHash": "ASNWH12DMDJEAPQM3N35AMDK",
            "status": "Associated",
            "userId": 100261
        }
    ]
},
{
    "count": 1,
    "response": "user_already_retired"
},
{
    "count": 1,
    "response": "user_already_deleted"
},
{
    "count": 1,
    "response": "license_refunded"
},
{
    "count": 1,

```

```

        "response": "license_already_assigned_to_a_different_user"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"disassociateVPPLicenseSrv_responses": [
    {
        "count": 1,
        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    },
    {
        "count": 1,
        "response": "token_revoked"
    }
]

```

```

},
{
    "count": 1,
    "response": "empty_shared_data_response"
},
{
    "count": 1,
    "response": "missing_licenseid_argument"
},
{
    "count": 1,
    "response": "invalid_argument"
},
{
    "count": 1,
    "response": "result_not_found"
},
{
    "count": 1,
    "response": "license_is_irrevocable"
},
{
    "count": 1,
    "response": "registered_user_not_found"
},
{
    "count": 1,
    "response": "license_not_found"
},
{
    "count": 1,
    "response": "admin_user_not_found"
},
{
    "count": 1,

```



```

        "response": "failed_creating_unclaim_job"
    },
    {
        "count": 1,
        "response": "license_not_associated"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"editVPPUserSrv_responses": [
    {
        "count": 1,
        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    }
]

```

```

},
{
    "count": 1,
    "response": "token_revoked"
},
{
    "count": 1,
    "response": "empty_shared_data_response"
},
{
    "count": 1,
    "response": "userid_or_clientuseridstr_required"
},
{
    "count": 1,
    "response": "invalid_argument"
},
{
    "count": 1,
    "response": "nothing_to_change"
},
{
    "count": 1,
    "response": "registered_user_not_found"
},
{
    "count": 1,
    "response": "failed_creating_claim_job"
},
{
    "count": 1,
    "response": "failed_creating_unclaim_job"
},
{
    "count": 1,

```

```

        "response": "appleid_cant_be_associated"
    },
    {
        "count": 1,
        "response": "no_registered_user_found"
    },
    {
        "count": 1,
        "response": "appleid_already_associated"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"manageVPPLlicensesByAdamIdSrv_responses": [
    {
        "count": 1,
        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    }
]

```

```

},
{
    "count": 1,
    "response": "invalid_authentication_token"
},
{
    "count": 1,
    "response": "token_revoked"
},
{
    "count": 1,
    "response": "empty_shared_data_response"
},
{
    "count": 1,
    "response": "adamidstr_required"
},
{
    "count": 1,
    "response": "adamidstr_invalid"
},
{
    "count": 1,
    "response": "pricing_param_invalid"
},
{
    "count": 1,
    "response": "association_disassociation_array_required"
},
{
    "count": 1,
    "response": "one_associate_array_required"
},
{
    "count": 1,

```

```

        "response": "one_disassociate_array_required"
    },
    {
        "count": 1,
        "response": "associate_size_exceeded"
    },
    {
        "count": 1,
        "response": "disassociate_size_exceeded"
    },
    {
        "count": 1,
        "response": "no_license_to_associate"
    },
    {
        "count": 1,
        "response": "not_enough_licenses_to_associate"
    },
    {
        "count": 1,
        "response": "no_license_to_disassociate"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "license_is_irrevocable"
    },
    {
        "count": 1,
        "response": "registered_user_not_found"
    },
    {

```

```

        "count": 1,
        "response": "license_not_found"
    },
    {
        "count": 1,
        "response": "license_already_assigned"
    },
    {
        "count": 1,
        "response": "user_already_retired"
    },
    {
        "count": 1,
        "response": "license_not_associated"
    },
    {
        "count": 1,
        "response": "user_already_deleted"
    },
    {
        "count": 1,
        "response": "frequent_reassociation"
    },
    {
        "count": 1,
        "response": "license_not_elligible"
    },
    {
        "count": 1,
        "response": "expected_response"
    }
],
"VPPClientConfigSrv_responses": [
    {
        "count": 1,

```

```

        "response": "service_unavailable",
    },
    {
        "count": 1,
        "response": "login_required"
    },
    {
        "count": 1,
        "response": "internal_error"
    },
    {
        "count": 1,
        "response": "url_has_moved"
    },
    {
        "count": 1,
        "response": "token_has_expired"
    },
    {
        "count": 1,
        "response": "invalid_authentication_token"
    },
    {
        "count": 1,
        "response": "token_revoked"
    },
    {
        "count": 1,
        "response": "empty_shared_data_response"
    },
    {
        "count": 1,
        "response": "clientcontext_parameter_exceeded_max_width"
    },
    {

```

```
        "count": 1,  
        "response": "invalid_apn_token"  
    },  
    {  
        "count": 1,  
        "response": "org_country_not_found"  
    },  
    {  
        "count": 1,  
        "response": "expected_response"  
    }  
]  
}
```