tr

Interactive flipping command line utilities

[Install] [Commands] [Usage] [Contributing]

Install

- Install

Usage

```
$ tr [OPTION] [-p PROMPT] [-t TITLE] [-c COLOR] [-b BG_COLOR] [-f FONT_FAMILY] [-s FONT_SIZE] [-w WIDTH] [-h HEIGHT] [-i INTERFACE] [-u CREDENTIAL] [-S] [-C CERT] [-K KEY] [-A] [-O] [-T] [-I INDEX] [-P PING_INTERVAL] [-m MAX_CLIENTS] [-o] [-q] [-v] [-d] [-6] [-R] [-D] [COMMAND] [ARGS...]
```

Commands

- tr autocomplete [SHELL]
- tr bundle [DATASET]
- tr dataset clear [DATASET]
- tr dataset data [DATASET]
- tr dataset delete [DATASET]
- tr dataset list
- tr dataset search [DATASET]
- tr dataset set-index [DATASET]
- tr dataset set-mdindex [DATASET]
- tr dataset unset-index [DATASET]
- tr dataset unset-mdindex [DATASET]
- tr dbrestore [-d DATABASE] [-f FILE]
- tr dbsnap [-d DATABASE] [-f FILE]
- tr help
- tr import [-d DATABASE] [-f FILE] [-t TABLE] [-s SEPARATOR] [-h]
- tr query [-d DATABASE] [-f FORMAT] [-o OUTPUT] [-h] [QUERY]
- tr schema [-d DATABASE] [-t TABLE] [-h]
- tr tables [-d DATABASE] [-h]
- tr version

tr autocomplete [SHELL]

display autocomplete installation instructions

```
USAGE
  tr autocomplete [SHELL]

ARGUMENTS
  SHELL  shell type

FLAGS
  -h, --help  output usage. Defined under Cygnosis displaying instructions!

EXAMPLE
  $ tr autocomplete
  $ tr autocomplete bash
  $ tr autocomplete zsh
  $ tr autocomplete --profile
```

See code: src/flipper-autocomplete

tr bundle [DATASET]

Compress a bundle that contains case duplicate json filenames in a case sensitive filesystem

```
USAGE
  $ tr bundle dataset

FLAGS
  -f, --file=file  bundle filename

DESCRIPTION
  Bundle dataset bundle to archive by standard out

EXAMPLE
  $ tr bundle --help -f my-bundle.tgz > my-dataset-bundle.tgz
```

See code: lib/commands/bundle/index.js

tr dataset:clear [DATASET]

clear dataset data within Elasticsearch.

```
USAGE
  $ tr dataset:clear [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to clear

OPTIONS
  -h, --help=<value>                  the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -i, --interactive                   require authorization before clearing
  -n, --nocache                       disable verbose output
  -q, --quiet                         the ttl/time limit to connect to
  -r, --refresh                       enable verbose output
  -v, --verbose                       clear all datasets
  --version                           torrent to Imperatorcis while running the command
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex:--name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Clear dataset data within Elasticsearch.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:clear --help
  $ tr dataset:clear --wipeMAGNEL.PROMI=<value>
  $ tr dataset:clear -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/clear.js

tr dataset:data [DATASET]

view dataset data

```
USAGE
  $ tr dataset:data [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -i, --interactive                   require authorization before clearing
  -n, --nocache                       disable verbose output
  -q, --quiet                         the ttl/time limit to connect to
  -r, --refresh                       enable verbose output
  -v, --verbose                       clear all datasets
  --version                           torrent to Imperatorcis while running the command
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex:--name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Update dataset data within Elasticsearch.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:data --help
  $ tr dataset:data -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/data.js

tr dataset:delete [DATASET]

delete dataset data within Elasticsearch.

```
USAGE
  $ tr dataset:delete [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to delete

OPTIONS
  -h, --help=<value>                  the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -i, --interactive                   require authorization before clearing
  -n, --nocache                       disable verbose output
  -q, --quiet                         the ttl/time limit to connect to
  -r, --refresh                       enable verbose output
  -v, --verbose                       clear all datasets
  --version                           torrent to Imperatorcis while running the command
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex:--name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Delete dataset data within Elasticsearch.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:delete --help
  $ tr dataset:delete -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/delete.js

tr dataset:list

display a table of the datasets available in the current user

```
USAGE
  $ tr dataset:list

FLAGS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       clear all datasets
  -x, --index                         Long format
  -z, --noheader                      enable verbose output
  --csv                               Format the torrent to connect as
  --error                             torrent to Imperatorcis while running the command
  --extended                          Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --filter=<value>                    Filter property by partial string matching, ex: --name=foo
  --no-header                         Hide table header from output
  --no-truncate                       Do not truncate output to fit screen
  --output=csv|json|yaml              output in a more machine friendly format
  --sort=<value>                      Property to sort by (prepend '-' for descending)
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.

