TSSRegistry

SPECIFICATIONS

Texting & Smart Services (TSS) Registry User API Version 2.0

June 7, 2019

Version 9

somos.com
o 844.HEY.SOMOS
f 732.514.6723
P.O. Box 8122
Bridgewater, NJ 08807-8122

Revision History

Version History		
Date	***Version***	***Description***
1/3/2017	1	Initial version of API 2.0 in TSS 1.14
04/03/2017	2	New field 'rejectCode' added to /api/v2/tss-text-set-pending-tfn API
11/16/2017	3	New API "tss-change-password" added
01/12/2018	4	New API '/api/v2/tss-bulk-tfn-query' added. Moved '/api/v2/tss-text-query-tfn' to Common API. New fields 'ownerSrName' and 'routingSrName" added to '/api/v2/tss-text-query-tfn' API. Field 'ror' removed from '/api/v2/tss-text-query-tfn' API.
07/18/2018	5	New API '/api/v2/tss-voice-ro-inventory-tfn' added under Section 4.4 Section 3.3 renamed to "Text Enabled TFN Inventory" Section 4.3 renamed to "Text Enabled TFN Inventory"
08/13/2018	6	Correct the sample responses to add "/api/v2" to the "url" field
10/29/2018	7	New Port_Pending staus, New reject code/reason
01/22/2019	8	New API "/api/v2/tss-text-history" added
06/07/2019	9	Updated reject codes and changes to description of Pending and Port_Pending statuses

Table of Contents

1. Audience

The audience for this document is development and quality assurance teams. There are concepts in this document that assume the reader already understands protocols, such as HTTP, REST, and JSON. The document also assumes that the reader is already familiar with TSS, SMS/800, and text enabling Toll-Free Numbers.

2. Introduction

This document describes the Application Programming Interface (API) between TSS and its users, such as Service Registrars and Resp Orgs (collectively referred to as TSS *User Clients)*. In general, TSS provides this API to allow users to take actions, via a protocol, that otherwise would only be available via the TSS web-based interface.

The TSS User API version 2.0 contains performance enhancements to support more throughput and improved response times. The TSS User API version 1.0 will be supported for a period of time, but version 2.0 is the recommended way of interacting with TSS.

The following diagram shows the relevant network elements for the TSS User API. The arrows highlighted in blue are the subjects of this document.

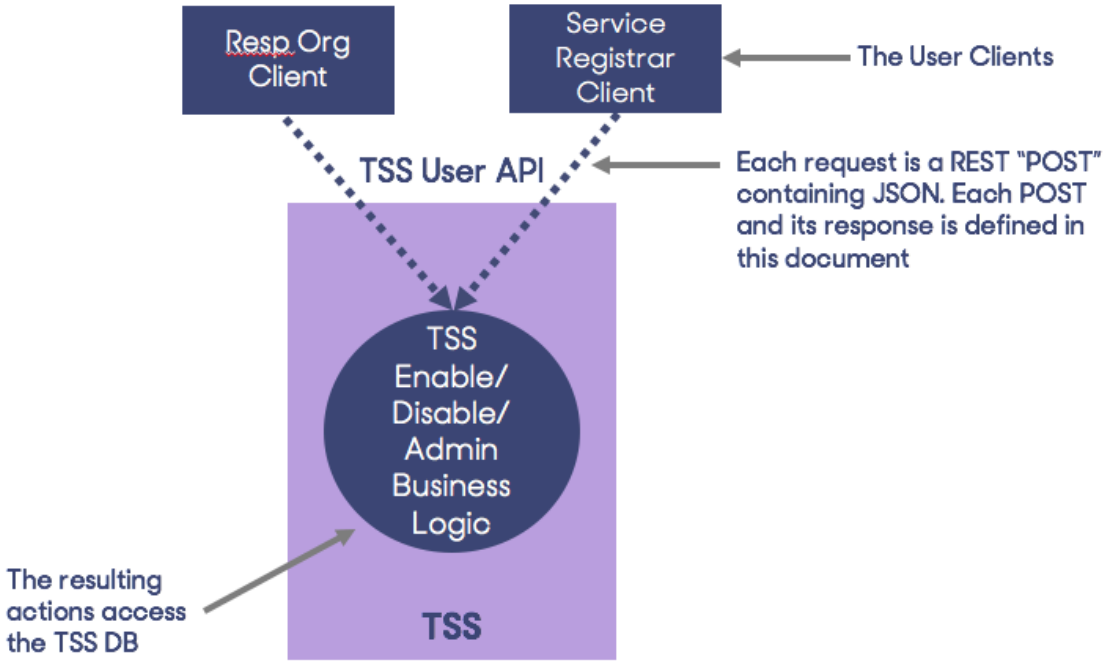

Figure 1: TSS User API

The User Client sends a REST request to https://api-texting.somos.com to be processed by the TSS Business Logic. The POST encapsulates JSON, accesses the TSS Database as necessary, and returns a response.

Note that HTTPS is used to encrypt the connection, and the user ID and password are required in the JSON messages for authentication.

The following chapters separate the API discussion into:

- Service Registrar API,
- Resp Org API,
- and Common API - relevant to both the Service Registrar and the Resp Org

3. Service Registrar API

The TSS REST API provides an easy-to-use set of HTTP endpoints that let you access data in simple JSON format. The Service Registrar makes a HTTP POST request containing JSON payload to the TSS Server. The TSS Server responds to the POSTs with JSON, after processing the request. Standard HTTP status codes are replied to the users of TSS API services using JSON over HTTP.

The HTTP requests are identified by its endpoints (URL). The following table enumerates the possible HTTP endpoints.

API URLs			
HTTP Method	URL Type	URL	Comments
POST	update	/api/v2/tss-text-update-tfn	Process a set of enable and disable requests
POST	query-tspid	/api/v2/tss-text-query-tspid	Retrieve contact information associated with a tspid.
POST	inventory	/api/v2/tss-text-sr-inventory-tfn	Retrieve a list of text enabled and pending TFNs for a TSPID.
POST	query-tfn	/api/v2/tss-text-query-tfn	Refer to the Common API chapter
POST	bulk-query-tfn	/api/v2/tss-bulk-tfn-query	Refer to the Common API chapter.
POST	change-password	/api/v2/tss-change-password	Refer to the Common API chapter.
POST	error	/api/v2/tss-user-api-error	Refer to the Common API chapter

The error URL is discussed later in this document since it is a common URL.

