

TRUD API guide

Draft version 1
April 2019



Table of contents

Introduction.....	1
The API.....	1
The key.....	1
List all releases of an item.....	2
List the latest release of an item.....	4
Download the latest release of an item.....	4
Automation.....	5
On UNIX-like systems with bash.....	5
Download the latest release of an item.....	5
Download all releases of an item.....	6
On Windows with PowerShell.....	8
Download the latest release of an item.....	8
Download all releases of an item.....	9

Introduction

The TRUD FTP server is to be retired in the coming weeks. Its replacement is a web service that is secure (the transfers are over HTTPS) and helps you automate the downloading of releases. The web service also saves the NHS the cost of maintaining the FTP server.

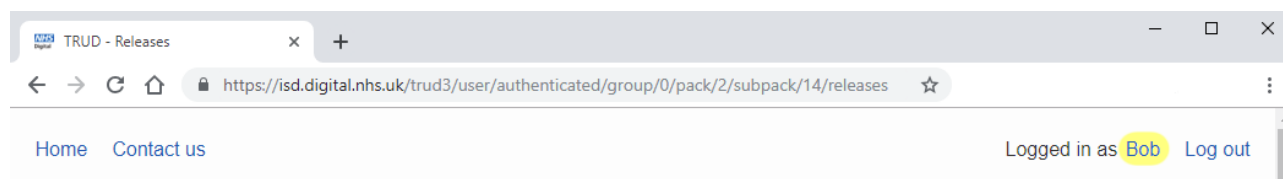
This is a draft document – please note that these new features are not yet available.

The API

The API comprises a RESTful API endpoint to list releases, and URLs to download release files directly.

The key

To access the API use the API key associated with your TRUD account. You can see this key by logging in to TRUD and viewing your account details – click your preferred name at the top-right of the page:



At the foot of your account details you'll see a box like this:

API key

db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0

Please do not share this key.

Changing the account password will generate a new API key, so if the key is compromised please change the account password.

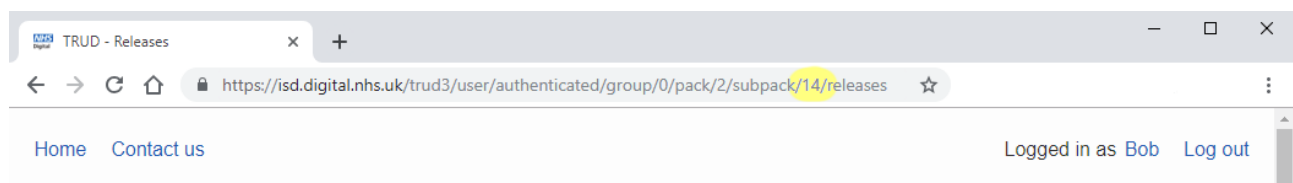
The [TRUD API guide \(PDF\)](#) describes how you can use this key.

Your key is a 40-character hexadecimal representation of a SHA-1 hash based on your account number, email address and password – if any of these change your key will change automatically.

Please don't share your key with others, in the same way you wouldn't share your account password. Downloads are logged against your key and hence your account, so a high number of downloads using the same key is a smoking gun! If you think someone else is using your key, simply change your password and the stolen key will be invalidated immediately.

List all releases of an item

In addition to your key you'll need the item number. You can find this by visiting the item releases page and examining the URL:



The item number follows the subpack URL component (TRUD veterans will know that items are referred to as “subpacks” internally, though this nomenclature sometimes leaks onto the pages).

On UNIX-like systems you can list all releases of item 14 like this (all on one line):

```
$ curl https://isd.digital.nhs.uk/trud3/api/v1/keys/db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0/items/14/releases
```

On Windows with PowerShell (all on one line):

```
> Invoke-RestMethod -Uri https://isd.digital.nhs.uk/trud3/api/v1/keys/db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0/items/14/releases
```

Throughout the rest of this document you'll see <API-KEY> instead of the full key to improve readability.

