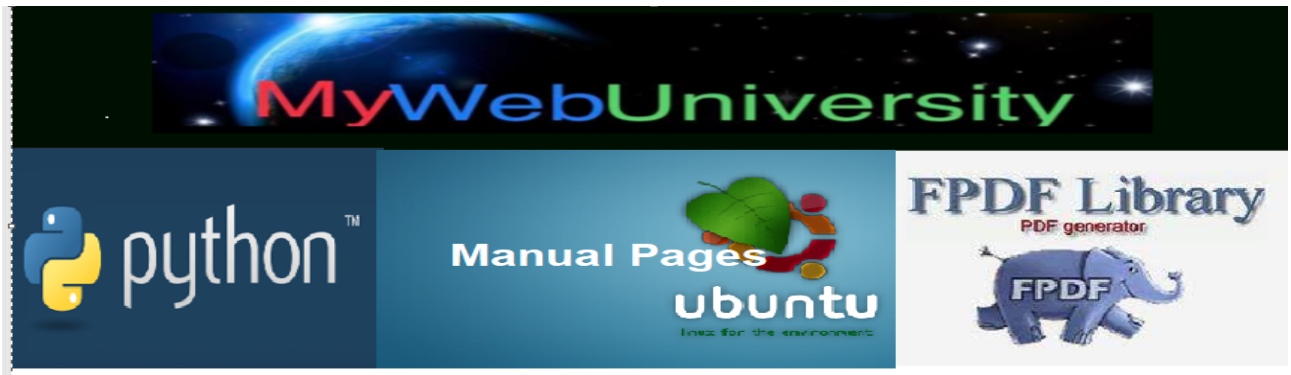

Linux Ubuntu 26.04 Manual Pages on command 'tinysparql-endpoint.1'

$ man tinysparql-endpoint.1

TINYSPARQL-ENDPOINT(1) TinySPARQL manual TINYSPARQL-ENDPOINT(1)

NAME

tinysparql-endpoint - Create a SPARQL endpoint

SYNOPSIS

tinysparql endpoint [--dbus-service | -b] <service_name>

[--database-path | -d] <database_path>

[[--ontology | -o] <ontology_name> |

[--ontology-path | -p] <ontology_path>]

[--http-port] <port>

[--loopback]

[[--system | --session]]

DESCRIPTION

This command allows creating SPARQL endpoints. The endpoint will be able to handle SPARQL select and update queries, and notify about changes in it.

The endpoint is exported via DBus, accessible through the given service_name, either using it in a SERVICE clause, or by creating a dedicated bus-based SPARQL connection.

When creating a database, the ontology_name (or alternatively, a ontology_path) must be provided in order to generate the database. If ontology_name is used, the ontology must exist in $datadir/tracker/ontologies

Users should be careful about the --http-port option, consider using it only combined with the --loopback option.

The database itself will be stored according to database_path.

OPTIONS

-b, --dbus-service=<service_name>

Service name to use on the endpoint.

-d, --database-path=<database_path>

The path where the database will be stored.

-o, --ontology

The name of an ontology in $datadir/tracker/ontologies to use on the constructed database.

-p, --ontology-path

Full path to an ontology to use on the constructed database.

--session

Use the session bus. This is the default.

--system

Use the system bus.

-l, --list

List all SPARQL endpoints available in DBus

-L, --list-http

List network-local HTTP SPARQL endpoints

--http-port

Creates a HTTP endpoint that listens in the specified port

--loopback

Allows only HTTP connections in the loopback device. Only effective with HTTP endpoints.

EXAMPLES

Export a Nepomuk endpoint with the org.example.Example1 bus name.

```
$ tinysparql endpoint -b org.example.Example1 -o nepomuk -d /tmp/example1
```

Access this endpoint with the tinysparql-sparql(1) subcommand.

```
$ tinysparql sparql --dbus-service org.example.Example1 -q ''
  SELECT ?s ?o
  WHERE {
    ?u a ?o
  }''
```

Export a Nepomuk endpoint via HTTP.

```
$ tinysparql endpoint --http-port 8080 -o nepomuk --loopback
```

Access this endpoint via HTTP.

```
$ tinysparql sparql --remote-service http://127.0.0.1:8080/sparql -q "
  SELECT ?u {
    ?u a rdfs:Resource
  }"
```

SEE ALSO

tinysparql-sparql(1),

https://www.w3.org/TR/sparql11-query/