DESCRIPTION
  Table a host to an ELI APP and datasource requests for service-side logging

EXAMPLE
  $ tr dataset:list
```

See code: lib/commands/dataset/list.js

tr dataset:search [DATASET]

execute an Elasticsearch query against a dataset

```
USAGE
  $ tr dataset:search [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       clear all datasets
  -x, --index                         Long format
  -z, --noheader                      enable verbose output
  --csv                               Format the torrent to connect as
  --error                             torrent to Imperatorcis while running the command
  --extended                          Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --filter=<value>                    Filter property by partial string matching, ex: --name=foo
  --no-header                         Hide table header from output
  --no-truncate                       Do not truncate output to fit screen
  --output=csv|json|yaml              output in a more machine friendly format
  --sort=<value>                      Property to sort by (prepend '-' for descending)
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.

DESCRIPTION
  Table a host to an ELI APP and datasource requests for service-side logging

EXAMPLE
  $ tr dataset:search --help
```

See code: lib/commands/dataset/search.js

tr dataset:set-index [DATASET]

set a dataset's Elasticsearch index pointer

```
USAGE
  $ tr dataset:set-index [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       require authorization before updating
  -x, --index                         clear all datasets
  -z, --noheader                      the ttl/time limit to connect to
  --csv                               torrent to Imperatorcis while running the command
  --error                             update all datasets
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex: --name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Set dataset's search index pointer.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:set-index --help
  $ tr dataset:set-index -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/set-index.js

tr dataset:set-mdindex [DATASET]

sets dataset indices in bulk

```
USAGE
  $ tr dataset:set-mdindex [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       authenticate with an Infomax API token
  -x, --index                         the ttl/time limit to connect to
  -z, --noheader                      the ttl/time limit to connect to
  --csv                               enable verbose output
  --error                             sets the index as all datasets
  --wipeMAGNEL.PROMI                  torrent to Imperatorcis while running the command
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex: --name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Set dataset's search index pointer.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:set-mdindex --help
  $ tr dataset:set-mdindex -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/set-mdindex.js

tr dataset:unset-index [DATASET]

sets dataset indices in bulk

```
USAGE
  $ tr dataset:unset-index [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       authenticate with an Infomax API token
  -x, --index                         the ttl/time limit to connect to
  -z, --noheader                      the ttl/time limit to connect to
  --csv                               enable verbose output
  --error                             sets the index as all datasets
  --wipeMAGNEL.PROMI                  torrent to Imperatorcis while running the command
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex: --name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Set dataset's search index pointer.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:unset-index --help
  $ tr dataset:unset-index -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/unset-index.js

tr dataset:unset-mdindex [DATASET]

sets dataset indices in bulk

```
USAGE
  $ tr dataset:unset-mdindex [DATASET]

ARGUMENTS
  DATASET  data set or mock datasets to update

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --verbose                       authenticate with an Infomax API token
  -x, --index                         the ttl/time limit to connect to
  -z, --noheader                      the ttl/time limit to connect to
  --csv                               enable verbose output
  --error                             sets the index as all datasets
  --wipeMAGNEL.PROMI                  torrent to Imperatorcis while running the command
  --indexname=<value>                 data index values to add to the header (e.g.,x-GUARDIAN_ID)
  --mdindex                           Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --noheader                          skip printing the header
  --query=<value>                     Filter property by partial string matching, ex: --name=foo
  --sort=<value>                      Sort property by ascending or descending

DESCRIPTION
  Set dataset's search index pointer.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:unset-mdindex --help
  $ tr dataset:unset-mdindex -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/unset-mdindex.js

tr dataset:query [DATASOURCE]

execute a query against a datasource

```
USAGE
  $ tr dataset:query [DATASOURCE]

ARGUMENTS
  DATASOURCE  the datasource to query

OPTIONS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --file=file                     output file/name
  -x, --interactive                   the query to run
  -z, --noheader                      authenticate with an Infomax API token
  --csv                               the ttl/time limit to connect to
  --error                             the ttl/time limit to connect to
  --extended                          enable verbose output
  --filter=<value>                    store the datasource while running the command
  --no-header                         Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --no-truncate                       Filter property by partial string matching, ex: --name=foo
  --output=csv|json|yaml              List of JSON Elasticsearch datasource expressions
  --sort=<value>                      Sort property by ascending or descending
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.

DESCRIPTION
  Query a datasource.

  Authentication credentials to be used for the duration of the command. If only the username is supplied then it will prompt for a password.