Here's the JSON response:

```
{
  "apiVersion" : "1",
  "releases" : [ {
    "id" : "DIAGIMG_26.0.0_20181001000001",
    "name" : "Clinical Imaging Procedures 26.0.0",
    "releaseDate" : "2018-10-01",
    "archiveFileUrl" :
    "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/nhs_diagimg_26.0.0_20181001000001.zip",
    "archiveFileName" : "nhs_diagimg_26.0.0_20181001000001.zip",
    "archiveFileSizeBytes" : 4456684,
  }
]
```

```
"archiveFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
"checksumFileUrl" :
"https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/
trud_nhs_diagimg_26.0.0_20181001000001.xml",
"checksumFileName" : "trud_nhs_diagimg_26.0.0_20181001000001.xml",
"checksumFileSizeBytes" : 174,
"checksumFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
"signatureFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/
26.0.0/trud_nhs_diagimg_26.0.0_20181001000001.xml.asc",
"signatureFileName" : "trud_nhs_diagimg_26.0.0_20181001000001.sig",
"signatureFileSizeBytes" : 484,
"signatureFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
"publicKeyFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/public-
keys/trud-public-key-2013-04-01.pgp",
"publicKeyFileName" : "trud-public-key-2013-04-01.pgp",
"publicKeyFileSizeBytes" : 1736,
"publicKeyId" : 6
}, {
" id" : "DIAGIMG_25.0.0_20180403000001",
" name" : "Clinical Imaging Procedures 25.0.0",
" releaseDate" : "2018-04-03",
" archiveFileUrl" :
"https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/25.0.0/
nhs_diagimg_25.0.0_20180403000001.zip",
" archiveFileName" : "nhs_diagimg_25.0.0_20180403000001.zip",
" archiveFileSizeBytes" : 5238156,
" archiveFileLastModifiedTimestamp" : "2018-03-28T11:40:23.000Z",
" checksumFileUrl" :
"https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/25.0.0/
trud_nhs_diagimg_25.0.0_20180403000001.xml",
" checksumFileName" : "trud_nhs_diagimg_25.0.0_20180403000001.xml",
" checksumFileSizeBytes" : 174,
" checksumFileLastModifiedTimestamp" : "2018-03-28T14:15:19.000Z",
" signatureFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/
25.0.0/trud_nhs_diagimg_25.0.0_20180403000001.xml.asc",
" signatureFileName" : "trud_nhs_diagimg_25.0.0_20180403000001.sig",
" signatureFileSizeBytes" : 484,
" signatureFileLastModifiedTimestamp" : "2018-03-28T14:15:19.000Z",
" publicKeyFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/public-
keys/trud-public-key-2013-04-01.pgp",
" publicKeyFileName" : "trud-public-key-2013-04-01.pgp",
" publicKeyFileSizeBytes" : 1736,
" publicKeyId" : 6
}, {
" id" : "DIAGIMG_24.0.0_20171002000001",
" name" : "DIAGIMG_24.0.0_20171002000001",
" releaseDate" : "2017-10-02",
" archiveFileUrl" :
"https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/24.0.0/NHS_DIAGIMG/
nhs_diagimg_24.0.0_20171002000001.zip",
" archiveFileName" : "nhs_diagimg_24.0.0_20171002000001.zip",
" archiveFileSizeBytes" : 5252366,
" archiveFileLastModifiedTimestamp" : "2017-10-02T12:58:21.000Z",
" checksumFileUrl" :
"https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/24.0.0/NHS_DIAGIMG/
trud_nhs_diagimg_24.0.0_20171002000001.xml",
" checksumFileName" : "trud_nhs_diagimg_24.0.0_20171002000001.xml",
" checksumFileSizeBytes" : 174,
" checksumFileLastModifiedTimestamp" : "2017-10-02T13:09:07.000Z",
" signatureFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/
24.0.0/NHS_DIAGIMG/trud_nhs_diagimg_24.0.0_20171002000001.xml.asc",
" signatureFileName" : "trud_nhs_diagimg_24.0.0_20171002000001.sig",
" signatureFileSizeBytes" : 484,
" signatureFileLastModifiedTimestamp" : "2017-10-02T13:09:19.000Z",
" publicKeyFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/public-
keys/trud-public-key-2013-04-01.pgp",
" publicKeyFileName" : "trud-public-key-2013-04-01.pgp",
" publicKeyFileSizeBytes" : 1736,
" publicKeyId" : 6
} ],
"HttpStatus" : 200,
```

```
  "message" : "OK"
}
```

The response contains download URLs for all files of all releases of the item. These URLs form the basis of the automation scripts found later in this document.

