Informes Técnicos Ciemat 1532
Julio, 2024

TJ-II WEB SERVICES

Van Milligen, B.P.
Baciero Adrados, A.

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

Publication available in catalog of official publications

ISSN: 2695-8864
NIPO: 152-24-007-1

Edition and Publication:
Editorial CIEMAT
Avda. Complutense, 40 28040-MADRID
e-mail: editorial@ciemat.es
Editorial news

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

TJ-II is a stellarator of the flexible heliac type, currently in operation at the Centro de Investigaciones Energéticas, MedioAmbientales y Tecnológicas (CIEMAT) in Madrid, Spain [1,2]. To facilitate data access and analysis, Web Services have been developed that are described in the following. The purpose of these Web Services is to allow inspecting and accessing TJII-relevant data and information from both inside and outside the National Fusion Laboratory, and from any platform (Windows, Linux, Mac). The main advantage of the Web Services is that they do not require installing any non-standard libraries, thus facilitating data access to visitors and external collaborators.

The webservices provide an interface allowing the retrieval of information from various TJ-II databases through the http protocol. The practical implementation of this interface involves various small Common Gateway Interface (CGI) programs that accept a limited set of parameters. When called, the web server locally executes the corresponding CGI program with the specified parameters, retrieves the requested information and returns output over the http protocol, either in the form of an HTML-formatted web page or in the form of a JavaScript Object Notation (JSON) encoded data file.

2 INTERACTIVE DATA DISPLAY

The interactive data display allows plotting up to 30 ‘signals’ (time traces) of data for a given discharge. In order to reduce the response time for displaying the data, no more than about 500 points are shown per time trace. Thus, the display is low resolution; it is only meant to offer a quick first inspection of the data.

Link to access TJII_data

Figure 1 shows a sample output page. The page is interactive and allows selecting other shots and signals to display. Note that the signal lists are drop-down lists that may vary (slightly) from one shot to the next. The current drop-down lists always correspond to the latest shot displayed. Therefore, it may be necessary to first change shot number and click ‘Submit’ before the corresponding drop-down signal lists are displayed correctly.

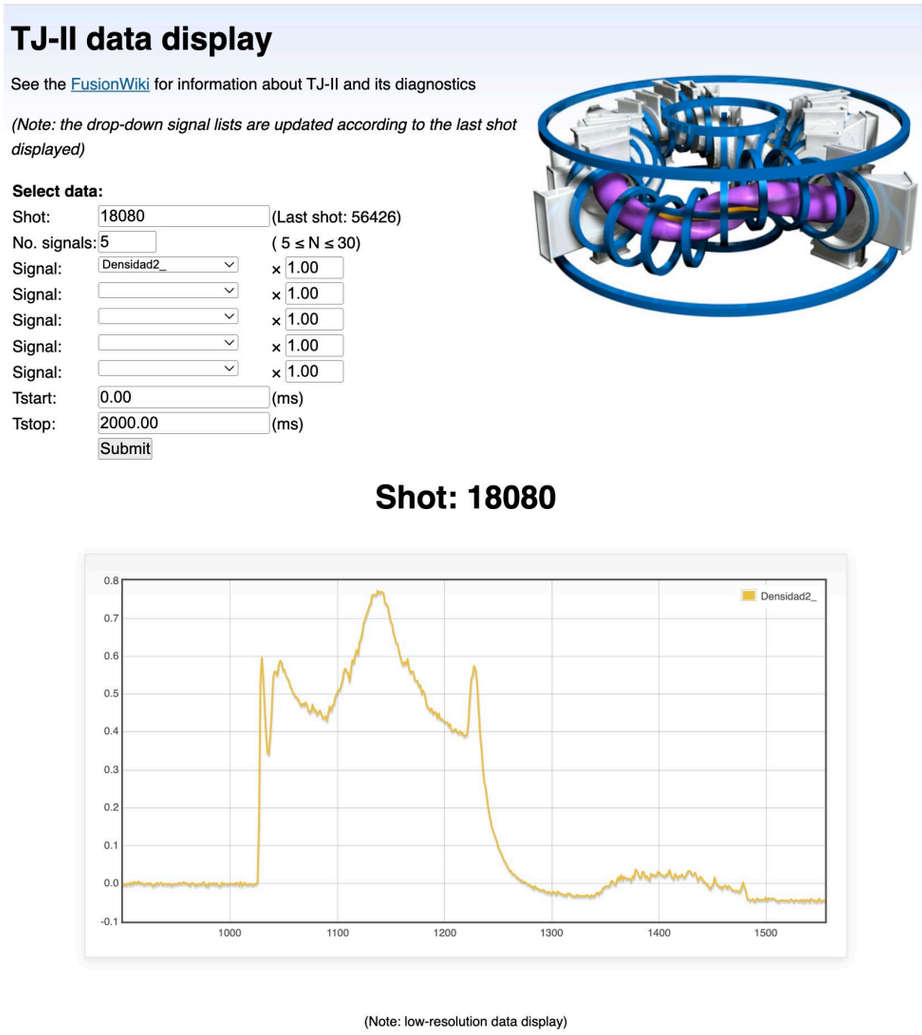

Figure 1 Sample web page output from the TJII_data.cgi program.