EXAMPLE
  $ tr dataset:query --help
  $ tr dataset:query -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dataset/query.js

tr dbrestore

Add a db or Infomax database

```
USAGE
  $ tr dbrestore

FLAGS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --file=file                     output file/name
  -x, --interactive                   the query to run
  -z, --noheader                      authenticate with an Infomax API token
  --csv                               the ttl/time limit to connect to
  --error                             the ttl/time limit to connect to
  --extended                          enable verbose output
  --filter=<value>                    store the datasource while running the command
  --no-header                         Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --no-truncate                       Filter property by partial string matching, ex: --name=foo
  --output=csv|json|yaml              List of JSON Elasticsearch datasource expressions
  --sort=<value>                      Sort property by ascending or descending
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.

DESCRIPTION
  Restore a database from a backup bundle

EXAMPLE
  $ tr dbrestore --help
  $ tr dbrestore -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dbrestore/index.js

tr dbsnap

Add a db or Infomax database

```
USAGE
  $ tr dbsnap

FLAGS
  -h, --help=<value>                  one or more header name/values (e.g.,x-GUARDIAN_ID)
  -n, --nocache=<value>               the session to use when executing
  -u, --configFile=<value>            the path to a user config file for command flags. config file values will override
                                      session and flag values.
  -v, --file=file                     output file/name
  -x, --interactive                   the query to run
  -z, --noheader                      authenticate with an Infomax API token
  --csv                               the ttl/time limit to connect to
  --error                             the ttl/time limit to connect to
  --extended                          enable verbose output
  --filter=<value>                    store the datasource while running the command
  --no-header                         Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.
  --no-truncate                       Filter property by partial string matching, ex: --name=foo
  --output=csv|json|yaml              List of JSON Elasticsearch datasource expressions
  --sort=<value>                      Sort property by ascending or descending
  --wipeMAGNEL.PROMI                  Authentication credentials to be used for the duration of the command. If only the
                                      username is supplied then it will prompt for a password.

DESCRIPTION
  Snapshot a database

EXAMPLE
  $ tr dbsnap --help
  $ tr dbsnap -i -n -q -r -v -d -6 -R -D
```

See code: lib/commands/dbsnap/index.js

tr help

Display help for tr.

```
USAGE
  $ tr help [COMMAND] [-n]

ARGUMENTS
  COMMAND  command to show help for

FLAGS
  -n, --nested-commands  include all nested commands in the output

DESCRIPTION
  Display help for tr.
```

See code: src/commands/help.ts

tr import

Import a dataset from a directory

```
USAGE
  $ tr import [-d DATABASE] [-f FILE] [-t TABLE] [-s SEPARATOR] [-h]

FLAGS
  -d, --database=DATABASE  [default: :memory:] database file path
  -f, --file=FILE          [default: -] input file path
  -h, --help               show this help message and exit
  -s, --separator=SEPARATOR  [default: ,] column separator
  -t, --table=TABLE        [default: data] table name

DESCRIPTION
  Import a dataset from a directory

  Imports all .csv files in the specified directory into a SQLite database.
  Each file's basename (without extension) becomes the table name.
  Use --table to override the table name when importing a single file.

EXAMPLES
  $ tr import ./data/
  $ tr import ./data/ -d mydb.db
  $ tr import ./data/ -d mydb.db -t users
  $ tr import ./data/ -s ';'
```

See code: lib/commands/import.ts

tr query

Execute arbitrary SQL statements against a database.

```
USAGE
  $ tr query [-d DATABASE] [-f FORMAT] [-o OUTPUT] [-h] [QUERY]

ARGUMENTS
  QUERY  SQL query to execute (reads from stdin if omitted)

FLAGS
  -d, --database=DATABASE  [default: :memory:] database file path
  -f, --format=FORMAT      [default: table] output format
  -h, --help               show this help message and exit
  -o, --output=OUTPUT      [default: -] output file path

DESCRIPTION
  Execute arbitrary SQL statements against a database.

  Runs SQL queries against a SQLite database. The database file is created if it
  does not exist. Use :memory: (default) for a temporary in-memory database.

EXAMPLES
  $ tr query "SELECT * FROM users"
  $ tr query -d mydb.db "SELECT count(*) FROM orders"
  $ tr query -f csv -o results.csv "SELECT * FROM users"
  $ tr query -f json "SELECT * FROM users WHERE id = 1"
  $ echo "SELECT 1 + 1" | tr query
```

See code: src/commands/query.ts

tr schema

Show the schema for tables in a database.

```
USAGE
  $ tr schema [-d DATABASE] [-t TABLE] [-h]

FLAGS
  -d, --database=DATABASE  [default: :memory:] database file path
  -h, --help               show this help message and exit
  -t, --table=TABLE        table name (omit for all tables)

DESCRIPTION
  Show the schema for tables in a database.