3.1 Update TFNs

The Update API processes a set of commands for enabling and disabling TFNs. This is a **bulk** API. Currently up to 1,000 commands in an array are allowed.
The following is an example REST/JSON update request from the user:

Note that enable requests require the user to provide all the fields; however, the disable requests only require the command and tfn fields.

```
POST /api/v2/tss-text-update-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "requests":[                    // an array of requests to process
   {                             // the start of the first request
     "command":"enable",
     "tfn":"8005001212",
     "routingTspid":"12345",
     "ownerTspid":"12345",
     "businessName":"Subscriber Business Name",
     "contactName":"Subscriber Name",
     "contactJobTitle":"Subscriber Job Title",
     "contactPhone":"Subscriber Phone",
     "contactEmail":"Subscriber Email"
   },                            // the end of the first request
   {                             // the start of the second request
     "command":"disable",
     "tfn":"8005001213"
   },                            // the end of the second request
   {                             // the start of the third request
     "command":"enable",
     "tfn":"8005001214",
     "routingTspid":"01234",
     "ownerTspid":"12345",
     "businessName":"Subscriber Business Name",
     "contactName":"Subscriber Name",
     "contactJobTitle":"Subscriber Job Title",
     "contactPhone":"Subscriber Phone",
     "contactEmail":"Subscriber Email"
   }                             // the end of the third request
   ]                             // the end of the request array
}
```

Update TFN Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-update-tfn	The URL	A request to process a set of requests.
id	A User ID already created in TSS.	The id is created by TSS administration.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset any time via the Web Site.
command	"enable" or "disable"	The requested action for the TSS to perform.
tfn	Exactly 10 printable digits.	At the time of this writing, the TFN is always 10 numeric characters. A poorly formatted TFN will not match a value in the voice DB and will be rejected. Only TFNs in working status in the SMS/800 voice registry can be enabled by TSS. And even those TFNs may be rejected for other reasons (e.g. lack of user permissions).
routingTspid	A valid TSPID in the TSS DB.	The TSPID used for text routing. It identifies a Service Registrar. TSS assigns a unique TSPID to each Service Registrar. The user must have permissions to perform this request on behalf of the TSPID or an error results.
ownerTspid	A valid TSPID in the TSS DB.	Typically the same as the routingTspid field, but it can be populated with a different TSPID to identify a Service Registrar that is responsible for the TFN; for example, billing responsibilities. The user must have permissions to perform this request on behalf of the TSPID or an error results.
businessName	1..256 characters	The toll-free subscriber's business name using the TFN.
contactName	1..256 characters	The name of the toll-free subscriber responsible for the enable request.
contactJobTitle	1..256 characters	The job title of the toll-free subscriber responsible for the enable request.
contactPhone	exactly 10 digits	The phone of the toll-free subscriber responsible for the enable request.
contactEmail	1..256 characters	The email of the toll-free subscriber responsible for the enable request.

The TFNs in the request array are processed independent of each other; the TSS Registry version 2.0 API supports **partial success** of the requests array.

A response similar to the following is issued by the TSS to the user if it was a valid request. Since it's a bulk API, the response includes a count of the TFNs in the request array, a success response, and a failure

response containing the error messages. The response array may not have the status in the same order as the original request array. The failed TFNs can be resubmitted later.

The success response includes a count of the successful TFN updates and status(es) of the TFNs. The failure response includes a count of the failed TFN updates and error message(s). Enable requests have the following fields in the response: tfn and status. Disable requests have the following fields in the response: tfn and errorMessage.

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
 "url":"/api/v2/tss-text-update-tfn", // positive confirmation for
                                      //debugging
 "totalTfnCountInRequest":3,   // total number of tfns in the request
 "successResponse":{     // success response
       "successCount":2, // total number of successful tfn updates
       "responses":[   // an array of success responses to the requests
         {"tfn":"8005551212","status":"<SeeStatusTableBelow>"},
         // first success response
         {"tfn":"8005551213","status":"<SeeStatusTableBelow>"}
         // second success response
        ]               // the end of the success responses array
}                       // the end of success response
"failureResponse":{     // failed response
      "failureCount":1, // total number of failed tfn updates
      "responses":[     // an array of failed responses to the requests
      {"tfn":"8005551214","errorMessage":"<SeeErrorMessageTableBelow>"}
         // first error response
      ]                 // the end of the failed responses array
}                       // the end of failure response
}
```

The following table describes the possible values for the status field.

Status Field Values			
Status	***Comments***	***Applies to enable***	***Applies to disable***
"Enabled"	The TFN is enabled for text.	yes	no
"Pending"	The TFN has been enabled by a Service Registrar and is pending validation by the Resp Org.	yes	no
"Port_Pending"	The TFN is already enabled by a different Service Registrar and another Service Registrar enables it and is pending validation by the Resp Org.	yes	no
"Disabled"	The TFN is disabled for text.	no	yes

The following table describes the possible values for the error message field.

Error Message Field Values			
ErrorMessage	***Comments***	***Applies to enable***	***Applies to disable***
"The command field (<CommandName>) is invalid."	The command specified in the request is invalid.	yes	yes
"The owner tspid (<OwnerTSPID>) must be a valid numeric tspid."	The owner TSPID specified in the request is invalid.	yes	no
"The routing tspid (<RoutingTSPID>) must be a valid numeric tspid."	The routing TSPID specified in the request is invalid.	yes	no
"The TFN (<TFN>) is missing or invalid. It must be specified as 10 printable digits."	The TFN field is missing or invalid.	yes	yes
"The user does not have permission to enable TFNs."	The user does not have permission to enable the TFN in the request.	yes	no
"The TFN (<TFN>) is not an active Toll-Free Number."	The TFN specified in the request is not a working TFN.	yes	no
"The user must have permission to activate on behalf of the tspid (<TSPID>)."	The user does not have permission to activate the TFN on behalf of TSPID.	yes	no
"The user must have permission to deactivate on behalf of the tspid (<TSPID>)."	The user does not have permission to deactivate the TFN on behalf of TSPID.	no	yes
"The TFN (<TFN>) is not an active Toll-Free Number."	The TFN is Unavailable i.e. not in the voice registry, therefore it cannot be text enabled.	yes	no

A redundant enable request (for a TFN already enabled by the requesting SR) is allowed and results in replacing the enable parameters for the TFN with a returned response of "Enabled". Similarly, a redundant disable request is allowed (for a TFN already disabled) -- the redundant disable request appears to the TSS to be an attempt to disable a number that is not currently enabled -- in the initial disabled event and any subsequent redundant disabled events, the response is "Disabled".

The tables above enumerate the possible results of enabled and disabled requests as well as the error messages as a result of failed TFN enable/disable requests. If an error occurs that prevents the TSS from completing the request (such as authentication failure or bad request), then the above response is not returned and instead a failure response, as discussed in the *Error* section, is returned.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
400	Bad request – invalid JSON format.
401	Authentication failed.
200	The request has succeeded (some TFN updates in the request array may not have been successful, the errorMessage field contains the reason for the failure).

3.2 Query TSPID

The following is an example REST/JSON query TSPID request from the user:

```
POST /api/v2/tss-text-query-tspid HTTP/1.0

Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "tspid":"12345"
}
```

Query TSPID Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-query-tspid	the URL	A request to retrieve information for the given TSPID.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
tspid	A valid TSPID in the TSS DB.	TSS assigns a unique TSPID to each Service Registrar.

A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url":"/api/v2/tss-text-query-tspid", // positive confirmation
   "tspid":"12345",                 // positive confirmation
   "srName":"ACME Service Reg",     // name of the Service Registrar
   "contactName":"Jane Doe",              // primary contact
   "contactEmail":"contact@AcmeSR.com" // primary contact
}
```

See the *Error* section for failure responses, such as a tspid that is not present.