3 DATA ACCESS INTERFACE

Various routines are available for data access, implementing a service similar to the RpcC Library [3] (based on remote procedure calls, RPC). Data access via the web service is much slower than data access via the RPC library, but may be useful in specific situations (e.g., for remote access from outside the Laboratory network, or in the case of modern Windows platforms no longer supporting the 32 bit RPC library). Output is returned in JSON format. Example calls are provided in Appendix A, “Data Access examples”.

3.1 OBTAINING THE LIST OF AVAILABLE SIGNALS

The list of available signals for a given discharge is obtained through the CGI program TJII_getlist.cgi. Input parameter: ‘shot’.

3.2 READING A TIME TRACE

The time trace of a given signal can be obtained through the CGI program TJII_getdata.cgi. Output is returned in JSON format. Input parameters: ‘shot’ and ‘signal’.

Reading long time traces may lead to timeout errors due to the relatively slow data transfer of the web access method. To facilitate accessing such long time traces (with a high sampling rate), one can delimit the time window of interest using the optional parameters ‘tstart’ and ‘tstop’.

To facilitate data export, the output can be formatted as plain text (instead of JSON) by setting the optional input parameter ‘output’ to value ‘plain’.

3.3 READING A PROFILE

The profile of a given signal can be obtained through the CGI program TJII_getprof.cgi. Output is returned in JSON format. Input parameters: ‘shot’ and ‘signal’.

The Thomson Scattering diagnostic generates the following profiles:

“PerfilRho_”, “PerfilNe_”, “PerfildNe_”, “PerfilTe_”, “PerfildTe_”, “PerfilPe_”, “PerfildPe_”

4 RELATIONAL DATABASE INTERFACE

The relational database stores some information about TJ-II experiments in the form of single parameters [4]. A list of stored parameters is provided in Appendix D, “List of Parameters in the Relational Database”, but note that this list may change slightly over time. Example calls are provided in Appendix B, “Relational Database Interface examples”.

4.1 RETRIEVING DATABASE PARAMETERS FOR A RANGE OF SHOTS OR DATES

The web interface provides a limited facility to consult the values of these parameters, through the CGI program TJII_getrdb.cgi. Input parameters: ‘variables’ and ‘shots’ or ‘dates’. Here, ‘variables’ is a comma-separated list of parameters, ‘shots’ is a comma-separated shot range, and ‘dates’ is a comma-separated range of dates in format ‘yyyymmdd’.

5 MAGNETIC CONFIGURATION INTERFACE

The interpretation of data from the TJ-II stellarator often requires converting real space coordinates to flux coordinates, in particular when comparing diagnostic systems located at different positions. The magnetic configuration in vacuum usually provides a good reference for such analyses and comparisons [5].

A library of functions is available to obtain information about the vacuum magnetic configuration of TJ-II [6]. Please consult this manual for more detailed information about the functions. The parameter 'funct' used in the CGI calls corresponds to the function name described in the cited function library manual.

The functions provided by this library can be accessed through the CGI program TJII_getmagn.cgi. Output is returned in JSON format. The functions allow listing the set of available magnetic configurations, obtaining magnetic flux and field coordinates for these configurations, and obtaining vacuum vessel contours in a poloidal cross section.

All quantities are given in MKS units (m, radians, T). To adjust for optimal ECRH absorption, the absolute magnetic field should be renormalized to 0.95 T at the magnetic axis ($\psi = 0$) at toroidal angle $\phi = 25.5°$.

Required arguments are indicated in the examples, provided in Appendix C: "Magnetic Configuration Interface Examples". Omitted numerical arguments are zero by default. Any non-required arguments will be ignored. All arguments must be specified in lower case.

5.1 INTERPOLATING BETWEEN CONFIGURATIONS

The list of available configurations is limited. However, one may interpolate returned (output) values between a "linear" subset of configurations (characterized mainly by the variation of a single configuration parameter). For example, the configurations 100_xx_yy form such a (linear) subset, in which only xx varies significantly. The parameter xx can then be used as the independent interpolation variable. For example, if one is interested in configuration 100_42_64, one may interpolate linearly between returned values for configurations 100_40_63 and 100_44_64. Interpolation can be improved further by using spline interpolation between the values returned for at least 3 configurations, e.g. 100_36_62, 100_40_63, 100_44_64 and 100_48_65.

6 CALLING THE WEB SERVICES FROM MATLAB

The data access and magnetic configuration functions can be called from MATLAB. For this purpose, the URL is constructed first using standard string operations, e.g.

```
config = '100_44_64';
x = 1.7;
y = 0;
z = 0.05;
urlbase = 'https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?config=%s&x=%f&y=%f&z=%f';
url = sprintf(urlbase,config,x,y,z);
```

and then the call is executed as:

```
options = weboptions('ContentType','json','Timeout',30);
result = webread(url,options);
```

The returned object, 'result', is a MATLAB structure. Its elements can be accessed using standard dot notation, e.g. result.psi would be the magnetic flux.

