



UNITED STATES
FEDERAL COMMUNICATIONS COMMISSION
INFORMATION TECHNOLOGY

NETWORK OUTAGE REPORTING SYSTEM (NORS) APPLICATION PROGRAMMING INTERFACE (API) DEVELOPER GUIDE

445 12TH STREET SW, WASHINGTON, DC 20554

Revision Log

Date	Description	Author
3/2/17	Initial Release	FCC IT Development

Table of Contents

- Revision Log 1
- Table of Contents 2
- 1. Summary 3
- 2. REST 3
 - REST Console 4
 - Postman 5
 - Download Postman 6
 - Authentication 6
 - Making and Saving Calls 7
 - Validating your JSON 10
- 3. SOAP 10
 - SOAP 10
 - WSDL 10
 - SoapUI 11
 - Authentication 12
- 4. XML Upload 13
- 5. Troubleshooting 15
- Appendix A - Fields 16
- Appendix B – Transitioning from the Appian API 28
 - SOAP 28
 - XML 29

1. Summary

The Federal Communications Commission’s (FCC) Network Outage Reporting System (NORS) allows communications providers to report information about significant disruptions or outages to their communications systems that meet specified thresholds set forth in Part 4 of the FCC’s rules (47 C.F.R. Part 4). Users can access NORS from a link on the FCC’s website (<https://www.fcc.gov/network-outage-reporting-system-nors>), or programmatically, through the NORS Application Programming Interface (API). This document provides details on the latter.

This guide assumes prior understanding of the functions and business processes of the NORS application. Refer to the NORS User Manual specific NORS questions.

The FCC has provided two industry-standard methods for accessing the NORS API:

- **Simple Object Access Protocol (SOAP)**, utilizing eXtensible Markup Language (XML)
- **REpresentational State Transfer (REST)**, utilizing JavaScript Object Notation (JSON)

Additionally, a bulk XML File Upload capability has been provided to retain backwards compatibility with previous versions of the NORS application.

REST is the newer of the two standards, and going forward, the FCC will focus the bulk of its resources on developing API enhancements in REST. It is therefore recommended that you use REST if at all possible.

2. REST

REST is the current industry standard for data interchange between applications. In REST, each request consists of the pairing of an HTTP **VERB** with an optional data body in **JSON** (Javascript Object Notation) format, allowing web applications to perform the same functions as traditional client/server applications writing to a back-end database.

The HTTP VERBs tell the receiving application what to do with the data in the request, and they roughly match the traditional actions of a CRUD (Create, Read, Update, and Delete) application:

HTTP VERB	CRUD Equivalent	Notes
GET	READ	GET is the default verb used by browsers when accessing a website. The resource identifier to be read is given in the URL, usually as part of a QUERY STRING (all the stuff behind the “?” in a URL). For a GET, there is no data in the body of the request, but data is returned in the body of the response.
POST	INSERT	POST is used to insert a new record in the target application. The target resource is given in the URL, and the data to be inserted is provided in the body of the request. The target application creates a new record and returns the new record’s Identifier in its response.
PUT	UPDATE	A PUT request tells the target application to update the given record ID (provided as a parameter in the URL) with the data in the body of the request.

HTTP VERB	CRUD Equivalent	Notes
PATCH	UPDATE	PATCH is a variation of a PUT request. In strict REST, a PUT replaces the entire record with the record given in the body, while PATCH tells the target to only update the specific fields provided. In the NORS implementation, we are only using the PUT verb, but it works in a way that combines the functions of PUT and PATCH.
DELETE	DELETE	DELETE does exactly that. The NORS API does not allow any deletes.

Given the above definitions, it follows that the following functions are available through the NORS REST API:

GET – Request any Outage (that you have rights to see) by the Outage Number.

POST Notification – The first step in Outage reporting is to create a new Notification. The Outage Number is generated by the system and returned to the requestor.

PUT Initial – Next, the Outage report is updated any number of times with status Initial.

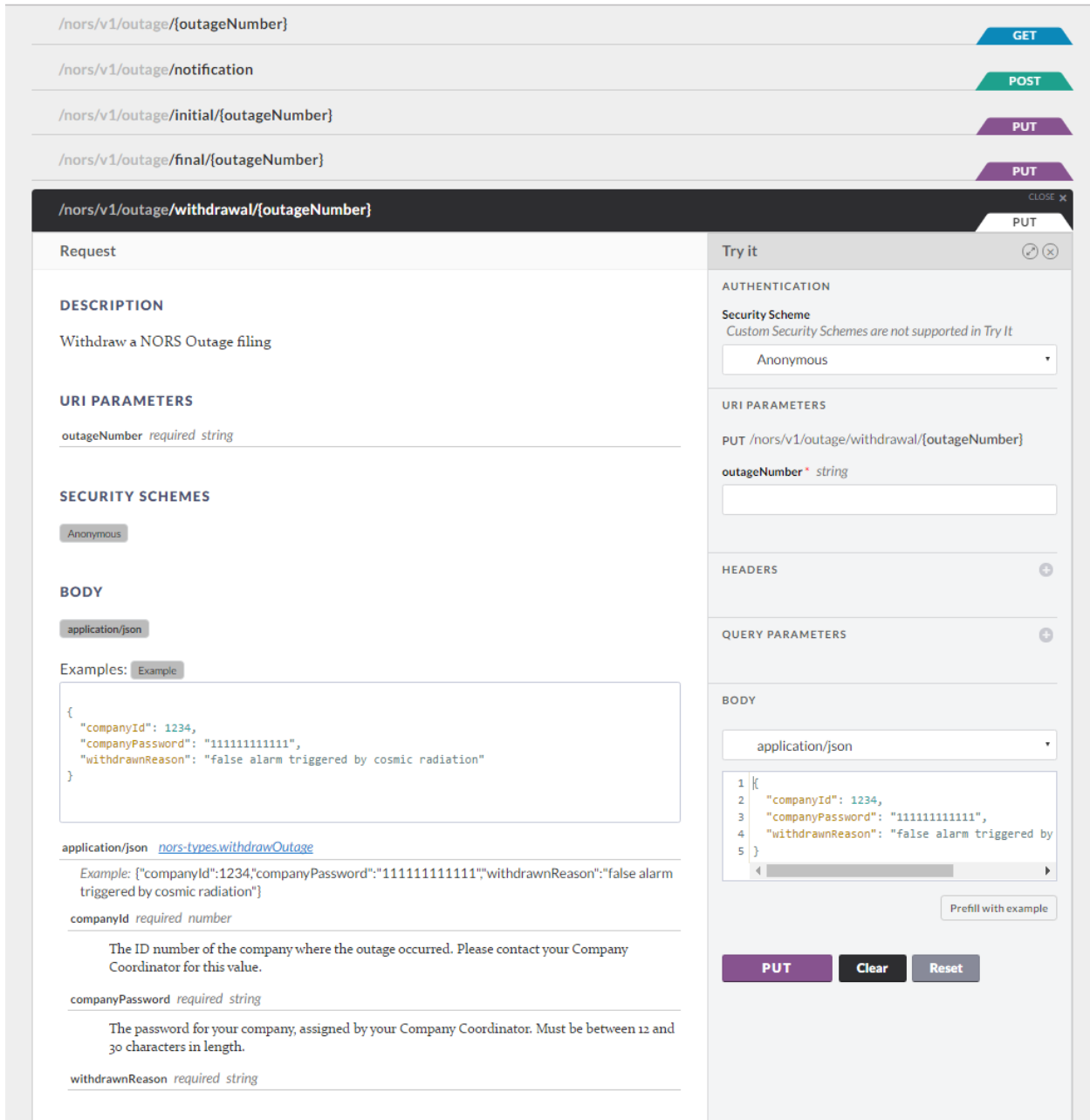
PUT Final – The last update of the Outage report is set with status Final. No further updates to the report are allowed.