The following table describes the fields in the response.

Query TSPID Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tspid	The TSPID provided in the original query to make debugging easier.
srName	The name of the Service Registrar associated with the TSPID.
contactName	The name of the primary contact for the Service Registrar.
contactEmail	The email of the primary contact for the Service Registrar.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
500	An error occurred within the TSS Registry.

3.3 Text Enabled TFN Inventory

The following is an example REST/JSON text enabled inventory request from the user:

```
POST /api/v2/tss-text-sr-inventory-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
  "id":"user",
  "pwd":"password",
  "tspid":"12345",        // looks for owner and routing TSPID
  "partialTfn":"8"        // a single "8" means all 8xxXxxXxxx TFNs
}
```

This request asks the TSS to return TFNs where the given tspid is in use as an owner *or* a routing TSPID. The user can provide more digits in the partialTfn to narrow the request -- such as "800555" would provide all 800555xxxx values that match the owner or the routing TSPID.

TFN Inventory Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-sr-inventory-tfn	the URL	A request to retrieve a list of text enabled and pending TFNs for a particular owner/routing TSPID.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
tspid	A valid Owner or Routing TSPID in the TSS DB.	Identifies an owner or routing Service Registrar that is associated with a set of TFNs. The user must have permissions to perform this request on behalf of the tspid or an error results.
partialTfn	printable char, 1..10 chars, always starting with "8" with up to 10 total. All chars are '0'..'9'.	Indicates the scope of the inventory request. A single "8" means all 8xx numbers are requested (8xxxxxxxxx) for the given tspid. An example of "800555" means that any TFN starting with "800555" is being requested (800555xxxx), where the given tspid is in use as an owner or a routing TSPID.

If the id does not have permissions for the given tspid, then a failure results. The maximum size in the response is 10,000 TFNs. Therefore, if the partialTfn results in more than the maximum, then an

error response results and the user must narrow the partialTfn in the request by providing more digits in the partialTfn.

The response list is ordered, ascending, by TFN. A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
 "url": "/api/v2/tss-text-sr-inventory-tfn", // positive confirmation
 "responses":[                          // an array of responses
   {                                    // the first response
     "tfn":"8005001212",
     "status":"Pending",                // see status table below
     "routingTspid":"01234",
     "ownerTspid":"12345",
     "ror":"12345",
     "date":"04/21/2015 18:03:56 GMT",    // GMT, the only timezone
     "businessName":"Subscriber Business Name"
   },                                   // the end of the first response
   {                                    // the start of the second response
     "tfn":"8005001213",
     "status":"Enabled",                    // see status table below
     "routingTspid":"12121",
     "ownerTspid":"12345",
     "ror":"ROR12",
     "date":"04/09/2015 07:49:06 GMT",    // GMT, the only timezone
     "businessName":"Subscriber Business Name"
   }                                    // the end of the second response
 ]                                      // the end of the response array
}
```

Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the *status* field on a successful request.

Status Field Values on Successful Request	
Status	***Comments***
"Enabled"	The TFN is enabled for text.
"Pending"	The TFN has been enabled by a Service Registrar and is pending validation by the Resp Org.
"Port_Pending"	The TFN is already enabled by a different Service Registrar and another Service Registrar enables it and is pending validation by the Resp Org.

The following table describes the fields in the response.

Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tfn	The TFN associated with this entry.
status	See the Status table above.
routingTspid	The TSPID used for text routing. It identifies a Service Registrar. TSS assigns a unique TSPID to each Service Registrar.
ownerTspid	Identifies a Service Registrar that is responsible for the TFN.
ror	The Resp Org ID associated with voice registration of the TFN.
date	The timestamp when the TFN was put on the pending list or when it was enabled. It is *always* relative to GMT. It is *always* of the format "MM/DD/YYYY HH:MM:SS GMT" where the fields have leading zeros. Finally, the HH uses a 24 hour clock: 00..23.
businessName	The toll-free subscriber's business name using the TFN.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation, max TFN inventory exceeded).
401	Authentication failed.
403	The user is not associated with a Service Registrar.
500	An error occurred within the TSS Registry.

4. Resp Org API

The API between TSS and the Resp Org is REST encapsulating JSON. The Resp Org sends POSTs containing JSON to the TSS Server. The TSS Server responds to the POSTs with JSON, after processing the request.

The POST is identified by its URL. The following table enumerates the possible POST URLs.

POST URLs			
HTTP Method	***URL Type***	***URL***	***Comments***
POST	get pending	/api/v2/tss-text-get-pending-tfn	Read-only retrieval of TFNs that are pending validation by a Resp Org.
POST	set pending	/api/v2/tss-text-set-pending-tfn	Approve or reject a pending TFN by the Resp Org.
POST	inventory	/api/v2/tss-text-ro-inventory-tfn	Retrieve a list of text enabled and pending TFNs for a RO.
POST	voice-inventory	/api/v2/tss-voice-ro-inventory-tfn	Retrieve a list of voice enabled and text enabled TFNs for a RO.
POST	query-tfn	/api/v2/tss-text-query-tfn	Refer to the Common API chapter
POST	bulk-query-tfn	/api/v2/tss-bulk-tfn-query	Refer to the Common API chapter.
POST	change-password	/api/v2/tss-change-password	Refer to the Common API chapter.
POST	error	/api/v2/tss-user-api-error	Refer to the Common API chapter

The error URL is discussed later in this document since it is a common URL.

4.1 Get Pending TFN

The following is an example REST/JSON get-pending request from the user:

```
POST /api/v2/tss-text-get-pending-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "ror":"RespOrgID"    // issue this post again for each ROR desired
}
```

Get Pending TFN Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-get-ending-tfn	the URL	A request to retrieve information about all TFNs in the Pending state.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
ror	5 alpha-numeric digits	A Resp Org ID associated with the toll free voice service.

A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url": "/api/v2/tss-text-get-pending-tfn", // positive confirmation
   "pendingList":
    [                                    // start of the list
      {                                  // start of 1st entry
       "tfn":"8007001212",
       "date":"04/21/2015 08:03:56 GMT", // GMT, the only timezone
       "sr":"ServiceRegistrarName",              // Owner SR
       "businessName":"Subscriber Business Name",
       "contactName":"Subscriber Contact Name",  // A person
       "contactPhone":"Subscriber Contact Phone",
       "contactEmail":"Subscriber Contact Email",
       "tspid":"12345"                           // Owner SR ID
      }                                   // end of 1st entry
       // The above is a single entry in the pendingList.  More entries
       // may be present, each in a comma-separated curly brace stanza.
   ] // end of the pendingList
}
```

The following table describes the above fields.