The construction of the URL string and the CGI function call can be encapsulated in MATLAB functions for more user-friendliness. The following generic interface function (TJII_get.m) allows calling all CGI functions described above (TJII_getlist, TJII_getdata, TJII_getprof, TJII_getrdb, TJII_getmagn) from MATLAB:

```
function result = TJII_get(varargin)
% varargin: the first argument is 'list', 'data', 'prof', 'rdb', 'magn' % subsequent arguments are a
comma-separated list of keyword,value combinations
url = ['https://info.fusion.ciemat.es/cgi-bin/TJII_get',varargin{1},'.cgi?'];
for i=2:2:length(varargin)-1
    if i>2, form = '&'; else form=''; end
    if isa(varargin{i+1},'char'), form2='%s=%s'; else form2='%s=%g'; end
    form = [form,form2];
    newarg = sprintf(form,varargin{i},varargin{i+1});
    url = [url,newarg];
end
options = weboptions('ContentType','json','Timeout',30);
result = webread(url,options);
end
```

The first argument specifies the function to call. Subsequent arguments are (keyword, value) combinations. Numerical arguments can either be specified as numbers (56400, 0.05) or as character strings ('56400', '0.05'). Examples:

```
result = TJII_get('list','shot',56400)
result = TJII_get('magn','funct','flux_cyl','config','100_44_64','r',1.7,'phi',0,'z',0.05)
result = TJII_get('rdb','shots','56400,56401','variables','Ndes,configuracion')
```

6.1 CALLING THE MAGNETIC CONFIGURATION FUNCTIONS FROM MATLAB

The following is an example of a call to obtain information about the magnetic configuration, using the function 'TJII_get.m' above:

```
result = TJII_get('magn','funct','flux_cyl','config','100_44_64','r',1.7,'phi',0,'z',0.05)
```

This example executes a call to TJII_getmagn.cgi with "funct=flux_cyl&config=100_44_64&r=1.7&phi=0&z=0.05" and returns the result. In this particular example, result.psi = 0.0974.

Using this function, the following example MATLAB script will plot a rotational transform (iota) profile for configuration 100_44_64:

```
clear
config = '100_44_64';
n=50;
for i=1:n
    psi(i) = (i-1)/(n-1);
    result = TJII_get('magn','funct','iota','config',config,'psi',psi(i));
    iotar(i) = -result.iota;
end
close all
plot(sqrt(psi),iotar,'o-')
```

6.2 PLOTTING A TIME TRACE USING MATLAB

The following MATLAB function will return a time trace:

```
function [time,data,shot,error] = TJII_getdata_web(shot,signal,tstart,tstop)
% Input:
% shot: integer shot number. Enter 0 to obtain last shot number
% signal: character string (signal name)
% tstart,tstop: OPTIONAL, start and stop time in ms (saves time with long signals)
% Output:
% If no error, then error is empty, otherwise it contains an error message (string)
% Note that shot is reset to its actual value (last shot) if input shot=0
if nargin <= 2
    urlbase = 'http://www-fusion.ciemat.es/cgi-bin/TJII_getdata.cgi?shot=%d&signal=%s';
    url = sprintf(urlbase,shot,signal);
else
    urlbase = 'http://www-fusion.ciemat.es/cgi-
bin/TJII_getdata.cgi?shot=%d&signal=%s&tstart=%d&tstop=%d';
    url = sprintf(urlbase,shot,signal,tstart,tstop);
end
fprintf('%s¥n',url)

options = weboptions('ContentType','json','Timeout',30);
temp = webread(url,options);
error = temp.error;
if isfield(temp,'shot'), shot = temp.shot; end
if isfield(temp,'time'), time = temp.time; else time=NaN; end
if isfield(temp,'data'), data = temp.data; else data=NaN; end
if ~isempty(error), warning(error); end
end
```

This function is called by the following script:

```
shot = 42136;
signal = 'Densidad2_';
[time,data,shot,err] = TJII_getdata_web(shot,signal);
plot(time,data,'LineWidth',2)
set(gca,'FontSize',18)
```

Resulting in the graph shown in Figure 2.

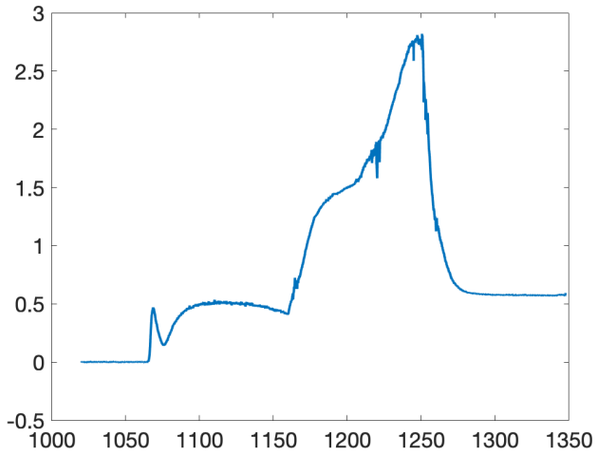

Figure 2 Output produced by the MATLAB example for plotting a time trace.