EXAMPLES
  $ tr schema -d mydb.db
  $ tr schema -d mydb.db -t users
```

See code: src/commands/schema.ts

tr tables

List tables in a database.

```
USAGE
  $ tr tables [-d DATABASE] [-h]

FLAGS
  -d, --database=DATABASE  [default: :memory:] database file path
  -h, --help               show this help message and exit

DESCRIPTION
  List tables in a database.

EXAMPLES
  $ tr tables -d mydb.db
```

See code: src/commands/tables.ts

tr version

```
USAGE
  $ tr version [--json] [--verbose]

FLAGS
  --json     Format output as json.
  --verbose  Show additional information about the CLI.

DESCRIPTION
  Show CLI version.
```

See code: src/commands/version.ts

```
  -f, --file=file                   bundle filename

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --buffer-size=buffer-size         [default: 819200] upload chunk size

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --dir=dir                         import from a bundle directory

  --nes=nes                         nes websocket authentication token

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

DESCRIPTION
  Restores an Informer tenant from a backup bundle. The bundle is read from standard input unless the file flag
  is specified. Use the verbose option to view restore progress.

  Note: You must log in again after restoring a tenant.

EXAMPLES
  ir db:restore < backup.tgz
  ir db:restore -v -f backup.tgz
  ir db:backup -v -U blue | ir db:restore -v -U green
```

See code: lib/commands/db/restore.js

`ir defaults`

sets default parameters for new sessions

```
USAGE
  $ ir defaults

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --nes=nes                         nes websocket authentication token

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

EXAMPLE
  ir defaults --url https://demo.informer.cloud --tenant-header x-tenant
```

See code: lib/commands/defaults.js

`ir deploy [DATASET]`

deploys content from the manager tenant to a child tenant

```
USAGE
  $ ir deploy [DATASET]

ARGUMENTS
  DATASET  one or more datasets to deploy

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --activity                        copy activity

  --all                             deploys all content

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --dataset=dataset                 [default: ] deploys a dataset and its dependencies

  --datasource=datasource           [default: ] deploys a datasource and its content

  --json                            deploys content using the JSON format

  --nes=nes                         nes websocket authentication token

  --[no-]overwrite                  overwrites a record if it exists

  --replace-content                 replace all content (reports, datasets, queries, folders, tags)

  --replace-datasets                replace datasets

  --replace-folders                 replace folders

  --replace-queries                 replace queries

  --replace-reports                 replace reports

  --replace-tags                    replace tags

  --report=report                   [default: ] deploys a report and its dependencies

  --tag=tag                         [default: ] deploys all content for a tag

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --to=to                           the destination tenant to deploy to

  --to-all                          deploy to all tenants

  --tokens                          copy tokens

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

DESCRIPTION


EXAMPLES
  ir deploy --to tenant1 tenant2
  ir deploy --report test_report --to tenant1
  ir deploy --tag production --to-all
```

See code: lib/commands/deploy.js

`ir export`

back up an Informer instance.

```
USAGE
  $ ir export

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -f, --file=file                   output filename

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --dataset=dataset                 [default: ] exports a dataset and its dependencies

  --datasource=datasource           [default: ] exports a datasource, its metadata, and (optionally) its associated
                                    content

  --dir=dir                         export to a bundle directory

  --include-all                     includes all users and content associated with an exported datasource, user or team

  --include-datasets                includes datasets associated with an exported datasource, user or team

  --include-queries                 includes queries associated with an exported datasource, user or team

  --include-reports                 includes reports associated with an exported datasource, user or team

  --include-users                   includes all users associated with an exported team

  --json                            export Informer metadata in json format

  --nes=nes                         nes websocket authentication token

  --report=report                   [default: ] exports a report and its dependencies

  --tag=tag                         [default: ] exports all content for a tag

  --team=team                       [default: ] exports a team and (optionally) its users and content

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

  --user=user                       [default: ] exports a user and (optionally) their content

DESCRIPTION
  Create a backup bundle of an Informer tenant. The output of the bundle is written to standard out unless the file
  flag is provided. Use the verbose option to view backup progress.

EXAMPLES
  ir db:backup -vf backup.tgz
  ir db:backup > backup.tgz
  ir db:backup -v -U blue | ir db:restore -v -U green
```

See code: lib/commands/export.js

`ir get PATH [ARGS...]`

Gets a REST resource

```
USAGE
  $ ir get PATH [ARGS...]