The releases are listed from latest to oldest, just as they are on the item releases page.

List the latest release of an item

This is essentially the same as listing all releases of an item, but only the latest release is returned in the response.

On UNIX-like systems you can list the latest release of item 14 like this:

```
$ curl https://isd.digital.nhs.uk/trud3/api/v1/keys/<API-KEY>/items/14/releases?latest
```

On Windows with PowerShell:

```
> Invoke-RestMethod -Uri https://isd.digital.nhs.uk/trud3/api/v1/keys/<API-KEY>/items/14/releases?latest
```

Here's the JSON response with the download URLs highlighted:

```
{
  "apiVersion" : "1",
  "releases" : [ {
    "id" : "DIAGIMG_26.0.0_20181001000001",
    "name" : "Clinical Imaging Procedures 26.0.0",
    "releaseDate" : "2018-10-01",
    "archiveFileUrl" :
    "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/nhs_diagimg_26.0.0_20181001000001.zip",
    "archiveFileName" : "nhs_diagimg_26.0.0_20181001000001.zip",
    "archiveFileSizeBytes" : 4456684,
    "archiveFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
    "checksumFileUrl" :
    "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/trud_nhs_diagimg_26.0.0_20181001000001.xml",
    "checksumFileName" : "trud_nhs_diagimg_26.0.0_20181001000001.xml",
    "checksumFileSizeBytes" : 174,
    "checksumFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
    "signatureFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/trud_nhs_diagimg_26.0.0_20181001000001.xml.asc",
    "signatureFileName" : "trud_nhs_diagimg_26.0.0_20181001000001.sig",
    "signatureFileSizeBytes" : 484,
    "signatureFileLastModifiedTimestamp" : "2019-03-31T07:08:24.000Z",
    "publicKeyFileUrl" : "https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/public-keys/trud-public-key-2013-04-01.pgp",
    "publicKeyFileName" : "trud-public-key-2013-04-01.pgp",
    "publicKeyFileSizeBytes" : 1736,
    "publicKeyId" : 6
  } ],
  "HttpStatus" : 200,
  "message" : "OK"
}
```

Download the latest release of an item

The responses above include the URLs of the release files. Use these URLs to download the files.

On UNIX-like systems you can download the latest release of item 14 like this:

```
$ curl https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/nhs_diagimg_26.0.0_20181001000001.zip -o nhs_diagimg_26.0.0_20181001000001.zip
```

On Windows with PowerShell:

```
> Invoke-WebRequest -Uri
https://isd.digital.nhs.uk/api/v1/keys/<API-KEY>/files/DIAGIMG/26.0.0/
nhs_diagimg_26.0.0_20181001000001.zip -OutFile nhs_diagimg_26.0.0_20181001000001.zip
```

Automation

We've written bash and PowerShell scripts for you to use as a basis for your own automation scripts. Links to these scripts are given below, followed by the scripts themselves for reference.