6.3 PLOTTING A PROFILE USING MATLAB

Similarly, the following MATLAB function will return a profile:

```
function [time,prof,shot,error] = TJII_getprof_web(shot,signal)
% Input:
% shot: integer shot number. Enter 0 to obtain last shot number
% signal: character string (signal name)
% Output:
% If no error, then error is empty, otherwise it contains an error message (string)
% Note that shot is reset to its actual value (last shot) if input shot=0
urlbase = 'http://www-fusion.ciemat.es/cgi-bin/TJII_getprof.cgi?shot=%d&signal=%s';
url = sprintf(urlbase,shot,signal);
fprintf('%s¥n',url)

options = weboptions('ContentType','json','Timeout',30);
temp = webread(url,options);
error = temp.error;
if isfield(temp,'shot'), shot = temp.shot; end
if isfield(temp,'time'), time = temp.time; else time=NaN; end
if isfield(temp,'prof'), prof = temp.prof; else prof=NaN; end
if ~isempty(error), warning(error); end
end
```

This function is called by the following script:

```
shot = 42136;
signal = 'PerfilRho_';
[time,rho,shot,err] = TJII_getprof_web(shot,signal);
signal = 'PerfilTe_';
[time,Te,shot,err] = TJII_getprof_web(shot,signal);
signal = 'PerfildTe_';
[time,dTe,shot,err] = TJII_getprof_web(shot,signal);
errorbar(rho,Te,dTe,'o','LineWidth',2)
set(gca,'FontSize',18)
```

Resulting in the graph shown in Figure 3

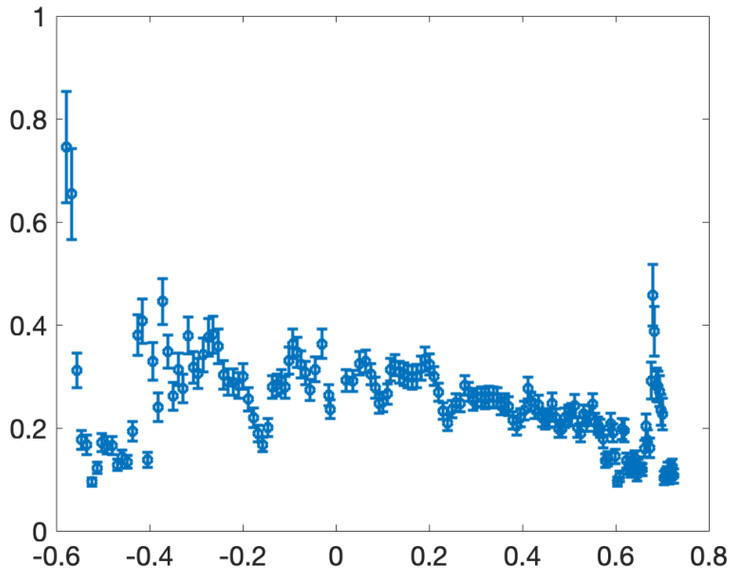

Figure 3 Output produced by the MATLAB example for plotting a profile.

6.4 QUERYING THE RELATIONAL DATABASE FROM MATLAB

The relational database can also be queried using the function TJII_get.m above. Examples:

```
result = TJII_get('rdb','shots','56400,56401','variables','Ndes,configuracion')
result = TJII_get('rdb','dates','20240314','variables','Ndes,configuracion')
```

7 CALLING THE WEB SERVICES FROM PYTHON

The data access and magnetic configuration functions can be called directly from Python with the Requests library (see section 8.1 below for information on obtaining the Requests library). Example:

```
import requests
getmagn_url = 'https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi'
funct = 'flux_car'
config = '100_44_64'
x = 1.7
y = 0.0
z = 0.05
payload = [('funct', funct), ('config', config), ('x', x), ('y', y), ('z', z)]
# verify=False can be needed to skip problems with SSL certificates
#              and then, it prints a warning message
result = requests.get(getmagn_url, params=payload, verify=False).json()
```

7.1 TJ2SERVICES LIBRARY

In addition to the above basic access method via Python, a function library has been elaborated to facilitate accessing the TJ-II Webservices via Python. It consists of a file named “tj2services.py” that can be obtained from the LNF intranet: Manuals and Software -> Python -> python_tj2_web_services.zip. The library requires Python 3.6 and Requests, a widely used Python library. The easiest method to install Requests is to use pip:

```
python -m pip install requests
```

All arguments of the functions below are required and they should be provided as keyword arguments. The function returns a dictionary with the result. If the function was successful, the key “error” of the dictionary should be an empty string. For example:

```
import tj2services
r = tj2services.getdata(shot=50000, signal='ACTON275')
print('Error value:', r['error'])  # Check error
print('Time and data:', r['time'], r['data'])
```

All parameters of the functions of the magnetic configuration interface are in MKS units (m, radian, T).