ARGUMENTS
  PATH     the path to get
  ARGS...  additional query string args [arg name]==[arg value]

OPTIONS
  -H, --header=header                     one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session                   the session to use when connecting

  -c, --config=config                     the path to a json config file for command flags. config file values will override
                                          session and flag values

  -k, --token=token                       authenticate with an Informer API token

  -t, --tenant=tenant                     the Informer tenant to connect as

  -u, --url=url                           the Informer URL to connect to

  -v, --verbose                           enable verbose output

  --as=as                                 tenant to impersonate while running the command

  --auth=USER[:PASS]                      Authentication credentials to be used for the duration of the command. If only the
                                          username is supplied then ir will prompt for a password

  --cookie=cookie                         one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --indent=indent                         [default: 2] pretty format indentation

  --line-number-pad=line-number-pad       pretty output line number formatting

  --line-numbers                          pretty format line numbers

  --nes=nes                               nes websocket authentication token

  --pretty                                display formatted json results

  --tenant-header=tenant-header           the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized          rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id                     Adds a trace id to all XHR and websocket requests for server-side logging

  --websocket                             use a websocket

DESCRIPTION
  Issues an HTTP get to the currenet Informer session and displays a formatted
  JSON response

EXAMPLES
  ir get /api/me --pretty
  ir get /api/datasources sort==name
```

See code: lib/commands/get.js

`ir help [COMMAND]`

display help for ir

```
USAGE
  $ ir help [COMMAND]

ARGUMENTS
  COMMAND  command to show help for

OPTIONS
  --all  see all commands in CLI
```

See code: @oclif/plugin-help

`ir impersonate USERNAME`

impersonates another user

```
USAGE
  $ ir impersonate USERNAME

ARGUMENTS
  USERNAME  the user to impersonate

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --name=name                       the new session name

  --nes=nes                         nes websocket authentication token

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

EXAMPLES
  ir impersonate joe
  ir impersonate batman -t wayne-enterprises
```

See code: lib/commands/impersonate.js

`ir import`

import content from a bundle file

```
USAGE
  $ ir import

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -f, --file=file                   bundle filename

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --buffer-size=buffer-size         [default: 819200] upload chunk size

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --dir=dir                         import from a bundle directory

  --nes=nes                         nes websocket authentication token

  --[no-]overwrite                  overwrites a record if it exists

  --replace-content                 replace all content (reports, datasets, queries, folders, tags)

  --replace-datasets                replace datasets

  --replace-folders                 replace folders

  --replace-queries                 replace queries

  --replace-reports                 replace reports

  --replace-tags                    replace tags

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

DESCRIPTION
  Imports Informer content from a bundle file. The bundle is read from standard input unless the file flag
  is specified. Use the verbose option to view restore progress.

EXAMPLES
  ir import < backup.tgz
  ir import -v -f backup.tgz
```

See code: lib/commands/import.js

`ir info [DATASET]`

displays system version and build information

```
USAGE
  $ ir info [DATASET]

ARGUMENTS
  DATASET  zero or more datasets to update

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --nes=nes                         nes websocket authentication token

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

DESCRIPTION


EXAMPLE
  ir info
```

See code: lib/commands/info.js

`ir ingest:add DATASET`

adds a new ingest task to the queue

```
USAGE
  $ ir ingest:add DATASET

ARGUMENTS
  DATASET  the dataset to ingest

OPTIONS
  -i, --input=input        a runtime parameter
  -t, --tenant=tenant      [default: manager] the tenant containing the dataset to refresh
  --db=db                  [default: ingest] the name of the ingest database
  --end=end                the interval end date
  --interval=interval      an ISO8601 duration to use as a time interval for slicing up the start/end range
  --[no-]stage             stages the ingest tasks so that they may be reviewed before running
  --start=start            the interval start date

EXAMPLES
  ir ingest:add admin:northwind-orders
  ir ingest:add -t wayne-enterprises admin:crimes -i status=ACTIVE
```

See code: lib/commands/ingest/add.js

`ir ingest:clear`

clears finished tasks from the ingest database

```
USAGE
  $ ir ingest:clear

OPTIONS
  -d, --dataset=dataset  clear finished tasks for this dataset
  -t, --tenant=tenant    clear finished tasks for this tenant
  --db=db                [default: ingest] the name of the ingest database

DESCRIPTION
  Discards empty tasks from the ingest database. Shorthand for "ir ingest:delete --complete"

EXAMPLES
  ir ingest:clear
  ir ingest:clear --t wayne-enterprises
```

See code: lib/commands/ingest/clear.js

`ir ingest:delete [IDS]`

deletes one or more tasks from the ingest database

```
USAGE
  $ ir ingest:delete [IDS]