PUT Withdrawal – Before the Final stage, it’s possible to request a Withdrawal of an Outage report. This is the only way for the user to effectively delete an invalid report, but from a strict technical standpoint, it’s still an update of an existing report.

REST Console

The REST API includes a web-based console that provides additional online help and technical documentation for the API; it is a good place to start when starting to build your own interface (https://api.fcc.gov/nors_console/).

This site describes all the available fields for Outage reporting and the types of fields and allowed values for each field, a list of the individual calls that can be made, and a sample of what the request body should look like for each type of call. Simply click on one of the HTTP verbs to expand the section.



Request

DESCRIPTION

Withdraw a NORS Outage filing

URI PARAMETERS

outageNumber *required string*

SECURITY SCHEMES

Anonymous

BODY

application/json

Examples: [Example](#)

```
{
  "companyId": 1234,
  "companyPassword": "111111111111",
  "withdrawnReason": "false alarm triggered by cosmic radiation"
}
```

application/json [nors-types.withdrawOutage](#)

Example: ["companyId":1234,"companyPassword":"111111111111","withdrawnReason":"false alarm triggered by cosmic radiation"]

companyId *required number*

The ID number of the company where the outage occurred. Please contact your Company Coordinator for this value.

companyPassword *required string*

The password for your company, assigned by your Company Coordinator. Must be between 12 and 30 characters in length.

withdrawnReason *required string*

Try it

AUTHENTICATION

Security Scheme
Custom Security Schemes are not supported in Try It

Anonymous

URI PARAMETERS

PUT /nors/v1/outage/withdrawal/{outageNumber}

outageNumber * *string*

HEADERS

QUERY PARAMETERS

BODY

application/json

```
1 | {
2 |   "companyId": 1234,
3 |   "companyPassword": "111111111111",
4 |   "withdrawnReason": "false alarm triggered by
5 | }
```

Prefill with example

PUT Clear Reset

Note: Although there is a section of the page that says “Try it”, it currently does not support the Basic Authentication scheme that is required by the system, and therefore will not work. To test, please use a utility like Postman, described below.

Postman

All modern development languages provide tools and methods to simplify the process of making REST calls. However, during development, it helps to have utilities to assist with building and test. Without explicit endorsement, the following section describes the use of a popular tool for this purpose. Postman is an extension for the Google Chrome browser that can also be downloaded for standalone use. In addition to providing facilities to control all aspects of the HTTP request, it allows you to save all your requests for easier testing.

Download Postman

Postman is available at <https://www.getpostman.com>

Note that when you first install it, it may ask you to sign up or log in, but there is a link at the bottom of the page that allows you to continue without logging in.

Authentication

The NORS API uses Basic Authentication to validate users of the system. Following the Basic Authentication standard, a header must be included with each request in the format

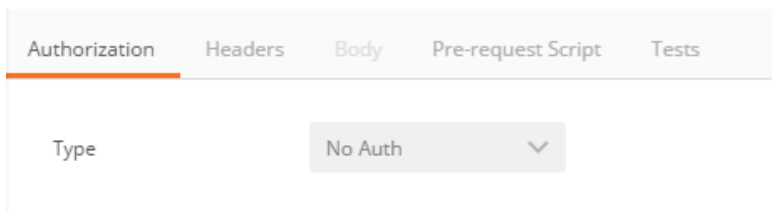
```
Authorization=Basic xxxxxxxxxx
```

Where the “xxxxxxx” above is a Base64-encoded string representing the user’s

```
Email:password
```

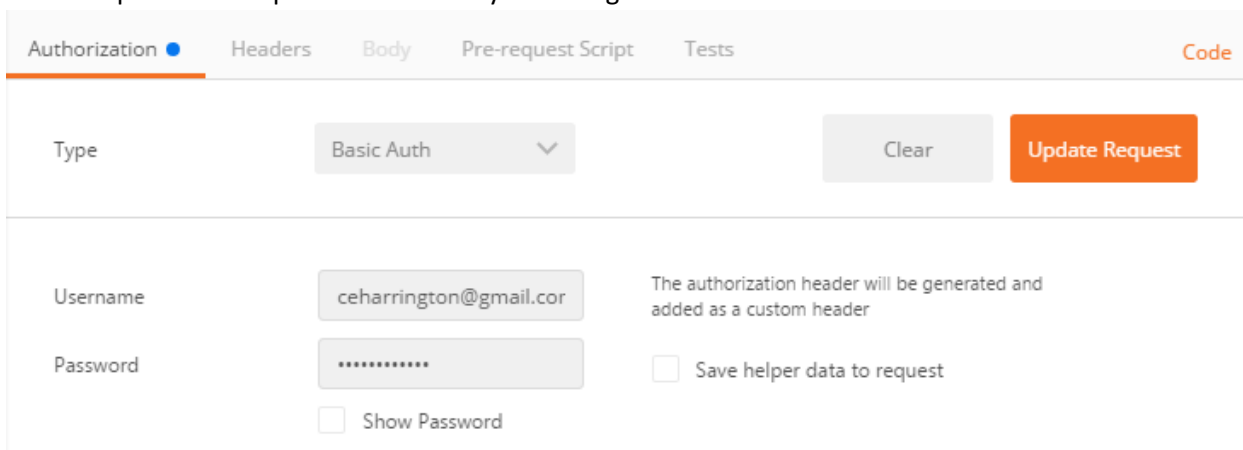
When using Postman to make your test call, you can add the Authorization header as shown below.

First, select the Authorization tab in the main content section. By default, “**No Auth**” will be selected. Click on the dropdown and select “**Basic Auth**”.



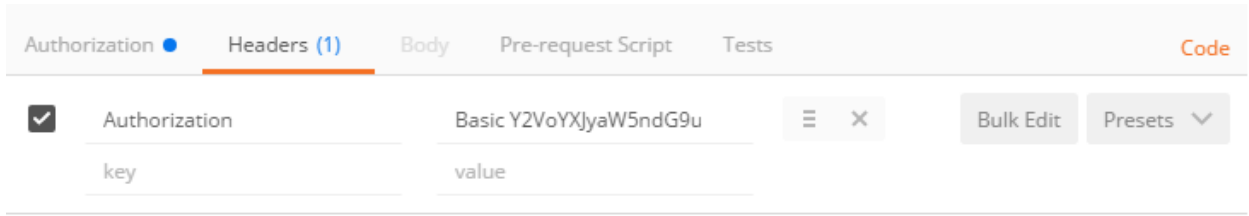
The screenshot shows the Postman interface with the 'Authorization' tab selected. The 'Type' dropdown menu is set to 'No Auth'.

Fill in the username and password, and press the orange “Update Request” button to encode the values and add them to the request header. Note that you **must do this every time you make a change**, or it will not update the request header with your changes.



The screenshot shows the Postman interface with the 'Authorization' tab selected. The 'Type' dropdown menu is set to 'Basic Auth'. The 'Username' field contains 'ceharrington@gmail.cor' and the 'Password' field is masked with dots. The 'Update Request' button is highlighted in orange. A note states: 'The authorization header will be generated and added as a custom header'. There are also 'Clear' and 'Show Password' options.