7.2 TJ2SERVICES EXAMPLES

For some examples, the matplotlib library may be required. The first example is for reading the time trace of a given signal (getdata):

```
import tj2services
import matplotlib.pyplot as plt
r = tj2services.getdata(shot=50000, signal='ACTON275')
plt.plot(r['time'], r['data'])
plt.show()
```

Resulting in the graph shown in Figure 4.

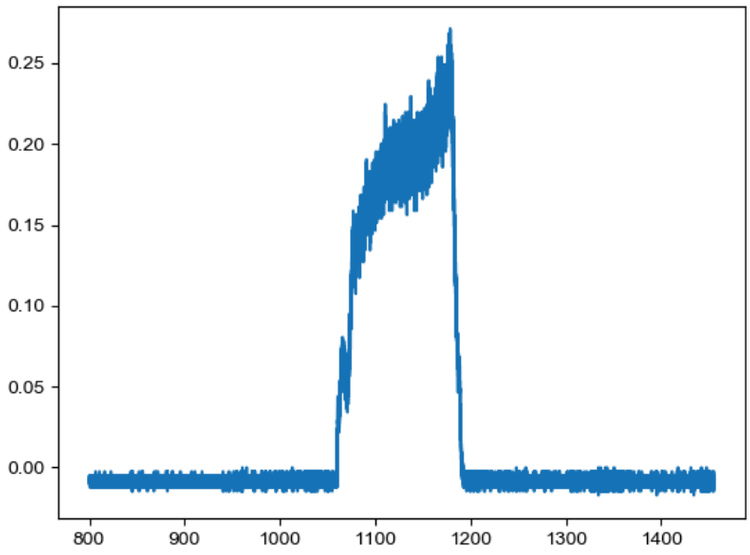

Figure 4 Output produced by the Python example for plotting a time trace.

Example: reading and plotting Thomson scattering diagnostic data (getprof):

```
import tj2services
import matplotlib.pyplot as plt
r_rho = tj2services.getprof(shot=50000, signal='PerfilRho_')
r_ne = tj2services.getprof(shot=50000, signal='PerfilNe_')
r_dne = tj2services.getprof(shot=50000, signal='PerfildNe_')
x = r_rho['prof']
y = r_ne['prof']
yerr = r_dne['prof']
plt.errorbar(x, y, yerr=yerr, marker='o', lw=0, elinewidth=1)
plt.show()
```

Resulting in the graph shown in Figure 5.

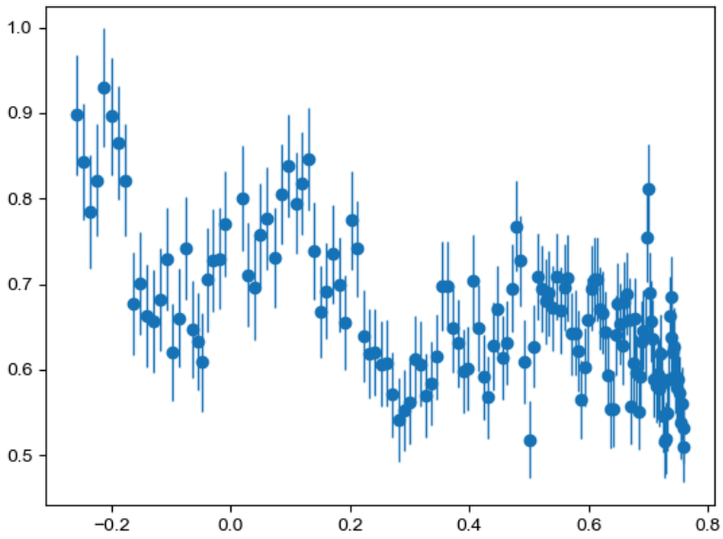

Figure 5 Output produced by the Python example for plotting a profile.

Example: reading database parameters (getrdb):

```
import tj2services
r1 = tj2services.getrdb_shots(variables='CONFIGURACION', shots=49999)
print('First query:')
print(r1['info'])
print()
r2 = tj2services.getrdb_shots(variables=['NDES', 'CONFIGURACION'], shots=(50000, 50002))
print('Second query:')
print(r2['info'])
```

Resulting in the following output:

```
First query:
[{'CONFIGURACION': '082_061_063'}]
Second query:
[{'NDES': 50000, 'CONFIGURACION': '082_061_063'}, {'NDES': 50001, 'CONFIGURACION': '082_061_063'},
{'NDES': 50002, 'CONFIGURACION': '082_061_063'}]
```

Example: obtaining the normalized magnetic flux (flux_car):

```
import tj2services
r = tj2services.flux_car(config='100_44_64', x=1.7, y=0, z=0.05)
print(f'Value of the flux for (x={r["x"]}, y={r["y"]}, z={r["z"]}):', r['psi'])
```

Resulting in the following output:

```
Value of the flux for (x=1.7, y=0.0, z=0.050000001): 0.097419985
```

7.3 TJ2SERVICES LIBRARY APPLICATION PROGRAMMING INTERFACE (API)

Below, the list of available library functions is summarized.