On UNIX-like systems with bash

Download the latest release of an item

Example usage:

```
$ ./download-latest-release-v1.sh db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0 14
```

[download-latest-release-v1.sh](#)

```
#!/bin/bash

trud_host=isd.digital.nhs.uk
curl_options=--silent

# Helper functions.

program_name=$(basename "$0" .sh)

usage_help_exit() {
    echo "\
Usage: $program_name KEY ITEM
Download the latest release of ITEM via the account having API key KEY.

    --help    display this help and exit

Report bugs to information.standards@nhs.net"
    exit 0
}

usage_error_exit() {
    (($# > 0)) && error "$@"
    echo "Try $program_name --help for more information." >&2
    exit 2
}

error() {
    echo "$program_name: error: $@" >&2
}

error_exit() {
    error "$@"
    exit 1
}

# Parse the command line.
t=$(getopt --name "$program_name" --options d --longoptions dev,help -- "$@")
(($? == 0)) || error_exit "getopt failed."
eval set -- "$t"

while true; do
    case "$1" in
        -d|--dev)
```

```

        curl_options="--insecure"
        trud_host=localhost
        shift
        ;;
    --help)
        usage_help_exit
        ;;
    --)
        break
        ;;
    *)
        usage_error_exit
        ;;
esac
done
shift

# Validate the arguments.
(($# < 2)) && usage_error_exit "too few arguments."
((2 < $#)) && usage_error_exit "too many arguments."
key="$1"
((${#key} < 40)) && usage_error_exit "API key too short (expected 40 characters, found $
{#key})."
((40 < ${#key})) && usage_error_exit "API key too long (expected 40 characters, found $
{#key})."
[[ $key =~ ^[a-f0-9]{40}$ ]] || usage_error_exit "API key does not consist entirely of
hexadecimal characters."
item="$2"
[[ -z $item ]] && usage_error_exit "item number is empty."
[[ $item =~ ^[0-9]+$ ]] || usage_error_exit "item number \"$item\" is not a natural
number."

# Check for dependencies.
hash curl 2>/dev/null || error_exit "curl is not available."
hash jq 2>/dev/null || error_exit "jq is not available."

# Get information about the latest release.
response=$(curl $curl_options --silent
https://$trud_host/trud3/api/v1/keys/$key/items/$item/releases?latest)
message=$(jq --raw-output '.message' <<< $response)
[[ -z $message ]] && error_exit "no response."
[[ $message != OK ]] && error_exit "$message"
release=$(jq --compact-output '.releases[0]' <<< $response)

# Get the archive file URL and archive file name for the release.
archive_file_url=$(jq --raw-output '.archiveFileUrl' <<< $release)
[[ -z $archive_file_url ]] && error_exit "archive file URL is empty."
archive_file_name=$(jq --raw-output '.archiveFileName' <<< $release)
[[ -z $archive_file_name ]] && error_exit "archive file name is empty."

# Download the release.
exec curl $curl_options "$archive_file_url" --output "$archive_file_name"

```

Download all releases of an item

Example usage:

```
$ ./download-all-releases-v1.sh db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0 14
```

[download-all-releases-v1.sh](#)

```
#!/bin/bash
```

```
trud_host=isd.digital.nhs.uk
curl_options=--silent
```

```
# Helper functions.
```

```

program_name=$(basename "$0" .sh)

usage_help_exit() {
    echo "\
Usage: $program_name KEY ITEM
Download all releases of ITEM via the account having API key KEY.

    --help    display this help and exit

Report bugs to information.standards@nhs.net"
    exit 0
}

usage_error_exit() {
    (($# > 0)) && error "$@"
    echo "Try $program_name --help for more information." >&2
    exit 2
}

error() {
    echo "$program_name: error: $@" >&2
}

error_exit() {
    error "$@"
    exit 1
}

# Parse the command line.
t=$(getopt --name "$program_name" --options d --longoptions dev,help -- "$@")
(($? == 0)) || error_exit "getopt failed."
eval set -- "$t"

while true; do
    case "$1" in
        -d|--dev)
            curl_options="--insecure"
            trud_host=localhost
            shift
            ;;
        --help)
            usage_help_exit
            ;;
        --)
            break
            ;;
        *)
            usage_error_exit
            ;;
    esac
done
shift

# Validate the arguments.
(($# < 2)) && usage_error_exit "too few arguments."
((2 < $#)) && usage_error_exit "too many arguments."
key="$1"
((${#key} < 40)) && usage_error_exit "API key too short (expected 40 characters, found ${#key})."
((40 < ${#key})) && usage_error_exit "API key too long (expected 40 characters, found ${#key})."
[[ $key =~ ^[a-f0-9]{40}$ ]] || usage_error_exit "API key does not consist entirely of hexadecimal characters."
item="$2"
[[ -z $item ]] && usage_error_exit "item number is empty."
[[ $item =~ ^[0-9]+$ ]] || usage_error_exit "item number \"$item\" is not a natural number."

# Check for dependencies.