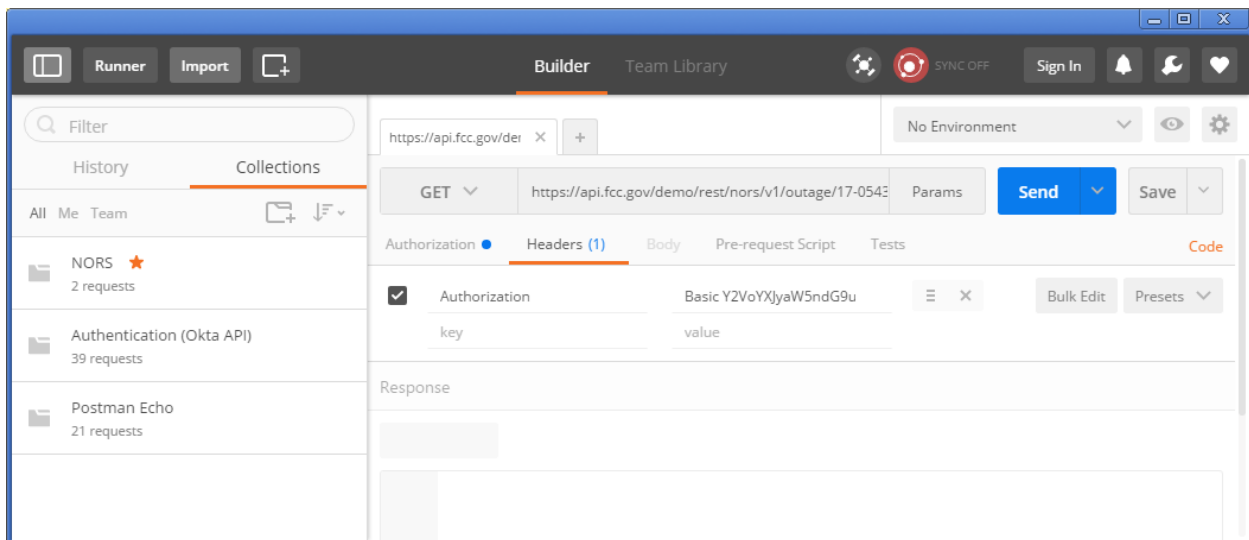
Now if you select the Headers tab, you’ll see that an authorization header will have been added.



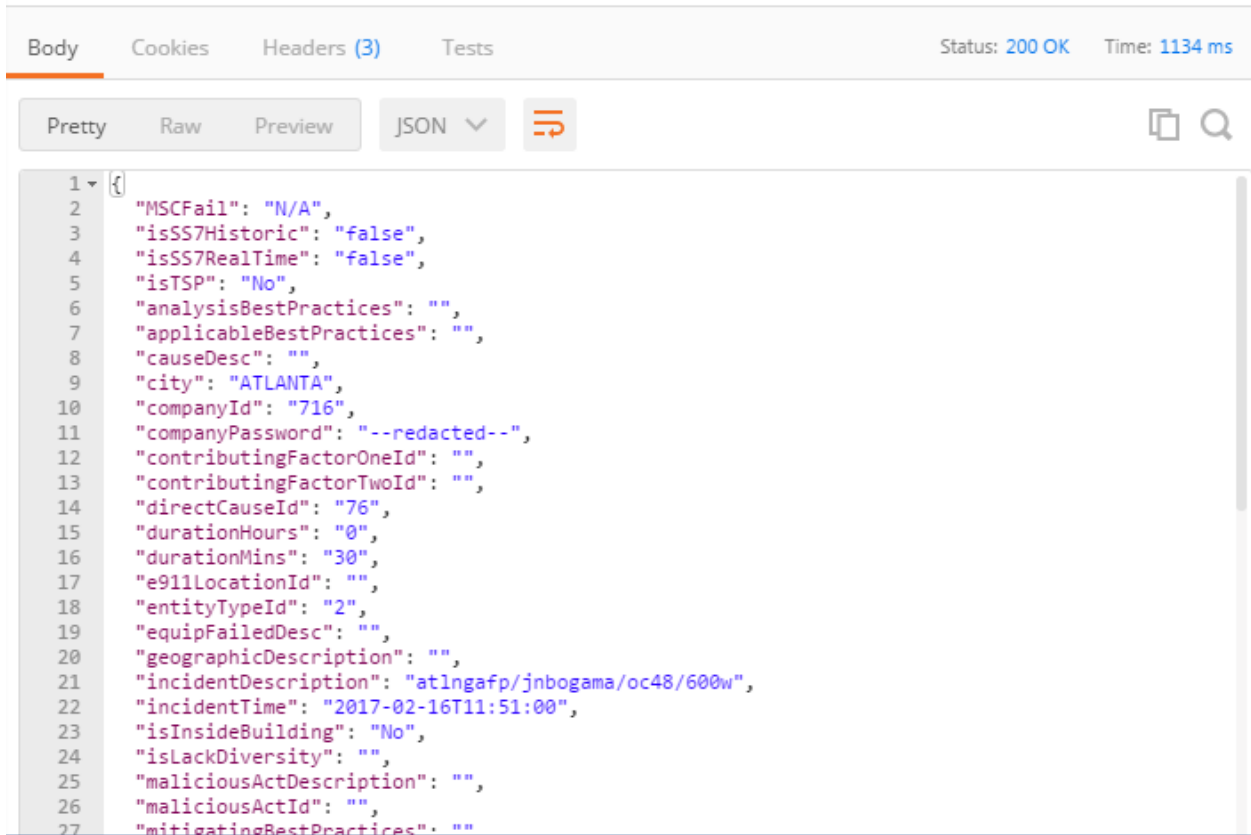
Making and Saving Calls

In the main content area, enter a valid endpoint and HTTP verb.

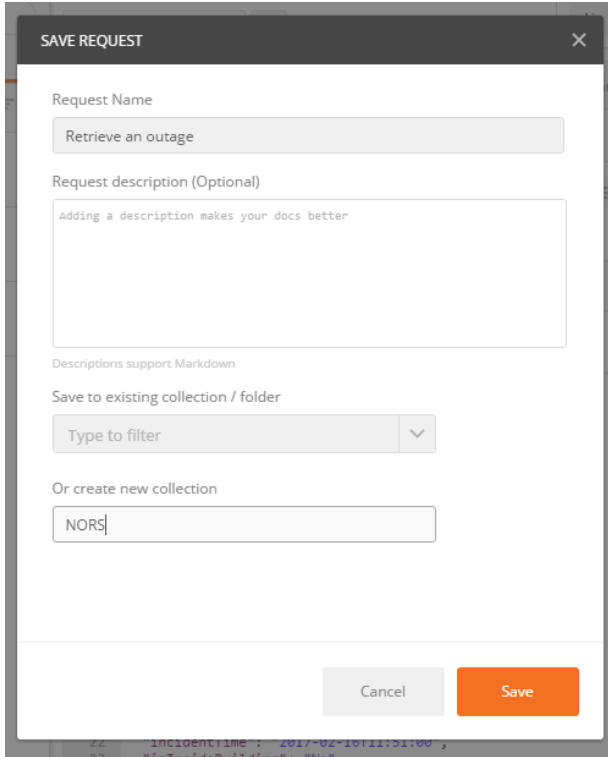
For our simple example, we'll perform a GET on an existing outage. Make sure "GET" is selected in the dropdown, and then enter the endpoint <https://api.fcc.gov/rest/nors/v1/outage/xx-yyyyyyyy>, where xx-yyyyyyyy is one of your existing outage reports.