- *tj2services.getlist*(*, shot). Obtaining the list of available signals for a given discharge.
- *tj2services.getdata*(*, shot, signal). Reading the time trace of a given signal.
- *tj2services.getprof*(*, shot, signal). Reading a given profile.
- *tj2services.getrdb_shots*(*, variables, shots). Retrieving database parameters for a range of shots. “variables” can be a string or a list of strings; and “shots”, a pulse number or a sequence of two shots (shot_min, shot_max).
- *tj2services.getrdb_dates*(*, variables, dates). Retrieving database parameters for a range of dates. “variables” can be a string or a list of strings; and “dates”, a date or a sequence of two dates (date_min, date_max). Date can be a string (“YYYYDDMM” or “Y-M-D”) or a datetime.date instance.
- *tj2services.flux_cyl*(*, config, r, phi, z). To obtain the normalized magnetic flux, cylindrical coordinates.
- *tj2services.flux_car*(*, config, x, y, z). To obtain the normalized magnetic flux, cartesian coordinates.
- *tj2services.grad_flux_cyl*(*, config, r, phi, z). To obtain the gradient of the flux magnetic field, cylindrical coordinates.
- *tj2services.grad_flux_car*(*, config, x, y, z). To obtain the gradient of the flux magnetic field, cartesian coordinates.
- *tj2services.flux_surf_cyl*(*, config, psi, theta, phi). To convert magnetic flux coordinates to cylindrical coordinates.
- *tj2services.flux_surf_car*(*, config, psi, theta, phi). To convert magnetic flux coordinates to cartesian coordinates.
- *tj2services.b_field_cyl*(*, config, r, phi, z). To obtain the magnetic field, cylindrical coordinates.
- *tj2services.b_field_car*(*, config, x, y, z). To obtain the magnetic field, cartesian coordinates.

- *tj2services.grad_b_cyl*(*, config, r, phi, z). To obtain the gradient of the absolute magnetic field, cylindrical coordinates.
- *tj2services.grad_b_car*(*, config, x, y, z). To obtain the gradient of the absolute magnetic field, cartesian coordinates.
- *tj2services.iota*(*, config, psi). To obtain the rotational transform.
- *tj2services.iota_cyl*(*, config, r, phi, z). To obtain the rotational transform, cylindrical coordinates.
- *tj2services.iota_car*(*, config, x, y, z). To obtain the rotational transform, cartesian coordinates.
- *tj2services.volume*(*, config, psi). To obtain the volume.
- *tj2services.volume_cyl*(*, config, r, phi, z). To obtain the volume, cylindrical coordinates.
- *tj2services.volume_car*(*, config, x, y, z). To obtain the volume, cartesian coordinates.
- *tj2services.surface*(*, config, psi). To obtain the surface.
- *tj2services.surface_cyl*(*, config, r, phi, z). To obtain the volume, cylindrical coordinates.
- *tj2services.surface_car*(*, config, x, y, z). To obtain the volume, cartesian coordinates.
- *tj2services.configlist*(). To obtain a list of available configurations.
- *tj2services.tj2*(*, phi). To obtain the contour of the vacuum vessel at a given phi.

8 CALLING THE WEB SERVICES USING OTHER METHODS

If the use of MATLAB or Python is not desired or not possible, one may still obtain the required information using other methods. Most programming languages include similar facilities to access a URL link and recover the output.

For example, unix/linux systems dispose of a command ‘curl’ that can be invoked as follows in a terminal window:

```
curl --user-agent "Mozilla/5.0" --output - "https://info.fusion.ciemat.es/cgi-bin/TJII_getlist.cgi?shot=40000"
```

The output of this call can be redirected to a file (using --output <file>). The generated file contains the returned information in JSON format. The JSON file contents can be converted to other formats (like CSV) using standard tools.

The most basic way to obtain information is simply by entering the required CGI string in the address bar of a web browser. The contents of the returned web page (in JSON format) can be selected and copied to a text file and saved.

9 REFERENCES

[1] C. Alejaldre et al. First plasmas in the TJ-II flexible heliac. Plasma Phys. Control. Fusion, 41:A539, 1999.

[2] C. Hidalgo et al. Overview of the TJ-II stellarator research programme towards model validation in fusion plasmas. Nucl. Fusion, 62(4):042025, 2022.

[3] TJ2 RPC Library, see http://fudaqs2.ciemat.es/, item 'Data Help' (you need to be logged in)

[4] E. Sánchez, A.B. Portas, J. Vega. Librería de Acceso a la Base de Datos Relacional de TJ-II: Guía del Usuario, Technical report, CIEMAT-1029, 2003

[5] B.Ph. van Milligen et al. Integrated data analysis at TJ-II: the density profile. Rev. Sci. Instrum., 82(7):073503, 2011.

[6] V. Tribaldos, B.Ph. van Milligen, A. López-Fraguas. TJ-II Library Manual. Technical report, CIEMAT-963, 2001

APPENDIX

A. DATA ACCESS EXAMPLES

This appendix provides examples of the URL strings that can either be used directly (by clicking on them or copying them to a web browser), or constructed by and called from a program script or user interface, to obtain data from the TJ-II shot database.