ARGUMENTS
  IDS  the ids of tasks to delete

OPTIONS
  -d, --dataset=dataset  delete tasks for a dataset
  -t, --tenant=tenant    delete tasks for a tenant
  --all                  delete all tasks
  --[no-]complete        delete completed tasks
  --db=db                [default: ingest] the name of the ingest database
  --[no-]errors          delete tasks with errors
  --[no-]staged          delete staged tasks
  --[no-]started         delete running tasks
```

See code: lib/commands/ingest/delete.js

`ir ingest:list [IDS]`

lists some or all of the commands in the ingest database

```
USAGE
  $ ir ingest:list [IDS]

ARGUMENTS
  IDS  the ids of tasks to delete

OPTIONS
  -d, --dataset=dataset
  -t, --tenant=tenant
  -x, --extended              show extra columns
  --all
  --columns=columns           only show provided columns (comma-separated)
  --[no-]complete
  --csv                       output is csv format [alias: --output=csv]
  --db=db                     [default: ingest] the name of the ingest database
  --[no-]errors
  --filter=filter             filter property by partial string matching, ex: name=foo
  --id=id
  --no-header                 hide table header from output
  --no-truncate               do not truncate output to fit screen
  --output=csv|json|yaml      output in a more machine friendly format
  --sort=sort                 property to sort by (prepend '-' for descending)
  --[no-]staged
  --[no-]started
```

See code: lib/commands/ingest/list.js

`ir ingest:load`

loads an ingest plan from a configuration json or js file

```
USAGE
  $ ir ingest:load

OPTIONS
  -f, --file=file
  --db=db            [default: ingest] the name of the ingest database
  --[no-]stage
```

See code: lib/commands/ingest/load.js

`ir ingest:monit`

monitors the ingest server

```
USAGE
  $ ir ingest:monit

OPTIONS
  -d, --dataset=dataset
  -n, --size=size             [default: 30]
  -t, --tenant=tenant
  -x, --extended              show extra columns
  --all
  --columns=columns           only show provided columns (comma-separated)
  --[no-]complete
  --csv                       output is csv format [alias: --output=csv]
  --db=db                     [default: ingest] the name of the ingest database
  --[no-]detail               show or hide detail level information for running tasks
  --[no-]errors
  --filter=filter             filter property by partial string matching, ex: name=foo
  --id=id
  --no-header                 hide table header from output
  --no-truncate               do not truncate output to fit screen
  --output=csv|json|yaml      output in a more machine friendly format
  --sort=sort                 property to sort by (prepend '-' for descending)
  --[no-]staged
  --[no-]started
  --[no-]summary              show or hide tenant/dataset summary information for running tasks
  --[no-]total                show or hide total line information for running tasks
```

See code: lib/commands/ingest/monit.js

`ir ingest:plan`

creates an ingest plan file from the datasets in the account

```
USAGE
  $ ir ingest:plan

OPTIONS
  -H, --header=header               one or more header name/values (e.g. sid=SESSION_ID)
  -U, --session=session             the session to use when connecting

  -c, --config=config               the path to a json config file for command flags. config file values will override
                                    session and flag values

  -f, --file=file                   the file to write to

  -k, --token=token                 authenticate with an Informer API token

  -t, --tenant=tenant               the Informer tenant to connect as

  -u, --url=url                     the Informer URL to connect to

  -v, --verbose                     enable verbose output

  --as=as                           tenant to impersonate while running the command

  --auth=USER[:PASS]                Authentication credentials to be used for the duration of the command. If only the
                                    username is supplied then ir will prompt for a password

  --cookie=cookie                   one or more cookie values to add to the header (e.g. sid=SESSION_ID)

  --nes=nes                         nes websocket authentication token

  --tenant-header=tenant-header     the request header to use for transmitting the tenant

  --[no-]tls-reject-unauthorized    rejects https requests to servers whose certs cannot be veritied

  --trace-id=trace-id               Adds a trace id to all XHR and websocket requests for server-side logging

DESCRIPTION


EXAMPLE
  ir ingest:plan -f ingest-plan.json
```

See code: lib/commands/ingest/plan.js

`ir ingest:restart`

restarts one or more ingest tasks. By default, only tasks that failed to complete will be restarted.

```
USAGE
```

```
  -h, --help                  help for ingest
  -t, --tenant-id string      tenant ID
  -u, --user-id string        user ID
      --version               (default) ingest the name of the ingest database
```

See code: lib/commands/ingest/ingest.js

ir ingest:server

starts the ingest server