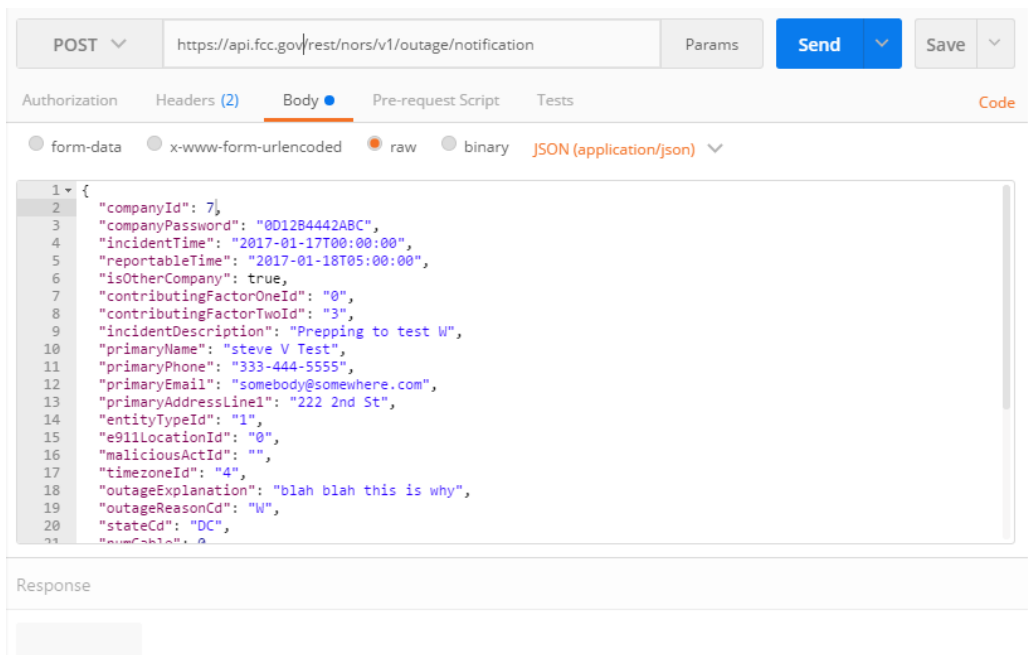
Press Send to get a response.



Next to the SEND button is a SAVE button. Click on the arrow **next to** the SAVE, and select SAVE TO. This pops up a dialog window that allows you to fill out a Name and optional description. You can also create a new Collection, allowing you to group multiple saved requests together. These saved requests will appear in the left side of the Postman window.



GETs do not require a request body, but POSTs and PUTs do, so for Notifications, Initials, and Finals, you will need to navigate to the **Body** tab and add the JSON body for your outage. In the Body options, select “raw” to be able to paste in the entire JSON, and select the content type of “application/json”.



POST https://api.fcc.gov/rest/nors/v1/outage/notification Params Send Save

Authorization Headers (2) **Body** Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)

```

1 {
2   "companyId": 7,
3   "companyPassword": "001284442ABC",
4   "incidentTime": "2017-01-17T00:00:00",
5   "reportableTime": "2017-01-18T05:00:00",
6   "isOtherCompany": true,
7   "contributingFactorOneId": "0",
8   "contributingFactorTwoId": "3",
9   "incidentDescription": "Prepping to test W",
10  "primaryName": "steve V Test",
11  "primaryPhone": "333-444-5555",
12  "primaryEmail": "somebody@somewhere.com",
13  "primaryAddressLine1": "222 2nd St",
14  "entityTypeId": "1",
15  "e911LocationId": "0",
16  "maliciousActId": "",
17  "timezoneId": "4",
18  "outageExplanation": "blah blah this is why",
19  "outageReasonCd": "W",
20  "stateCd": "DC",
21  "statusCd": "A"

```

Response

Validating your JSON

The JSON body in your request must be properly formatted. There are a few rules to keep in mind:

- Begin and end your JSON body with curly braces {}
- Each field should be given as "field" : "value"
- Place a comma after each key:value pair, except the last one
- Although some fields accept values without the quotes, the simplest option is to just quote every value. For example, { "durationHours" : "2" }
- For the fields called "num..." or "duration...", do not pass a null value (""). Use a zero ("0"), or omit the field from the JSON completely.
- It's ok to omit fields for which you don't want to provide a value.
- Conversely, you're allowed to fill in as many of the optional fields as you like, at each of the stages of Notification, Initial, and Final. They will get updated in the outage report.
- There are also no ill-effects of providing fields that don't match any of the NORS outage fields. This may be useful if, for example, your internal system contains a reference ID code for a given outage. Including this field in your call to the NORS API will not affect the report submission; the NORS API will simply ignore the unrecognized field.
- For an example starting point, you can use the sample JSON that's available in the NORS REST Console.
- If you'd like to ensure that your JSON is valid, there are a number of online validators available. A simple one can be found at <http://jsonprettyprint.com> . Note that this site is neither sponsored by, nor endorsed by, the FCC.

3. SOAP

SOAP

Simple Object Access Protocol (SOAP) is an open-standard messaging protocol for exchanging information among computers without regard for operating system. SOAP allows those computers to communicate using Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML). For more details on SOAP, and an explanation of related terms, please reference <https://en.wikipedia.org/wiki/SOAP> .

WSDL

Web Service Definition Language (WSDL) is a "description language." As defined by the World Wide Web Consortium (W3C), WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly and then bound to a concrete network protocol and message format to define an endpoint. (**Source:** <https://www.w3.org/TR/wsdl>)

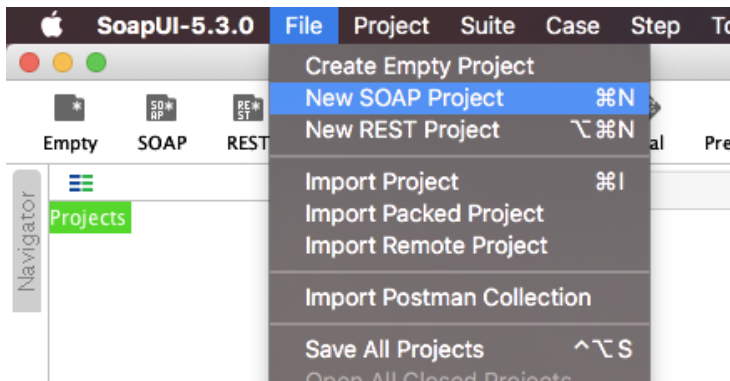
WSDL is XML, but does not carry data to the web service. Instead, WSDL describes to your machine how to connect to the web service machine and make requests to the web services. The description contained in the WSDL XML includes the name of the service, the location of the service, and ways to communicate with the service. For NORS, the WSDL file is available from https://api.fcc.gov/soap/NORS_Outage_Receiver?wsdl , but you can also find a link to it in the NORS application under the "NORS API/XML" heading.