```

```

hash curl 2>/dev/null || error_exit "curl is not available."
hash jq 2>/dev/null || error_exit "jq is not available."
hash tac 2>/dev/null || error_exit "tac is not available."

# Get information about all releases.
response=$(curl $curl_options --silent
https://$strud_host/trud3/api/v1/keys/$key/items/$item/releases)
message=$(jq --raw-output '.message' <<< $response)
[[ -z $message ]] && error_exit "no response."
[[ $message != OK ]] && error_exit "$message"

# For each release.
declare -i release_count
release_count=0
while read release
do
    # Count the number of releases.
    release_count+=1

    # Get the archive file URL and archive file name for this release.
    archive_file_url=$(jq --raw-output '.archiveFileUrl' <<< $release)
    [[ -z $archive_file_url ]] && error_exit "archive file URL is empty."
    archive_file_name=$(jq --raw-output '.archiveFileName' <<< $release)
    [[ -z $archive_file_name ]] && error_exit "archive file name is empty."

    # Download the release.
    curl $curl_options "$archive_file_url" --output "$archive_file_name"

done < <(jq --compact-output '.releases[]' <<< $response|tac)
(($release_count > 0)) || error_exit "no releases found."

exit 0

```

On Windows with PowerShell

Download the latest release of an item

Example usage:

```
> .\download-latest-release-v1.ps1 -Key db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0 -Item 14
```

[download-latest-release-v1.ps1](#)

```

param (
    [Parameter(Mandatory=$true)]
    [ValidatePattern("[0-9a-f]{40}$")]
    [String]
    $Key,

    [Parameter(Mandatory=$true)]
    [ValidateRange(0, 9999)]
    [Int]
    $Item
)

$strud_host = "isd.digital.nhs.uk"

# The progress bar can significantly impact performance.
# See https://github.com/PowerShell/PowerShell/issues/2138
$ProgressPreference = 'SilentlyContinue'

try {

```



```

    $response = Invoke-RestMethod -Uri
https://$strud_host/trud3/api/v1/keys/$Key/items/$Item/releases
} catch {
    # We don't need the stack trace.
    #Write-Error $_.Exception.Message
    [Console]::ForegroundColor = 'red'
    [Console]::Error.WriteLine($_.Exception.Message)
    exit
}

foreach ($release in $response.releases) {
    Invoke-WebRequest -Uri $release.archiveFileUrl -OutFile $release.archiveFileName
}

```

Download all releases of an item

Example usage:

```
> .\download-all-releases-v1.ps1 -Key db66ec5e77c4eb0cd1ef90a9b534a6446e2a9a0 -Item 14
```

[download-all-releases-v1.ps1](#)

```

param (

    [Parameter(Mandatory=$true)]
    [ValidatePattern("[0-9a-f]{40}$")]
    [String]
    $Key,

    [Parameter(Mandatory=$true)]
    [ValidateRange(0, 9999)]
    [Int]
    $Item
)

$strud_host = "isd.digital.nhs.uk"

# The progress bar can significantly impact performance.
# See https://github.com/PowerShell/PowerShell/issues/2138
$ProgressPreference = 'SilentlyContinue'

try {
    $response = Invoke-RestMethod -Uri
https://$strud_host/trud3/api/v1/keys/$Key/items/$Item/releases
} catch {
    # We don't need the stack trace.
    #Write-Error $_.Exception.Message
    [Console]::ForegroundColor = 'red'
    [Console]::Error.WriteLine($_.Exception.Message)
    exit
}

foreach ($release in $response.releases) {
    Invoke-WebRequest -Uri $release.archiveFileUrl -OutFile $release.archiveFileName
}

```