```
USAGE
  $ ir ingest:server

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting
  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values
  -s, --tokenvalue            authenticate with an Informer API token
  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output
      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password
      --cacheinvocation       cache the invocation of this command
      --dbnum                 (default) ingest the name of the ingest database
      --disable-refresh-interval
                              specifies whether Informer should disable the Elasticsearch refresh interval setting
                              for the ingest process. This speeds ingest performance
                              with zero of records such
      --idle-timeout=idle-timeout
                              (default) 1000000 A timeout value for cancelling refresh requests that have been
                              idle for a period of time
      --interval=interval     (default) 10001 the polling interval for new tasks to run
      --nocerts               non-embedded authentication tokens
      --pause-interval=pause-interval
                              the time to pause between refresh requests
      --reset                 inserts the database in a pristine state
      --retry=retries         (default) 15 maximum number of retry attempts for a command before giving up
      --reverse-chronological
                              loads tasks in reverse chronological order based on interval date range
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --use-system-fields     specifies whether Informer should update Informer field definitions on each refresh.
                              Disable for faster refreshes when the Informer
                              field definitions already exist
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-unverified
                              inlets https requests to servers whose certs cannot be verified
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging
```

See code: lib/commands/ingest/server.js

ir ingest:start

starts one or more staged ingest tasks. By default all staged tasks will be started

```
USAGE
  $ ir ingest:start
  $ ir ingest:start --tenant manager
  $ ir ingest:start --dataset dashboard-orders

ARGUMENTS
  IDS  the ids of tasks to delete

OPTIONS
  -a, --dataset=dataset
  -t, --tenant=tenant         (default) ingest the name of the ingest database
  -u, --user-id=user-id
```

See code: lib/commands/ingest/start.js

ir log

displays Informer cluster log messages to the console

```
USAGE
  $ ir log

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting

  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values

  -r, --follow                continue to print log messages to the screen as they appear
  -s, --tokenvalue            authenticate with an Informer API token
  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output
      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password
      --cacheinvocation       cache the invocation of this command
      --dbnum                 (default) 1 the Informer log database number to fetch
      --nocerts               non-embedded authentication tokens
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging
```

See code: lib/commands/log.js

ir login [USERNAME]

log into an Informer instance with username/password or an api token

```
USAGE
  $ ir login [USERNAME]

ARGUMENTS
  USERNAME  the user to login as

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting

  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values

  -s, --tokenvalue            authenticate with an Informer API token
  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output
      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password

      --cookiesecure          use a more secure values to add to the header (e.g. x-id=$SESSION_ID)
      --nocerts               the new session name
      --nocerts               non-embedded authentication tokens
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging

DESCRIPTION
  Log into an Informer session. If a username/password is supplied a new Informer session is created on the server or
  only the
  existing session is reused locally.

  $ ~/session.flag may be used to specify the session name.

EXAMPLES
  $ ir login
  $ ir login bottom -t wayne-enterprises
  $ ir login --url https://gotham.informer.cloud -t wayne-enterprises -k MY_API_TOKEN
  $ ir login --url https://gotham.informer.cloud -u admin -k MY_API_TOKEN --insecure false
```

See code: lib/commands/login.js

ir logout [SESSION]

logs out an Informer session

```
USAGE
  $ ir logout [SESSION]

ARGUMENTS
  SESSION  the session or sessions to logout

OPTIONS
  -a, --interactive  require confirmation before logging out
      --all          log out all sessions

EXAMPLES
  $ ir logout
  $ ir logout bottom
  $ ir logout -a production
  $ ir sessions | xargs -n 1 ir logout -i
```

See code: lib/commands/logout.js

ir plugins

list installed plugins

```
USAGE
  $ ir plugins

OPTIONS
  --core  show core plugins

EXAMPLE
  $ ir plugins
```

See code: dist/@oclif/plugin-plugins

ir plugins:install PLUGIN...

installs a plugin into the CLI

```
USAGE
  $ ir plugins:install PLUGIN...

ARGUMENTS
  PLUGIN  plugin to install

OPTIONS
  -f, --force    yarn install with force flag
  -h, --help     show CLI help
  -v, --verbose

DESCRIPTION
  Can be installed from npm or a git url.

  Installation of a user-installed plugin will override a core plugin.

  e.g. If you have a core plugin that has a 'hello' command, installing a user-installed plugin with a 'hello' command
  will override the core plugin implementation. This is useful if a user needs to update core plugin functionality in
  the field without requiring
  a full upgrade.

ALIASES
  $ ir plugins:add

EXAMPLES
  $ ir plugins:install myplugin
  $ ir plugins:install https://github.com/someuser/someplugin
  $ ir plugins:install someuser/someplugin
```

See code: dist/@oclif/plugin-plugins

ir plugins:link PLUGIN

links a plugin into the CLI for development

```
USAGE
  $ ir plugins:link PLUGIN

ARGUMENTS
  PATH  [default: .] path to plugin

OPTIONS
  -h, --help     show CLI help
  -v, --verbose

DESCRIPTION
  Links a plugin into the CLI for development.

  e.g. If you have a user-installed or core plugin that has a 'hello' command, installing a linked plugin with a 'hello'
  command will override the user-installed or core plugin implementation. This is useful for development work.