SoapUI

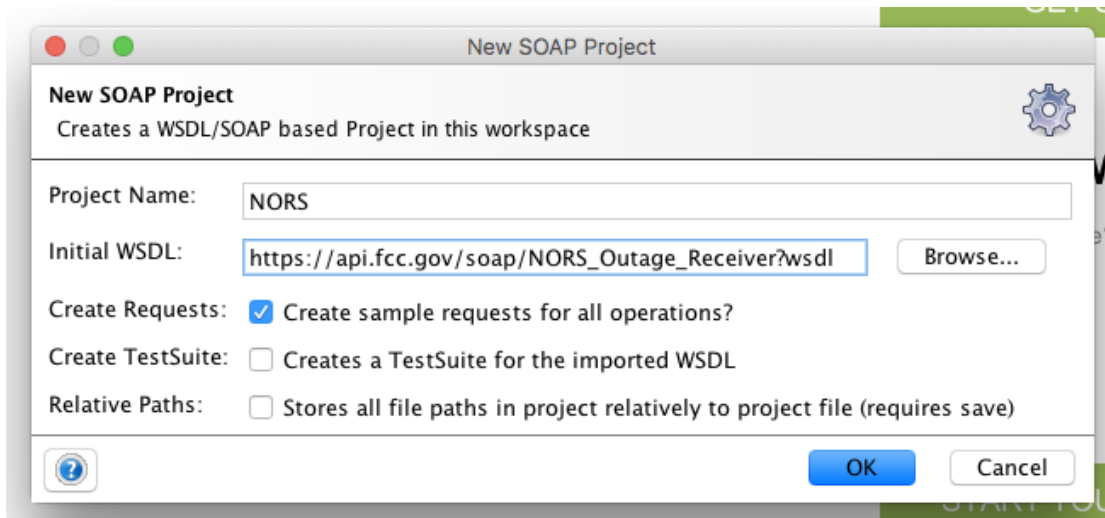
There are a number of SOAP clients available for testing and development assistance. Without explicit endorsement, we will use one of the more popular ones for the example below, called SoapUI. It is available from <https://www.soapui.org/>.

Many development toolkits can automatically generate a working SOAP Envelope from a provided WSDL file, and SoapUI is no exception. First, you will need the WSDL file from NORS; see the above section for instructions on how to get it.

Open the application and select “New SOAP Project” from the menu. Note that the screenshots below are from MacOS; menu locations in Windows may vary slightly.

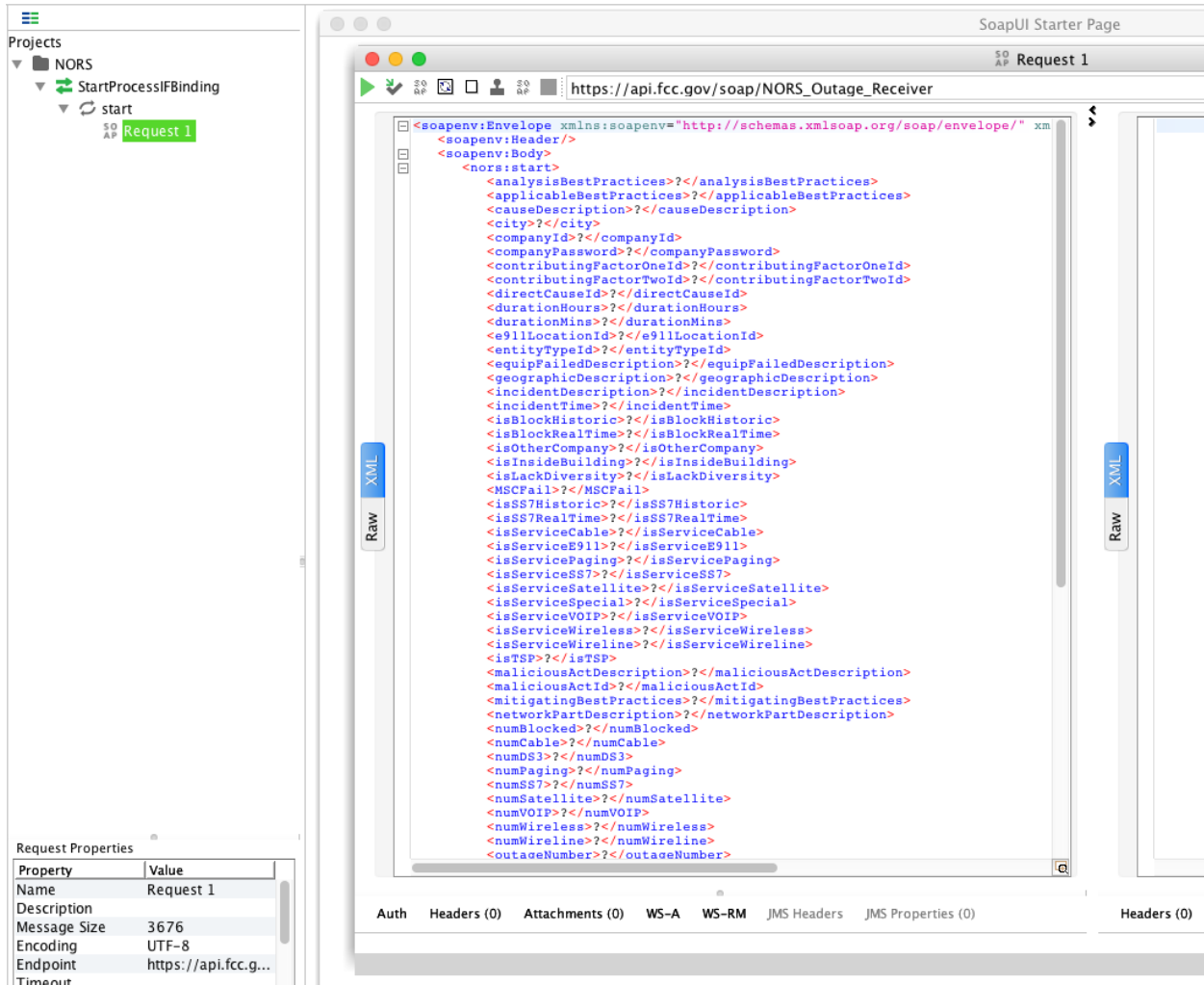


Give the project a name, and either browse for the downloaded WSDL file, or provide the URL of the NORS WSDL in the field shown. Click OK to generate the project files.



A sample SOAP request envelope will be created for you. Fill in values for the fields as necessary. Note that with this version of the SOAP API, it's not necessary to include optional fields that you intend to leave blank. [Appendix A](#) has a listing of the field names and values, but you can also refer to the REST

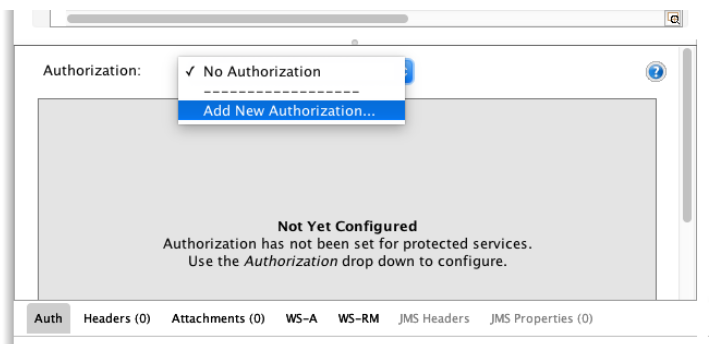
API Console for a more detailed explanation of the fields.



Make sure to include a valid OutageTypeCd.

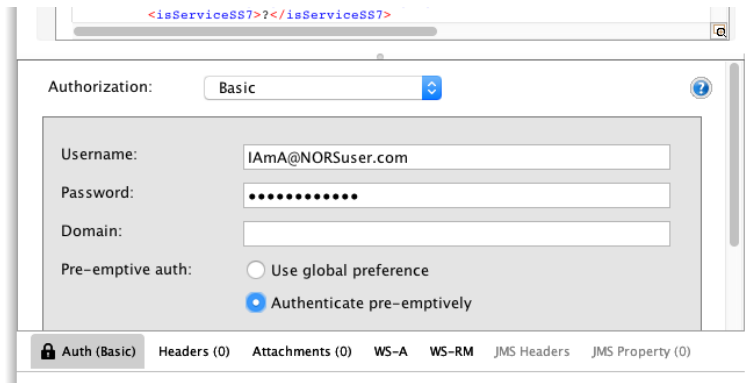
Authentication

As with a REST call, authentication is also required for SOAP requests to NORS. In SoapUI, this can be added by clicking on the **“Auth”** link at the bottom left of the request window.