Get Pending Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
pendingList	The array of pending TFN entries. The start and end of the array are identified by square brackets ([...]). Each entry in the list is its own curly brace stanza ({...}). The remaining rows below identify the fields in an individual entry.
tfn	The TFN associated with this pending entry (an entry in the list).
date	The timestamp when the TFN was put on the pending list. It is *always* relative to GMT. It is *always* of the format "MM/DD/YYYY HH:MM:SS GMT" where the fields have leading zeros. Finally, the HH uses a 24 hour clock: 00..23.
sr	The Owner Service Registrar name associated with this TFN.
businessName	The toll-free subscriber's business name using the TFN.
contactName	The name of the toll-free subscriber responsible for enabling the TFN.
contactPhone	The phone of the toll-free subscriber responsible for enabling the TFN.
contactEmail	The email of the toll-free subscriber responsible for enabling the TFN.
tspid	The Owner TSPID associated with the Service Registrar.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	The user does not have permission to retrieve pending TFNs.
500	An error occurred within the TSS Registry.

4.2 Set Pending TFN

The following is an example REST/JSON set-pending request from the user in order to approve or reject a pending TFN:

To approve a pending TFN:

```
POST /api/v2/tss-text-set-pending-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "tfn":"8005551212",
 "action":"approve"
}
```

To reject a pending TFN:

```
POST /api/v2/tss-text-set-pending-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>
```

```
{
 "id":"user",
 "pwd":"password",
 "tfn":"8005551212",
 "action":"reject",
 "rejectCode":"<rejectCode>"    // See Reject Reason Codes table
}
```

Set Pending TFN Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-set-pending-tfn	the URL	A request to approve or reject a TFN that is on the Pending list. This is only intended to be used by a Resp Org. Any other user will not have permissions to make this change. A failure response is returned if the id is not associated with the Resp Org that controls the TFN.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
tfn	Exactly 10 printable digits.	Only TFNs that are in the TSS Pending state can be approved or rejected. Also, only the Resp Org that controls the voice aspect of the TFN can approve a Pending TFN. And even those requests could be disallowed for other reasons (e.g. lack of individual user permissions, or the TFN is not in the Pending state).
action	"approve" or "reject"	The action the Resp Org wishes to take for the given TFN.
rejectCode	A valid reject code	The reject reason code. This field is required if action is "reject". If an invalid rejectCode is provided, an error message is returned along with a list of valid reject reason codes.

The following table describes the list of valid reject reason codes.

Reject Reason Codes	
Reject Code	***Description***
01	LOA Data Mismatch
02	Toll-Free Shared or Bundled Service
03	Unauthorized LOA Contact
04	Not Commercially Viable/Integrated Service

A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url": "/api/v2/tss-text-set-pending-tfn", // positive confirmation
   "tfn": "8005551212"                 // positive confirmation
}
```

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	The user does not have permission to approve pending TFN.
500	An error occurred within the TSS Registry.

Any failure is communicated via an error response as discussed in the *Error* section later in this document.

4.3 Text Enabled TFN Inventory

The following is an example REST/JSON text enabled inventory request from the user:

```
POST /api/v2/tss-text-ro-inventory-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "ror":"RespOrgID"     // issue this post again for each ROR desired
 "partialTfn":"8"      // a single "8" means all 8xxXxxXxxx TFNs
}
```

This request asks the TSS to return a list of TFNs where each TFN is either in the Pending or Enabled state for the given ror. The user can provide more digits in the partialTfn to narrow the request -- such as "800555" would provide all enabled and pending TFNs with the pattern "800555xxxx" associated with the ror.

RO Inventory Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-ro-inventory-tfn	the URL	A request to retrieve a list of text enabled and pending TFNs associated with a particular Resp Org ID (ror).
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
ror	5 alpha-numeric digits	A Resp Org ID associated with the toll free voice service.
partialTfn	printable char, 1..10 chars, always starting with "8" with up to 10 total. All chars are '0'..'9'.	Indicates the scope of the inventory request. A single "8" means all 8xx numbers are requested (8xxxxxxxxx) for the given ror. An example of "800555" means that any TFN starting with "800555" is being requested (800555xxxx), where the given ror is associated with it as pending or enabled.

If the id does not have permissions for the given ror, then a failure results. The maximum size in the response is 10,000 TFNs. Therefore, if the partialTfn results in more than the maximum, then an error response results and the user must narrow the partialTfn in the request by providing more digits in the partialTfn.

The response list is ordered, ascending, by TFN. A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
 "url": "/api/v2/tss-text-ro-inventory-tfn",// positive confirmation
 "responses":[                              // an array of responses
   {                                        // the first response
     "tfn":"8005001212",
     "status":"Pending",                    // see status table below
     "ownerTspid":"12345",
     "date":"04/21/2015 18:03:56 GMT", // GMT, the only timezone
     "businessName":"Subscriber Business Name"
   },                                       // the end of the first response
   {                                        // start of the second response
     "tfn":"8005001213",
     "status":"Enabled",                    // see status table below
     "ownerTspid":"12345",
     "date":"04/09/2015 07:49:06 GMT", // GMT, the only timezone
     "businessName":"Subscriber Business Name"
   }                                        // end of the second response
 ]                                          // end of the response array
}
```

Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the *status* field on a successful request.

Status Field Values on Successful Request	
Status	***Comments***
"Enabled"	The TFN is enabled for text.
"Pending"	The TFN has been enabled by a Service Registrar and is pending validation by the Resp Org.
"Port_Pending"	The TFN is already enabled by a different Service Registrar and another Service Registrar enables it and is pending validation by the Resp Org.

The following table describes the fields in the response.

Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tfn	The TFN associated with this pending entry.
status	See the Status table above.
ownerTspid	Identifies a Service Registrar that is responsible for the TFN.
date	The timestamp when the TFN was put on the pending list or when it was enabled. It is *always* relative to GMT. It is *always* of the format "MM/DD/YYYY HH:MM:SS GMT" where the fields have leading zeros. Finally, the HH uses a 24 hour clock: 00..23.
businessName	The toll-free subscriber's business name using the TFN.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	The user is not associated with a Resp Org or does not have permission to query on behalf of ROR.
500	An error occurred within the TSS Registry.

4.4 Voice Enabled TFN Inventory

The following is an example REST/JSON voice enabled inventory request from the user:

```
POST /api/v2/tss-voice-ro-inventory-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "ror":"RespOrgID",     // issue this post again for each ROR desired
 "partialTfn":"8",      // a single "8" means all 8xxXxxXxxx TFNs
 "page":"1",            // page number (optional)
 "sortOrder":"ASC"      // sort order (optional)
}
```

This request asks the TSS to return a list of voice enabled TFNs for the given ror. If any of the TFNs are enabled for text, the response will include details about the text enabled TFNs. The user can provide more digits in the partialTfn to narrow the request (see table below for examples).

RO Inventory Fields		
Field	***Range***	***Comments***
/api/v2/tss-voice-ro-inventory-tfn	the URL	A request to retrieve a list of voice enabled TFNs associated with a particular Resp Org ID (ror).
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
ror	5 alpha-numeric digits	A Resp Org ID associated with the toll free voice service.
partialTfn	printable char, 1..10 chars, always starting with "8" with up to 10 total. All chars are '0'..'9'.	Indicates the scope of the inventory request. A single "8" means all 8xx numbers are requested (8xxxxxxxxx) for the given ror. An example of "800555" means that any TFN starting with "800555" is being requested (800555xxxx), where the given ror is associated with it.
page (optional field)	Numeric value	Indicates the requested page number for data in the response. The response will include the total number of pages matching the request criteria (see table "Response Fields" below).

		Defaults to 1 if not specified. If data for requested page does not exist, an error message is returned.
sortOrder (optional field)	The sort order	The sort order for the request. Valid values are "ASC" or "DESC". Defaults to ASC if not specified.

If the id does not have permissions for the given ror, then a failure results. The maximum size in the response is 10,000 TFNs. If the response includes more than 10,000 TFNs, the remaining set of TFNs can be retrieved by sending another request specifying the page number in the page field.

The response list is ordered by TFN based on the sortOrder field in the request. A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
 "url": "/api/v2/tss-voice-ro-inventory-tfn",// positive confirmation
 "totalRecords":"624953",      // total TFNs matching request criteria
 "totalPages":"63",            // total number of pages
 "currentPage":"1              // current page
 "tfns":[                      // an array of responses containing tfns
   {                                   // the first response
     "tfn":"8005001212",
     "status":"Voice Enabled",         // see status table below
     "ownerTspid":null,
     "date":null,                      // GMT, the only timezone
     "businessName":null
   },                                  // the end of the first response
   {                                   // start of the second response
     "tfn":"8005001213",
     "status":"Text Enabled",          // see status table below
     "ownerTspid":"12345",
     "date":"04/09/2015 07:49:06 GMT", // GMT, the only timezone
     "businessName":"Subscriber Business Name"
   }                                   // end of the second response

 ]                                     // end of the response array
}
```

Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the *status* field on a successful request.

Status Field Values on Successful Request	
Status	***Comments***
"Voice Enabled"	The TFN is voice enabled.
"Text Enabled"	The TFN is enabled for text.

The following table describes the fields in the response.

Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.

tfn	The TFN matching the request criteria.
status	See the Status table above.
ownerTspid	Identifies a Service Registrar that is responsible for the TFN. This field will have a "null" value if the TFN is not "Text Enabled"
date	The timestamp when the TFN was enabled. It is *always* relative to GMT. It is *always* of the format "MM/DD/YYYY HH:MM:SS GMT" where the fields have leading zeros. Finally, the HH uses a 24 hour clock: 00..23. This field will have a "null" value if the TFN is not "Text Enabled"
businessName	The toll-free subscriber's business name using the TFN. . This field will have a "null" value if the TFN is not "Text Enabled"
totalRecords	Total number of TFNS matching request criteria
totalPages	Total number of pages to retrieve all TFNs matching criteria
currentPage	Current page (indicating the page containing the response data)

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	The user is not associated with a Resp Org or does not have permission to query on behalf of ROR.
500	An error occurred within the TSS Registry.

5. Common API

The API between TSS and its User Clients is REST encapsulating JSON. The Client sends POSTs containing JSON to the TSS Server. The TSS Server responds to the POSTs with JSON, after processing the user's request.

The POST is identified by its URL. The following table enumerates the possible POST URLs that are common to both the Service Registrar and the Resp Org.

POST URLs			
HTTP Method	***URL Type***	***URL***	***Comments***
POST	error	/api/v2/tss-user-api-error	To inform TSS of an error situation.
POST	change-password	/api/v2/tss-change-password	To change password.
POST	query-tfn	/api/v2/tss-text-query-tfn	Retrieve the TFN text status, and other information
POST	bulk-query-tfn	/api/v2/tss-bulk-tfn-query	Retrieve the TFN text status, and other information for a list of TFNs.

Each URL in the table above is described in the following sections.

5.1 Error Detection and Error Communication

Error detection and communication happens at TSS and at the User Client. Both types of failures require one side to communicate the failure to the other. They are both discussed within separate sections below.

5.1.1 TSS Error Detection

TSS uses an *error* field in the response to a POST to communicate a JSON content failure or its inability to complete the request for some other reason. A *message* field is also supplied to provide more information in free-form, printable, text (to aid debugging). The following is an example response to a POST indicating an error was encountered by TSS:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/plain;charset=ISO-8859-1
Content-Length: <num>
Date: Tue, 13 Jan 2015 16:36:00 GMT
Connection: close

{
   "url": "<the url used by the POST>",
   "error": "SeeTableBelowForPossibleValues",
   "message": "The user does not have permissions." // for example
}
```

The following table describes the possible values for the *error* field.

Error Field Values

Error Value	*Comments*	*Action*
"temporary"	A temporary error on TSS has occurred. The HTTP connection has been successful to this point, but TSS is unable to complete the POST request at this time. Sending the same request again, but at a later time, is expected to be successful. An example could be a temporary database outage at TSS during maintenance.	The User Client SHOULD retry the request later. It is recommended that the User Client wait 60 seconds before the retry.
"permanent"	A permanent error has occurred that MUST be addressed by the User Client. Resending the request will NOT yield success, so retrying is not reasonable without some additional action first being taken. An example is an incorrect password in the POST. Similarly if the user id lacks permissions to make this change, then the error is permanent and requires the user's permissions to be corrected first.	The User MUST stop sending this POST and inspect the message field to learn more about the problem. Contacting TSS may be required based on the issue. Once the problem has been addressed, the User can resend the POST.