OBTAINING THE LIST OF AVAILABLE SIGNALS

https://info.fusion.ciemat.es/cgi-bin/TJII_getlist.cgi?shot=40000

READING A TIME TRACE

https://info.fusion.ciemat.es/cgi-bin/TJII_getdata.cgi?shot=40000&signal=Densidad2_

Optional limitation of time window using parameters tstart and tstop:

https://info.fusion.ciemat.es/cgi-bin/TJII_getdata.cgi?shot=40000&signal=LOP01&tstart=1200&tstop=1210

To facilitate data export, the output can be formatted as plain text (instead of JSON) by setting “output=plain”:

https://info.fusion.ciemat.es/cgi-bin/TJII_getdata.cgi?shot=40000&signal=Densidad2_&output=plain

READING A PROFILE

https://info.fusion.ciemat.es/cgi-bin/TJII_getprof.cgi?shot=18080&signal=PerfilTe_

B. RELATIONAL DATABASE INTERFACE EXAMPLES

This appendix provides examples of the URL strings that can either be used directly (by clicking on them or copying them to a web browser), or constructed by and called from a program script or user interface, to obtain data from the TJ-II relational database.

RETRIEVING DATABASE PARAMETERS FOR A RANGE OF SHOTS

For a single shot:

https://info.fusion.ciemat.es/cgi-bin/TJII_getrdb.cgi?variables=Ndes,configuracion&shots=40000

For a (comma-separated) range of shots:

https://info.fusion.ciemat.es/cgi-bin/TJII_getrdb.cgi?variables=Ndes,configuracion&shots=40000,40002

RETRIEVING DATABASE PARAMETERS FOR A RANGE OF DATES

For a single date (format yyyymmdd):

https://info.fusion.ciemat.es/cgi-bin/TJII_getrdb.cgi?variables=Ndes,configuracion&dates=20170308

Or for a (comma-separated) range of dates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getrdb.cgi?variables=Ndes,configuracion&dates=20170308,20170309

Requesting a large quantity of information in a single call may lead to buffer overflow errors.

C. MAGNETIC CONFIGURATION INTERFACE EXAMPLES

This appendix provides examples of the URL strings that can either be used directly (by clicking on them or copying them to a web browser), or constructed by and called from a program script or user interface, to obtain information regarding TJ-II magnetic configurations.

OBTAINING THE LIST OF AVAILABLE MAGNETIC CONFIGURATIONS

Function configlist – to obtain a list of available configurations:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=configlist

OBTAINING MAGNETIC CONFIGURATION DATA

Function flux_cyl – to obtain the normalized magnetic flux, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=flux_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function flux_car – to obtain the normalized magnetic flux, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=flux_car&config=100_44_64&x=1.7&y=0&z=0.05

If 'funct' is not specified, 'flux_car' is called by default:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?config=100_44_64&x=1.7&y=0&z=0.05

Function grad_flux_cyl – to obtain the gradient of the magnetic flux, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=grad_flux_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function grad_flux_car – to obtain the gradient of the magnetic flux, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=grad_flux_car&config=100_44_64&x=1.7&y=0&z=0.05

Function flux_surf_cyl – to convert magnetic flux coordinates to cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=flux_surf_cyl&config=100_44_64&psi=0.5&theta=0&phi=0

Function flux_surf_car – to convert magnetic flux coordinates to cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=flux_surf_car&config=100_44_64&psi=0.5&theta=0&phi=0

Function b_field_cyl – to obtain the magnetic field, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=b_field_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function b_field_car – to obtain the magnetic field, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=b_field_car&config=100_44_64&x=1.7&y=0&z=0.05

Function grad_b _cyl – to obtain the gradient of the absolute magnetic field, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=grad_b_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function grad_b_car – to obtain the gradient of the absolute magnetic field, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=grad_b_car&config=100_44_64&x=1.7&y=0&z=0.05

Function iota – to obtain the rotational transform:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=iota&config=100_44_64&psi=0.5

Function iota_cyl – to obtain the rotational transform, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=iota_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function iota_car – to obtain the rotational transform, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=iota_car&config=100_44_64&x=1.7&y=0&z=0.05

Function volume – to obtain the enclosed volume:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=volume&config=100_44_64&psi=0.5

Function volume_cyl – to obtain the volume, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=volume_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function volume_car – to obtain the volume, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=volume_car&config=100_44_64&x=1.7&y=0&z=0.05

Function surface – to obtain the surface area:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=surface&config=100_44_64&psi=0.5

Function surface_cyl – to obtain the surface area, cylindrical coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=surface_cyl&config=100_44_64&r=1.7&phi=0&z=0.05

Function surface_car – to obtain the surface area, cartesian coordinates:

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=surface_car&config=100_44_64&x=1.7&y=0&z=0.05

OBTAINING VACUUM VESSEL CONTOURS

Function tj2 – to obtain the contour of the vacuum vessel at a given toroidal angle phi (in radians; values are stored at half-degree intervals):

https://info.fusion.ciemat.es/cgi-bin/TJII_getmagn.cgi?funct=tj2&phi=0.2182

D. LIST OF PARAMETERS IN THE RELATIONAL DATABASE

Below is a list of parameters currently included in the Relational Database [4]. Please note that the parameters contained in the database may vary (slightly) according to the date of the experiments, as more parameters have been included over time.