Select **“Add New Authorization”** from the dropdown.

In the new dialog box, fill in your FCC credentials.



Make sure to select the **“Pre-emptive Authentication”** checkbox. You can also set this preference in the global settings for the SoapUI application.

Click the **GREEN** arrow at the top left of the window to send the request. The results will appear in the right half of the request window. Look in the `<activityMsg>` field of the results for a “Success” message, or for an error messages if the request did not complete.

4. XML Upload

The XML Upload option has been retained for legacy purposes. For users that cannot use REST or SOAP to connect to NORS, bulk uploading of outage reports can be accomplished using this method.

The data should be properly formatted XML in the structure shown below:

```
<?xml version="1.0" encoding="utf-8"?>
<Outages>
  <NORS_Outage>
    <outageTypeCd>N</outageTypeCd>
    <companyId>8</companyId>
    <companyPassword>40860A8CC7#u</companyPassword>
    <entityTypeId>1</entityTypeId>
    <timeZoneId>1</timeZoneId>
    <outageReasonCd>E</outageReasonCd>
    <incidentTime>2012-12-13T12:12:12</incidentTime>
    <stateCd>CA</stateCd>
    <incidentDesc>str1234</incidentDesc>
    <numWireline>12</numWireline>
    <numWireless></numWireless>
    <numVoip></numVoip>
    <numPaging></numPaging>
    <numCable></numCable>
    <numSatellite></numSatellite>
    <numBlocked></numBlocked>
    <numDS3></numDS3>
    <numSS7></numSS7>
    <primaryName>John Smith</primaryName>
    <primaryPhone>2223334444</primaryPhone>
    <primaryEmail>john.smith@fcc.gov</primaryEmail>
    <primaryAddress1></primaryAddress1>
  </NORS_Outage>
</Outages>
```

```

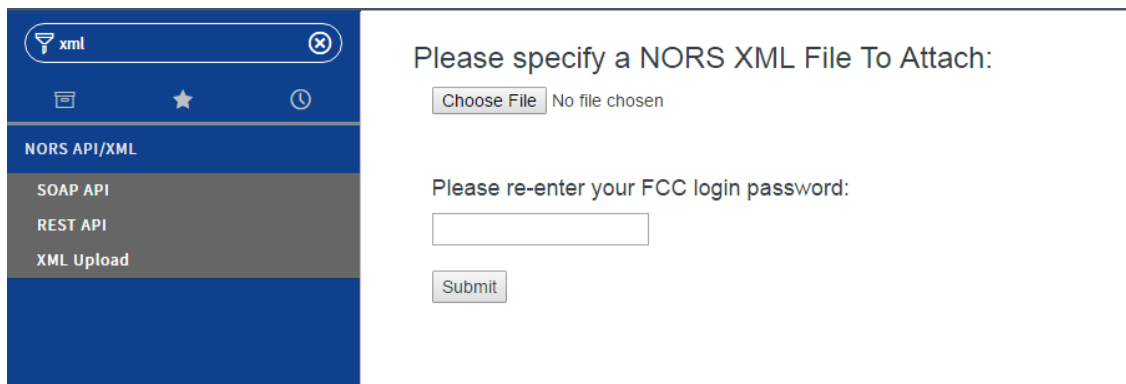
    <NORS_Outage>
      ...
    </NORS_Outage>
  </Outages>

```

Note:

- Multiple <NORS_Outage> nodes may be included in each XML upload file.
- If the outage inputter is associated with multiple companies, it's possible to submit separate outages for different companies in a single upload file. For this reason, the <companyPassword> field has been moved into the XML.
- The fields in the example above are meant for a structural example. Note that for each Outage Type, there are different sets of required fields. Please reference [Appendix A](#) for specific field definitions.
- It is not necessary to provide every field in every outage report. Optional fields can be left out.

Log into NORS and select **XML Upload** from the left navigation bar, underneath the “NORS API/XML” Heading.



On this page, select your XML file, retype your FCC login password, and press the **SUBMIT** button.

5. Troubleshooting

Most API errors result from one of the following issues. If you are experiencing problems in using the API, please first review your code/data to ensure compliance with the items below.

- ****DO NOT** cut & paste** values from a Microsoft Office document (including this one)!! Word and Excel insert many non-standard, unprintable, or unintended characters into copied text, and when they make it into code, they invariably cause problems.
- **Field names are all case-sensitive.** Please make sure your field names EXACTLY match the names as listed in Appendix A.
- For the authentication header, make sure to enable **pre-emptive authentication**.
- For REST, use **quotes for all field values**, regardless of type. For example, use `{"durationMins": "44"}` instead of `{"durationMins": 44}`. Depending on your specific programming language, this may require you to convert values to String. Note that the NORS API will always return all values as Strings.
- For REST, the fields that have names starting with "**num...**" and "**duration...**" will not accept null strings (""). You can either leave out the field from your data, or **pass in a zero** ("0").
- For REST, the different outage types require **different HTTP verbs**. Please see the directions for REST above, or reference the REST Console online documentation at https://api.fcc.gov/nors_console/.
- For REST, boolean-type values should be either "**true**" or "**false**". Make sure to use **lower case**.
- For SOAP, boolean-type values accept 1 or **true**, and 0 or **false**. Make sure to use **lower case**.
- For SOAP, check the **namespace definitions** in your SOAP envelope. If you used the system during our public testing period, note that the production namespace differs slightly from the one used during the test period.
- For field **MSCFai1**, the valid values are "**yes**", "**no**", and "**N/A**", NOT "**true**"/"**false**".
- If you are switching between the different API interface options, note that many of the fields have slightly **different names**.

Appendix A - Fields

The API for this version of NORS has been designed to minimize impact on users of the previous versions of the API, and for this reason, the field names have been retained. Unfortunately, those field names differed slightly between the XML Upload and SOAP API; all are listed below for your reference. Note that all of the API interfaces are case-sensitive.

For an online listing of the fields and their values, see the REST API console at https://api.fcc.gov/nors_console/.

A few general notes:

- For the ostensibly Boolean fields, SOAP and XML Uploads will accept "1" or "true" for True, and "0" or "false" for False.
- For the ostensibly Boolean fields, REST accepts only "true" or "false".