If the User receives a response with an error value from the table above, it MUST take the action described in the table above.

The message field is free-form text, where the contents are otherwise not constrained by this specification. It allows TSS to communicate the issue in printable text to aid debugging and integration. The User Client MUST NOT depend upon any values in the message text, since the text is subject to change completely at the discretion of the sender at any time without notice. Up to 1,000 printable characters (including whitespace such as carriage returns, etc.) are allowed for flexibility. The sender of a message field is encouraged to provide verbose information, as reasonable, to aid the debugging effort which could occur minutes, hours, or days later in a production environment.

It is unnecessary (and would likely be incomplete) to try to enumerate all the possible errors; however, they all fall into one of the categories in the table above ("temporary" or "permanent") and are driven by the action the User Client must take. Below are some examples of errors to aid development. The text in these examples is NOT required in the message field -- it is only for explanatory purposes in this document.

Temporary error examples:

- A database outage has occurred at TSS.
- A maintenance window is in progress at TSS.

Permanent error examples:

- An invalid ID or password was provided by the User Client.
- The user does not have permissions to complete the action.
- The TFN provided was not formatted correctly (e.g. "800pizza").
- The TFN (8000001212) was not in the voice DB therefore could not be enabled.
- A field is missing in the JSON portion of the POST, such as the id, pwd, tfn, etc.
- An unknown field is in the JSON portion of the POST.

The following table describes the possible error response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	Permission issues.
500	An error occurred within the TSS Registry.

5.1.2 User Client Error Detection and the Error POST

The User Client may also encounter errors in the TSS responses. These types of errors are permanent errors since the User Client is the client and TSS can never "retry" a response. Therefore, there is no "temporary" concept of error in this direction.

The User Client communicates an error to TSS through a POST to a specific error URL: "/tss-user-api-error". The following is an example POST to indicate an error encountered by the User Client:

```
POST /api/v2/tss-user-api-error HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "error":"permanent",      // "permanent" is the only allowed value here
 "message": "Explanatory text is included here."
```

```
}
```

The TSS treats the error POST as a permanent error and issues this response as a way of acknowledging the error POST:

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/plain;charset=ISO-8859-1
Content-Length: <num>
Date: Tue, 13 Jan 2015 16:37:00 GMT
Connection: close

{
   "error": "permanent",
   "message": "Acknowledging the error POST."  // for example
}
```

The message field follows the same rules as described earlier.

The TSS SHOULD inform its operations personnel of the error encountered by the specific User Client when the error POST occurs. Since this is a permanent error, it is expected that TSS and User Client personnel would work together to resolve the issue. There is no expected automatic recovery for these errors.

The following are some examples of potential User Client encountered errors. This list is not expected to be complete, nor is the text in these examples required in the message field -- it is only for explanatory purposes in this document:

- A TFN was invalidly formatted (e.g. too few chars) in a response.
- A required field is missing in the JSON portion of the response.

Since there are no temporary failures encountered in this case, the User Client MUST ONLY issue error POSTs that can be fixed by the TSS. In other words the User Client MUST NOT issue errors for issues that it controls and TSS can do nothing about. Examples of issues that MUST NOT become errors in this case are:

- A database outage has occurred *at the User Client*.
- A maintenance window is in progress *at the User Client*.

Instead, the User Client should refrain from sending *any* POST until the User Client is back in service.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.

5.2 Change Password

The following is an example REST/JSON to change the password through API:

```
POST c HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "newPwd":"newpassword"
}
```

Change Password Fields		
Field	***Range***	***Comments***
/api/v2/tss-change-password	the URL	A request to change password
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The existing password for the ID above.
newPwd	New password	The new password being created for the ID above.

A response similar to the following is issued by TSS to the user on *success*:

```
{
   "status": "Password is successfully updated"
}
```

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
500	An error occurred within the TSS Registry.

5.3 Query TFN

The following is an example REST/JSON query request from the user:

```
POST /api/v2/tss-text-query-tfn HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "tfn":"8005551212"
}
```

Query TFN Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-query-tfn	the URL	A request to retrieve status and other information for the given TFN.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The password is set when the id is created and can be reset at any time via the website.
tfn	Exactly 10 printable digits.	The TFN for which status is being requested.

]A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url":"/api/v2/tss-text-query-tfn",        // positive confirmation
   "tfn":"8005551212",                        // positive confirmation
   "status":"<seeTableBelow>",                // see table below
   "routingTspid":"12345",                    // OPTIONAL, see below
   "ownerTspid":"12345",                      // OPTIONAL, see below
   "businessName":"Subscriber Business Name", // OPTIONAL, see below
   "contactName":"Subscriber Contact Name",   // OPTIONAL, see below
   "contactPhone":"Subscriber Contact Phone", // OPTIONAL, see below
   "contactEmail":"Subscriber Contact Email", // OPTIONAL, see below
   "ownerSrName":"Owner SR Name",             // OPTIONAL, see below
   "routingSrName":"Routing SR Name"          // OPTIONAL, see below

}
```

It is acceptable to query TFNs that are owned by other SRs. All of the above information is returned in this case, with the exception that the routingTspid can have a value of null to protect the private data of the owner of the TFN. The other fields are not considered to be private.

Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the status field on a successful request.

Status Field Values	
Status	***Comments***
"Enabled"	The TFN is enabled for text.
"Pending"	The TFN has been enabled by a Service Registrar and is pending validation by the Resp Org.
"Port_Pending"	The TFN is already enabled by a different Service Registrar and another Service Registrar enables it and is pending validation by the Resp Org.
"Unavailable"	The TFN is not in the TSS Registry DB, nor is it found in the SCP database. The TFN is not in a state to be text enabled
"Available"	The TFN is not in the TSS Registry DB, however it is found as a Working TFN in the SCP database. This TFN can be text enabled.

The following table describes the fields in the response.