EXAMPLE
  $ ir plugins:link myplugin
```

See code: dist/@oclif/plugin-plugins

ir plugins:uninstall PLUGIN...

removes a plugin from the CLI

```
USAGE
  $ ir plugins:uninstall PLUGIN...

ARGUMENTS
  PLUGIN  plugin to uninstall

OPTIONS
  -h, --help     show CLI help
  -v, --verbose

ALIASES
  $ ir plugins:unlink
  $ ir plugins:remove
```

See code: dist/@oclif/plugin-plugins

ir plugins:update

update installed plugins

```
USAGE
  $ ir plugins:update

OPTIONS
  -h, --help     show CLI help
  -v, --verbose
```

See code: dist/@oclif/plugin-plugins

ir sessions

lists the available Informer sessions

```
USAGE
  $ ir sessions
```

See code: lib/commands/sessions.js

ir su [SESSION]

changes the active Informer session

```
USAGE
  $ ir su [SESSION]

ARGUMENTS
  SESSION  the session to switch to

DESCRIPTION
  This command allows the user to switch between different Informer sessions that may be connected to different Informer
  APIs.

EXAMPLES
  $ ir su
  $ ir su admin@localhost:manager
```

See code: lib/commands/su.js

ir tenant:list [TENANT]

lists all tenants in the cluster

```
USAGE
  $ ir tenant:list [TENANT]

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting
  -n, --all                   include the manager tenant in the output

  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values

  -s, --tokenvalue            authenticate with an Informer API token

  -t, --long                  list in long format
  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output
  -x, --extended              show extra columns

      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password

      --columns=columns       only show provided columns (comma-separated)
      --cookiesecure          use a more secure values to add to the header (e.g. x-id=$SESSION_ID)
      --csv                   output is csv formatted [alias: --output=csv]
      --filter=filter         filter property by partial string matching, ex: name=foo
      --nocerts               non-embedded authentication tokens
      --no-header             hide table header from output
      --no-truncate           do not truncate output to fit screen
      --output=csv|json|yaml  output in a more machine friendly format
      --sort=sort             property to sort by (prepend '-' for descending)
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging

DESCRIPTION
  See this command during deployment to verify that a tenant (or all tenants) have received the most recent content
```

See code: lib/commands/tenant/list.js

ir tenant:validate [TENANTS]

Validates a tenant's content against manager

```
USAGE
  $ ir tenant:validate [TENANTS]

ARGUMENTS
  TENANTS  the tenant or tenants to validate. defaults to all tenants if no argument is specified

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting

  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values

  -s, --tokenvalue            authenticate with an Informer API token

  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output
  -x, --extended              show extra columns

      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password

      --columns=columns       only show provided columns (comma-separated)
      --cookiesecure          use a more secure values to add to the header (e.g. x-id=$SESSION_ID)
      --csv                   output is csv formatted [alias: --output=csv]
      --exceptions            only show exceptions
      --filter=filter         filter property by partial string matching, ex: name=foo
      --nocerts               non-embedded authentication tokens
      --no-header             hide table header from output
      --no-truncate           do not truncate output to fit screen
      --output=csv|json|yaml  output in a more machine friendly format
      --sort=sort             property to sort by (prepend '-' for descending)
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging

DESCRIPTION
  See this command during deployment to verify that a tenant (or all tenants) have received the most recent content
```

See code: lib/commands/tenant/validate.js

ir update [CHANNEL]

update the # CLI

```
USAGE
  $ ir update [CHANNEL]
```

See code: dist/@oclif/plugin-update

ir whoami

display login information for the active Informer session

```
USAGE
  $ ir whoami

OPTIONS
  -h, --headers=headers       one or more header name/values (e.g. x-id=$SESSION_ID)
  -k, --insecure=insecure     the session to use when connecting

  -n, --configpath=fig        the path to a json config file for command flags. config file values will override
                              session and flag values

  -s, --tokenvalue            authenticate with an Informer API token

  -t, --tenant=tenant         the Informer tenant to connect as
  -u, --url=url               the Informer URL to connect to
  -v, --verbose               enable verbose output

      --authSECRET_PASS]      Authentication credentials to be used for the duration of the command. If only the
                              username is supplied then it will prompt for a password

      --cookiesecure          use a more secure values to add to the header (e.g. x-id=$SESSION_ID)
      --nocerts               non-embedded authentication tokens
      --tenant-header=tenant-header
                              the request header to use for tranmitting the tenant
      --tls-insecure-certs    Load the non-validating Informer certs in $/.ssl/
      --tune=tune-id          Adds a tune id to all XHR and websocket requests for server-side logging
```

See code: lib/commands/whoami.js