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
analysisBestPractices	analysisBestPractices	analysisBestPractices		
applicableBestPractices	applicableBestPractices	applicableBestPractices		
causeDesc	causeDescription	causeDesc		
city	city	city		
companyId	companyId	companyId	N I F W	Numeric identifier assigned to each company
companyPassword	companyPassword	companyPassword	N I F W	12-30 character secret passcode, set by each company's coordinator.
contributingFactorOneId	contributingFactorOneId	contributingFactorOneId		See <code>rootCauseId</code> below for values Additionally, value "0" is allowed to denote "None".
contributingFactorTwoId	contributingFactorTwoId	contributingFactorTwoId		See <code>rootCauseId</code> below for values Additionally, value "0" is allowed to denote "None".
directCauseId	directCauseId	directCauseId	F	See <code>rootCauseId</code> below for values
durationHours	durationHours	durationHours	F	
durationMinutes	durationMins	durationMins	F	
e911LocationId	e911LocationId	e911LocationId		0- None 1- ALI Only Affected

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				2- Phase II Only Affected 3- Phase I and Phase II Only Affected 4- More than Location Affected
entityTypeId	entityTypeId	entityTypeId	N I F	1- Wireless Carrier 2- Wireline Carrier 3- Cable Telephony Provider 4- Paging Provider 5- Satellite Provider 6- SS7 Network Provider 7 - E911 Service Provider 8- Facility Owner or Operator 9- VoIP Provider
equipFailedDesc	equipFailedDescription	equipFailedDesc		
geographicDescription	geographicDescription	geographicDescription		
incidentDesc	incidentDescription	incidentDescription	N I F	
incidentTime	incidentTime	incidentTime	N I F	In ISO 8601 format, without the timezone offset information. YYYY-MM-DDTHH:mm:ss
isBlockHistoric	isBlockHistoric	isBlockHistoric		"true" "false" (default)
isBlockRealtime	isBlockRealTime	isBlockRealTime		Note capitalization differences "true" "false" (default)
isOtherCompany	isOtherCompany	isOtherCompany		"true" "false" (default)
isInsideBuilding	isInsideBuilding	isInsideBuilding	F	"true" "false" (default) ""
isLackDiversity	isLackDiversity	isLackDiversity	F	"true" "false" (default) ""

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
MSCFail	MSCFail	MSCFail	F	Did the Mobile Switching Center (MSC) fail? Note that the only valid values are: "yes" "no" "N/A"
isSS7Historic	isSS7Historic	isSS7Historic		"true" "false" (default)
isSS7Realtime	isSS7RealTime	isSS7RealTime		Note capitalization differences "true" "false" (default)
isServiceCable	isServiceCable	isServiceCable		"true" "false" (default)
isServiceE911	isServiceE911	isServiceE911		"true" "false" (default)
isServicePaging	isServicePaging	isServicePaging		"true" "false" (default)
isServiceSS7	isServiceSS7	isServiceSS7		"true" "false" (default)
isServiceSatellite	isServiceSatellite	isServiceSatellite		"true" "false" (default)
isServiceSpecial	isServiceSpecial	isServiceSpecial		"true" "false" (default)
isServiceVoip	isServiceVOIP	isServiceVOIP		"true" "false" (default)
isServiceWireless	isServiceWireless	isServiceWireless		"true" "false" (default)
isServiceWireline	isServiceWireline	isServiceWireline		"true" "false" (default)
isTsp	isTSP	isTSP	F	Note capitalization differences. Was Telecommunications Service Priority involved in the Service Restoration? "true"

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				"false" (default) ""
maliciousActDescription	maliciousActDescription	maliciousActDescription	F	Only required if maliciousActId is not "no"
maliciousActId	maliciousActId	maliciousActId	F	0- No 1- Yes-Physical event 2- Unknown 3- Yes-Cyber event 9- Unselected
mitigatingBestPractices	mitigatingBestPractices	mitigatingBestPractices		
networkPartDesc	networkPartDescription	networkPartDesc		
numBlocked	numBlocked	numBlocked	N I F	At least one of the num... fields must be provided.
numCable	numCable	numCable	N I F	At least one of the num... fields must be provided.
numDS3	numDS3	numDS3	N I F	At least one of the num... fields must be provided.
numPaging	numPaging	numPaging	N I F	At least one of the num... fields must be provided.
numSS7	numSS7	numSS7	N I F	At least one of the num... fields must be provided.
numSatellite	numSatellite	numSatellite	N I F	At least one of the num... fields must be provided.
numVoip	numVOIP	numVOIP	N I F	Note capitalization differences. At least one of the num... fields must be provided.
numWireless	numWireless	numWireless	N I F	At least one of the num... fields must be provided.
numWireline	numWireline	numWireline	N I F	At least one of the num... fields must be provided.
outageExplanation	outageExplanation	outageExplanation		

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
outageNumber	outageNumber	<none>	I F W	For REST, this is part of the URL endpoint, but the API response will include the field.
outageReasonCd	outageReasonCd	outageReasonCd	N I F	NF- Not Filled-In Yet WE- Wireline-900,000 User-Minutes W - Wireless-900,000 User-Minutes C - Cable Telephony-900,000 User-Minutes M - MSC E - E911 B - Blocked Calls D - 1350 DS3s Minutes DS- DS3-Simplex Greater Than 5 Days S - SS7-MTP Messages A - Airport SF- Other Special Facilities- (Military,Nuclear,etc.) P - Paging ST- Satellite O - Other VE- VoIP-E911 VU- VoIP-900,000 User-Minutes
outageTypeCd	outageTypeCd	<none>		Notification, Initial, Final, Withdrawal For REST, this is part of the URL endpoint.
preventativeSteps	preventativeSteps	preventativeSteps		
primaryAddress1	primaryAddressLine1	primaryAddressLine1		
primaryAddress2	primaryAddressLine2	primaryAddressLine2		
primaryAddress3	primaryAddressLine3	primaryAddressLine3		
primaryEmail	primaryEmail	primaryEmail	N I F	
primaryName	primaryName	primaryName	N I F	
primaryPhone	primaryPhone	primaryPhone	N I F	In format “xxx-xxx-xxxx” or “xxxxxxxxxxx”
primaryExt	primaryExt	primaryExtension		
remarks	remarks	remarks		

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
reportableTime	reportableTime	reportableTime		In ISO 8601 format, without the timezone offset information. YYYY-MM-DDTHH:mm:ss
restoreMethodDesc	restoreMethodDescription	restoreMethodDesc		
rootCauseId	rootCauseId	rootCauseId	F	<ul style="list-style-type: none"> 1- Cable Damage - Cable Unlocated 2- Cable Damage - Digging Error 3- Cable Damage - Inadequate/No Notification 4- Cable Damage - Inaccurate/Incomplete Cable Locate 5- Cable Damage - Shallow Cable 6- Cable Damage - Other 7- Design - Firmware - Ineffective fault recovery or re-initialization action 8- Design - Firmware - Insufficient Software State Indications 9- Design - Firmware - Other 10- Design - Hardware - Inadequate Grounding Strategy 11- Design - Hardware - Poor Backplane or Pin Arrangement 12- Design - Hardware - Poor Card/Frame Mechanisms (latches 13- Design - Hardware - Other 14- Design - Software - Faulty Software Load - Office Data 15- Design - Software - Faulty Software Load - Program Data 16- Design - Software - Inadequate Defensive Checks 17- Design - Software - Ineffective Fault Recovery or Re-initialization Action

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				18- Design - Software - Other 19- Diversity Failure - External 20- Diversity Failure - Links 21- Diversity Failure - Power 22- Diversity Failure - Timing Equipment 23- Diversity Failure - Internal (Other) 24- Environment (External) - Earthquake 25- Environment (External) - Fire 26- Environment (External) - Lightning/Transient Voltage 27- Environment (External) - Storm - Water/Ice 28- Environment (External) - Storm - Wind/Trees 29- Environment (External) - Vandalism/Theft 30- Environment (External) - Vehicular Accident 31- Environment (External) - Other 32- Environment (Internal) - Cable Pressurization Failure 33- Environment (Internal) - Dirt 34- Environment (Internal) - Environmental System Failure (heat/humidity) 35- Environment (Internal) - Fire 36- Environment (Internal) - Fire Suppression (water) 37- Environment (Internal) - Manhole/Cable Vault Leak 38- Environment (Internal) - Roof/Air Conditioning Leak 39- Environment (Internal) - Other 40- Hardware Failure - Memory Unit Failure