NDes
fecha
hora
comentarioDesc
comentarioExp
configuracion
limitador_z2
presion_base
ne_corte
pared
validada
DesFen
DesEqh
puesto_DesEqh
DesGas
DesVal
Densidad_linea_ARR
Te_ECE_central_ARR
Intensidad_HXR_ARR
Monitores_impurezas_ARR
Ti_espectroscopia_ARR
Ti_CX_ARR
Te_Thomson_max_ARR
Densidad_linea_IT
Te_ECE_central_IT
Intensidad_HXR_IT
Monitores_impurezas_IT
Ti_espectroscopia_IT
Ti_CX_IT
Te_Thomson_max_IT
Densidad_linea_DLMAX
Te_ECE_central_DLMAX
Intensidad_HXR_DLMAX
Monitores_impurezas_DLMAX
Ti_espectroscopia_DLMAX
Ti_CX_DLMAX
Te_Thomson_max_DLMAX
Densidad_linea_PLA
Te_ECE_central_PLA
Intensidad_HXR_PLA
Monitores_impurezas_PLA
Ti_espectroscopia_PLA
Ti_CX_PLA
Te_Thomson_max_PLA
Densidad_linea_PMAX
Te_ECE_central_PMAX
Intensidad_HXR_PMAX
Monitores_impurezas_PMAX
Ti_espectroscopia_PMAX
Ti_CX_PMAX
Te_Thomson_max_PMAX
Densidad_linea_TEMAX
Te_ECE_central_TEMAX
Intensidad_HXR_TEMAX
Monitores_impurezas_TEMAX
Ti_espectroscopia_TEMAX
Ti_CX_TEMAX
Te_Thomson_max_TEMAX
Densidad_sondas_BOR
Densidad_haz_Li_BOR
Te_sondas_BOR
Te_thomson_BOR
Te_ECE_BOR
Potencial_electrico_BOR
Iota_BOR
Presion_BOR
Emisividad_SXR_BOR
Emision_bolometrica_BOR
Instante_referencia_BOR
Densidad_sondas_CEN
Densidad_haz_Li_CEN
Te_sondas_CEN
Te_thomson_CEN
Te_ECE_CEN
Potencial_electrico_CEN
Iota_CEN

Presion_CEN

Emisividad_SXR_CEN

Emision_bolometrica_CEN

Instante_referencia_CEN

Densidad_sondas_ZG

Densidad_haz_Li_ZG

Te_sondas_ZG

Te_thomson_ZG

Te_ECE_ZG

Potencial_electrico_ZG

Iota_ZG

Presion_ZG

Emisividad_SXR_ZG

Emision_bolometrica_ZG

Instante_referencia_ZG

limitador_z1

itf

icc

hx

vf

potencia_nominal_ECRH1

potencia_depositada_ECRH1

anchura_haz_ECRH1

punto_deposicion_ECRH1

inyeccion_OnOff_axis_ECRH1

modulacion_ECRH1

angulo_polarizacion_lineal_ECRH1

angulo_polarizacion_eliptica_ECRH1

angulo_toroidal_deposicion_ECRH1

angulo_1_ECRH1

angulo_2_ECRH1

n_paralelo_ECRH1

fmod_ECRH1

rho_ECRH1

tini_ECRH1

longitud_pulso_nominal_ECRH1

updated_ECRH1

longitud_pulso_real_ECRH1

VAccel_nominal_NBI1

IAccel_nominal_NBI1

tini_NBI1

longitud_pulso_nominal_NBI1

potencia_nominal_NBI1

VAccel_real_NBI1

IAccel_real_NBI1

longitud_pulso_real_NBI1

updated_NBI1

potencia_through_port_NBI1

VAccel_nominal_NBI2

IAccel_nominal_NBI2

tini_NBI2

longitud_pulso_nominal_NBI2

potencia_through_port_NBI2

potencia_nominal_NBI2

VAccel_real_NBI2

IAccel_real_NBI2

longitud_pulso_real_NBI2

updated_NBI2

potencia_nominal_ECRH2

potencia_depositada_ECRH2

anchura_haz_ECRH2

punto_deposicion_ECRH2

inyeccion_OnOff_axis_ECRH2

modulacion_ECRH2

angulo_polarizacion_lineal_ECRH2

angulo_toroidal_deposicion_ECRH2

angulo_1_ECRH2

angulo_2_ECRH2

n_paralelo_ECRH2

tini_ECRH2

longitud_pulso_nominal_ECRH2

fmod_ECRH2

rho_ECRH2

updated_ECRH2

longitud_pulso_real_ECRH2

PREVCOM

posicion_sonda_d4top

posicion_limitador_a3bot

posicion_limitador_c3bot

posicion_electrodo_a7top

posicion_sonda_b2bot

angulo_DR


2695-8864