Query TFN Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tfn	The TFN associated with this pending entry.
status	See the Status table above.
routingTspid	The TSPID used for text routing. It identifies a Service Registrar. TSS assigns a unique TSPID to each Service Registrar. *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null "routingTspid":null *Also, when the user does not have permissions to view the routing tspid for the TFN, the field is returned as* null.
ownerTspid	Typically the same as the routingTspid field, but it can be populated with a different TSPID to identify a Service Registrar that is responsible for the TFN; for example, billing responsibilities. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null
businessName	The toll-free subscriber's business name using the TFN. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null
contactName	The name of the toll-free subscriber responsible for enabling the TFN. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null
contactPhone	The phone of the toll-free subscriber responsible for enabling the TFN. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null
contactEmail	The email of the toll-free subscriber responsible for enabling the TFN. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null
ownerSrName	The name of the Owner Service Registrar who is responsible for the text-enabled TFN; for example, billing responsibilities. *All users have permissions to see this field.*

	This field is only valid for the Enabled and Pending status. Otherwise it is set to: null " ownerSrName ":null
routingSrName	The name of the Routing Service Registrar used for text routing. *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null " routingSrName ":null *Also, when the user does not have permissions to view the routing service registrar for the TFN, the field is returned as* null.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
403	The user does not have permission to query for <TFN>.
500	An error occurred within the TSS Registry.

5.4 Bulk TFN Query

The Bulk TFN API can be used to retrieve the TFN status, and other information for a list of TFNs. Currently up to 1,000 tfns in an array are allowed.

The following is an example REST/JSON query request to get the details of a list of TFNs:

```
POST /api/v2/tss-bulk-tfn-query HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "tfns":[                    // an array of tfns to query
      "8005001212",
      "8005001213",
      "8335001215
      ]
}
```

Bulk TFN Query Fields		
Field	***Range***	***Comments***
/api/v2/tss-bulk-tfn-query	the URL	A request to retrieve status and other information for a given list TFNs.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The existing password for the ID above.
tfns	The list of TFNs (each exactly 10 printable digits).	The list of TFNs for which status and details are being requested.

A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url":"/api/v2/tss-bulk-tfn-query",      // positive confirmation
   "tfnDetails":[                           // an array of tfn details
      {
         "tfn":"8005551212",                // positive confirmation
         "status":"<seeTableBelow>",        // see table below
         "roName":"Resp Org Name",          // OPTIONAL, see below
         "routingTspid":"12345",            // OPTIONAL, see below
         "ownerTspid":"12345",              // OPTIONAL, see below
         "ownerSrName":"Owner SR Name",     // OPTIONAL, see below
         "routingSrName":"Routing SR Name"  // OPTIONAL, see below
      },
      {
         "tfn":"8005001213",                // positive confirmation
         "status":"<seeTableBelow>",        // see table below
         "roName":" Resp Org Name ",        // OPTIONAL, see below
         "routingTspid":"12345",            // OPTIONAL, see below
         "ownerTspid":"12345",              // OPTIONAL, see below
         "ownerSrName":"Owner SR Name",     // OPTIONAL, see below
         "routingSrName":"Routing SR Name"  // OPTIONAL, see below
      },
      {
         "tfn":"8335001215",                // positive confirmation
         "status":"<seeTableBelow>",        // see table below
         "roName":" Resp Org Name ",        // OPTIONAL, see below
         "routingTspid":"12345",            // OPTIONAL, see below
         "ownerTspid":"12345",              // OPTIONAL, see below
         "ownerSrName":"Owner SR Name",     // OPTIONAL, see below
         "routingSrName":"Routing SR Name"  // OPTIONAL, see below
      }
    ]

}
```

Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the status field on a successful request.

Status Field Values	
Status	***Comments***
"Enabled"	The TFN is enabled for text.
"Pending"	The TFN has been enabled by a Service Registrar and is pending validation by the Resp Org.
"Port_Pending"	The TFN is already enabled by a different Service Registrar and another Service Registrar enables it and is pending validation by the Resp Org.
"Unavailable"	The TFN is not in the TSS Registry DB, nor is it found in the SCP database. The TFN is not in a state to be text enabled
"Available"	The TFN is not in the TSS Registry DB, however it is found as a Working TFN in the SCP database. This TFN can be text enabled.

The following table describes the fields in the response.

Bulk TFN Query Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tfn	The TFN.
status	See the Status table above.
roName	The Resp Org Name who is responsible for the voice-enabled TFN.
routingTspid	The TSPID used for text routing. It identifies a Service Registrar. TSS assigns a unique TSPID to each Service Registrar. *This field is only valid for the Enabled, Pending and Port_Pendingstatus. Otherwise it is set to:* null "routingTspid":null *Also, when the user does not have permissions to view the routing tspid for the TFN, the field is returned as* null.
ownerTspid	Typically the same as the routingTspid field, but it can be populated with a different TSPID to identify a Service Registrar that is responsible for the TFN; for example, billing responsibilities. *All users have permissions to see this field.* *This field is only valid for the Enabled, Pending and Port_Pending status. Otherwise it is set to:* null
ownerSrName	The name of the Owner Service Registrar who is responsible for the text-enabled TFN; for example, billing responsibilities. *All users have permissions to see this field.* *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null " ownerSrName ":null
routingSrName	The name of the Routing Service Registrar used for text routing. *This field is only valid for the Enabled and Pending status. Otherwise it is set to:* null " routingSrName ":null *Also, when the user does not have permissions to view the routing service registrar for the TFN, the field is returned as* null.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
500	An error occurred within the TSS Registry.

5.5 Text History

The text history API can be used to retrieve the history of events for a given text-enabled TFN in descending order of date and time the event occurred for that TFN.

This api will return the history of events based on roles and permissions for a given user. Below are some examples:

- If the user is a Resp Org and has access to the given TFN, this api will return the history of events for that TFN. If the user does not have access to the TFN, it will return an empty list.
- If the user is a Service Registrar who is currently responsible (owner SR) for a given TFN, this api will return the history of events that are pertaining to that TFN and Service Registrar combination. The user won't be able to get the history of events that happened before the TFN belonged to that Service Registrar.

The following is an example REST/JSON query request to get the event history for given text-enabled TFN:

```
POST /api/v2/tss-text-history HTTP/1.0
Content-Type: application/json
Content-Length: <num>

{
 "id":"user",
 "pwd":"password",
 "tfn":"8005001212"
}
```

Text History Fields		
Field	***Range***	***Comments***
/api/v2/tss-text-history	the URL	A request to retrieve event history for a given text-enabled TFN.
id	A User ID already created in TSS.	The id is created once via the website.
pwd	The User password associated with the above id.	The existing password for the ID above.
tfn	The TFN (exactly 10 printable digits).	The TFN for which event history is being requested.

A response similar to the following is issued by TSS to the user on *success*:

```
// The HTTP header is omitted for brevity.  The JSON portion follows:
{
   "url":"/api/v2/tss-text-history",     // positive confirmation
   "tfn":"8005001212",                   // TFN
   "events":[                            // an array of events
      {
         "tfn":"8005001212",
         "date":"01/12/2019 16:10:28 GMT",
         "event":"<seeTableBelow>",
         "rejectReasonId":null,
         "ownerTspid":"12345",
         "ownerSrName":"Owner SR Name",
         "routingTspid":"12345",
         "routingSrName":"Routing SR Name",
         "roName":"Resp Org Name",
         "ror":"ROR01",
         "businessName":"Subscriber Business Name",
         "contactName":"Subscriber Name",
         "contactJobTitle":"Subscriber Job Title",
         "contactPhone":"Subscriber Phone",
         "contactEmail":"Subscriber Email",
```