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				41- Hardware Failure - Peripheral Unit Failure 42- Hardware Failure - Processor Community Failure 43- Hardware Failure - Other 44- Insufficient Data 45- Other/Unknown 46- Power Failure (Commercial and/or Back-up) - Battery Failure 47- Power Failure (Commercial and/or Back-up) - Extended Commercial Power Failure 48- Power Failure (Commercial and/or Back-up) - Generator Failure 49- Power Failure (Commercial and/or Back-up) - Inadequate/Missing Power Alarm 50- Power Failure (Commercial and/or Back-up) - Inadequate Site-Specific Power Conti 51- Power Failure (Commercial and/or Back-up) - Insufficient Response to Power Alarm 52- Power Failure (Commercial and/or Back-up) - Lack of Power Redundancy 53- Power Failure (Commercial and/or Back-up) - Lack of Routine Maintenance/Testing 54- Power Failure (Commercial and/or Back-up) - Overloaded/Undersized Power Equipment 55- Power Failure (Commercial and/or Back-up) - Other 56- Procedural - Other Vendor - Ad hoc Activities 57- Procedural - Other Vendor - Documentation/Procedures Out-of-Date

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				58- Procedural - Other Vendor - Documentation/Procedures Unavailable 59- Procedural - Other Vendor - Insufficient Supervision/Control or Employee Error 60- Procedural - Other Vendor - Insufficient Training 61- Procedural - Other Vendor - Other 62- Procedural - Service Provider - Documentation/Procedures Out-of-Date Unusable or 63- Procedural - Service Provider - Documentation/Procedures Unavailable/Unclear/Inc 64- Procedural - Service Provider - Inadequate Routine Maintenance/Memory Back-Up 65- Procedural - Service Provider - Insufficient Staffing/Support 66- Procedural - Service Provider - Insufficient Supervision/Control or Employee Err 67- Procedural - Service Provider - Insufficient Training 68- Procedural - Service Provider - Other 69- Procedural - System Vendor - Ad hoc Activities 70- Procedural - System Vendor - Documentation/Procedures Out-of-Date 71- Procedural - System Vendor - Documentation/Procedures Unavailable/Unclear/Incomp 72- Procedural - System Vendor - Insufficient Staffing/Support

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				73- Procedural - System Vendor - Insufficient Supervision/Control or Employee Error 74- Procedural - System Vendor - Insufficient Training 75- Procedural - System Vendor - Other 76- Simplex Condition - Non-service Affecting 77- Simplex Condition - Service Affecting 78- Traffic/System Overload - Common Channel Signaling Network Overload 79- Traffic/System Overload - Inappropriate/Insufficient Network Management (NM) con 80- Traffic/System Overload - Ineffective Engineering/Engineering Tools 81- Traffic/System Overload - Mass Calling - Focused/Diffuse Network Overload 82- Traffic/System Overload - Media-Stimulated Calling - Insufficient Notification 83- Traffic/System Overload - Other 84- Spare Not Available 85- Spare On Hand - Failed 86- Spare On Hand - Manufacturer Discontinued (MD) 87- Cable Damage/Malfunction - Aerial/Non-Buried 88- Cable Damage/Malfunction - Cable Malfunction 89- Environment (External) - Animal Damage 90- Environment (External) - Flood 91- Environment (External) - Ice/Storm 92- Hardware Failure - Circuit Pack/Card Failure-Processor

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				<p>93- Hardware Failure - Circuit Pack/Card Failure-Other</p> <p>94- Hardware Failure - Passive Devices</p> <p>95- Hardware Failure - Self-contained Device Failure</p> <p>96- Hardware Failure - Shelf/Slot Failure</p> <p>97- Hardware Failure - Software Storage Media Failure</p> <p>98- Insufficient Data - Cleared While Testing</p> <p>99- Insufficient Data - Non-Service Provider Personnel</p> <p>100- Insufficient Data - Outside Owned Network</p> <p>101- Insufficient Data - Under Investigation</p> <p>102- Power Failure (Commercial and/or Back-up) - Rectifier Failure</p> <p>103- Power Failure (Commercial and/or Back-up) - Breaker Tripped/Blown Fuses</p> <p>104- Power Failure (Commercial and/or Back-up) - Scheduled Activity - Software Upgrade</p> <p>105- Power Failure (Commercial and/or Back-up) - Scheduled Maintenance - Hardware Rep</p> <p>106- Power Failure (Commercial and/or Back-up) - Unidentified Power Surge</p> <p>107- Power Failure (Commercial and/or Back-up) - Inadequate back-up Power Equipment L</p> <p>109- Procedural - Other Vendor - Insufficient Staffing/Support</p> <p>110- Planned maintenance to upgrade the system</p> <p>111- Planned maintenance to fix known problems</p>

XML Upload	SOAP	REST	REQUIRED FOR	DESCRIPTION
				112- Planned maintenance - failed 113- Planned maintenance - went longer or worse than expected
secondaryAddress1	secondaryAddressLine1	secondaryAddressLine1		
secondaryAddress2	secondaryAddressLine2	secondaryAddressLine2		
secondaryAddress3	secondaryAddressLine3	secondaryAddressLine3		
secondaryEmail	secondaryEmail	secondaryEmail		
secondaryName	secondaryName	secondaryName		
secondaryPhone	secondaryPhone	secondaryPhone		In format "xxx-xxx-xxxx" or "xxxxxxxxxxx"
secondaryExt	secondaryExt	secondaryExtension		
serviceOtherDesc	serviceOtherDescription	serviceOtherDescription		
stateCd	stateCd	stateCd	N I F	Standard 2-Letter State designation, plus option of "MM" to denote "Multi-State".
timeZoneId	timezoneId	timezoneId	N I F	Note capitalization differences. 1- Eastern 2- Mountain 3- Pacific 4- Atlantic 5- Central 6- Alaskan 7- Hawaii-Aleutian 8- Guam 9- Other 10- Greenwich Mean Time (GMT)
withdrawnReason	withdrawnReason	withdrawnReason	W	

Appendix B - Transitioning from the Appian API

This version of the NORS SOAP API has been designed to minimize the impact on users of the prior API. The goal has been to make it so that the only change that developers need to make is to update the endpoint of the SOAP call. Nevertheless, there are a few minor differences between the Appian-based SOAP API (v.2.0) and the new one (v.3.0).

SOAP

NORS 2.0	NORS 3.0	Notes
True/False	true/false	Boolean fields in v.3 accepted "True" or 1, and "False" or 0. This has been retained, but if using the words, they must be provided in all lower case .
	<blank> values vs null	The v.4 API does not require every field in every call, only the ones necessary for the given outage type. The remaining fields are optional. As a general rule, providing a field with blank data will clear any previously-saved data in that field, but leaving out the field completely will cause the API to ignore any data stored in that field in the update.
Boolean defaults to ""	Boolean defaults to false	For true Boolean fields (displayed as checkboxes in the UI), the default value is now set to false. The API continues to accept entries of "" (blank), but this will be translated to false in the system.
<none>	Preemptive Authentication	The new API requires preemptive authentication. This is the default in SOAP UI (and can be set in the global preferences), but NORS 2.0 was not strict about this setting.

XML

NORS 2.0	NORS 3.0	Notes
<p>companyPassword entered on the XML Upload page</p>	<p>companyPassword integrated into <NORS_Outage></p>	<p>In v.3, the companyPassword was provided on the entry page of the XML Upload. In v.4, the companyPassword has been incorporated as a separate field in the XML data. This more closely ties the company password to the actual data record, and allows entry of multiple companies' outages in the same upload, for those who have inputter rights for more than one company.</p>