```
            "username":"username"
         },
         {
            "tfn":"8005001212",
            "date":"01/03/2019 17:31:45 GMT",
            "event":"<seeTableBelow>",
            "rejectReasonId":null,
            "ownerTspid":"12345",
            "ownerSrName":"Owner SR Name",
            "routingTspid":"12345",
            "routingSrName":"Routing SR Name",
            "roName":"Resp Org Name",
            "ror":"ROR01",
            "businessName":"Subscriber Business Name",
            "contactName":"Subscriber Name",
            "contactJobTitle":"Subscriber Job Title",
            "contactPhone":"Subscriber Phone",
            "contactEmail":"Subscriber Email",
            "username":"username"
         }
      ]                                          // end of the response array

}
```


Failure responses are discussed in the *Error* section.

The following table enumerates the possible values for the event field on a successful request.

Event Field Values	
Event	***Comments***
"approved"	Text-enablement request for the TFN has been approved by Resp Org or admin.
"auto-approved"	Text-enablement request for the TFN has been automatically approved by the system. This can happen if an agreement between the Service Registrar and Resp Org has been set up to do automatic approval upon text-enablement of a TFN.
"deleted"	The TFN has been deleted from the TSS database. This can happen when a TFN gets voice deactivated.
"disabled"	The TFN has been disabled for texting.
"expired-system-approved"	The system approved the text-enablement expired request for the TFN (was in pending status and Resp Org did not approve/reject the request within 2 business days) .
"expired-system-port-approved"	The system approved the text-enablement expired request for the TFN (was in port-pending status and Resp Org did not approve/reject the request within 2 business days).
"pending"	Text-enablement request for the TFN is pending validation from Resp Org.
"pending-cancelled"	Pending text-enablement request for the TFN got cancelled. This can happen when there is another request for text-enabling on the same TFN.
"port-approved"	Text-enablement request for the TFN which was owned by a different Service Registrar has been approved by Resp Org or admin.

"port-auto-approved"	Text-enablement request for the TFN which was owned by a different Service Registrar has been automatically approved by the system. This can happen if an agreement between the Service Registrar and Resp Org has been set up to do automatic approval upon text-enablement of a TFN.
"port-cancelled"	Text-enablement request for the TFN which is owned by different Service Registrar has been cancelled. This can happen when there is subsequent request to text-enable the same pending TFN.
"port-pending"	The text-enabled TFN is owned by one Service Registrar and text-enablement request for the same TFN with different Owner TSPID is pending validation from Resp Org.
"port-rejected"	Text-enablement request for the TFN which is owned by a different Service Registrar has been rejected by Resp Org or admin.
"port-requested"	Text-enablement request for the TFN which is owned by a different Service Registrar has been received.
"ported"	The TFN which was owned by one Service Registrar has been ported to another Service Registrar (owner TSPID changed). This event is recorded when the port is completed (either when TFN gets "port-approved" or "expired-system-port-approved").
"rejected"	Text-enablement request for the TFN has been rejected by Resp Org or admin.
"requested"	Text-enablement request for the TFN has been received.
"updated"	The details for TFN has been updated. This can happen when an already enabled TFN is enabled again with same TSPIDs.

The following table describes the fields in the response.

Text History Response Fields	
Field	***Comments***
url	Repeated in the response to make debugging easier.
tfn	The TFN for which the text history is being retrieved.
events	An array of events representing the history for the given TFN.
date	The timestamp when the event occurred. It is always relative to GMT. It is always of the format "MM/DD/YYYY HH:MM:SS GMT" where the fields have leading zeros. Finally, the HH uses a 24 hour clock: 00..23.
event	See the 'Event Field Values' table above.
rejectReasonId	The reject reason code. *The field will have a value if the event type is 'rejected' else the field is returned as null.*
ownerTspid	The TSPID that identifies the Service Registrar that is responsible for the TFN; for example, billing responsibilities.
ownerSrName	The name of the Owner ervice Registrar who is responsible for the text-enabled TFN; for example, billing responsibilities.
routingTspid	The TSPID used for text routing. It identifies a Service Registrar. TSS assigns a unique TSPID to each Service Registrar. *When the user does not have permissions to view the routing tspid for the TFN, the field is returned as* null.
routingSrName	The name of the Routing Service Registrar used for text routing. *When the user does not have permissions to view the routing service registrar for the TFN, the field is returned as* null.
roName	The Resp Org Name who is responsible for the voice-enabled TFN.
ror	The Resp Org ID associated with voice registration of the TFN.
businessName	The toll-free subscriber's business name using the TFN.

contactName	The name of the toll-free subscriber responsible for the enable request.
contactJobTitle	The job title of the toll-free subscriber responsible for the enable request.
contactPhone	The phone of the toll-free subscriber responsible for the enable request.
contactEmail	The email of the toll-free subscriber responsible for the enable request.
username	The user name of the user who caused/initiated the event.

The following table describes the possible response HTTP status codes.

Response HTTP Status Codes	
HTTP Status Codes	***Reason***
200	The request has succeeded.
400	Bad request (invalid JSON format, failed validation).
401	Authentication failed.
500	An error occurred within the TSS Registry.

6. Using wget and curl for Debugging

There are several tools that can be used to aid testing of any of the POSTs. Below are two examples for querying a TFN that can be used in a UNIX environment. Be sure to replace the MyUserID and MyPass with proper values. Also note that the use of single and double quotes in the examples below is likely to be important to avoid command shell issues. Similarly, your shell may need to escape (backslash) special characters.

The wget tool can issue a query POST similar to this example:

```
/bin/wget --no-check-certificate \
      --header="Content-Type: application/json" -O - \
      --post-data='{"id":"MyUserID", "pwd":"MyPass", \
      "tfn":"8005551212"}' \
      https://texting.somos.com/api/v2/tss-text-query-tfn
```

The curl tool can also issue a query POST similar to this example:

```
/usr/bin/curl --insecure -i -H "Content-Type: application/json" -X \
      POST -d '{"id":"MyUserID", "pwd":"MyPass", \
      "tfn":"8005551212"}' \
      https://texting.somos.com/api/v2/tss-text-query-tfn
```

These are 3rd party tools that may also have “man” pages on your operating system. More information about the tools can be found via a web search engine.