

# TPJSONTOVIEW(3)

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<a href="#">1 SYNOPSIS</a>	<a href="#">1</a>
<a href="#">2 DESCRIPTION</a>	<a href="#">2</a>
<a href="#">3 RETURN VALUE</a>	<a href="#">3</a>
<a href="#">4 ERRORS</a>	<a href="#">4</a>
<a href="#">5 EXAMPLE</a>	<a href="#">5</a>
<a href="#">6 BUGS</a>	<a href="#">6</a>
<a href="#">7 SEE ALSO</a>	<a href="#">7</a>
<a href="#">8 COPYING</a>	<a href="#">8</a>

## Chapter 1

# SYNOPSIS

```
#include <atmi.h>
```

```
char * tpjsontoview(char *view, char *buffer)
```

For XATMI client link with *-latmiclt -latmi -lview -lnstd -lpthread -lrt -lm*

For XATMI server link with *-latmisrvl -latmisrvnomainl -latmisrvinteg -latmi -lview -lnstd -lpthread -lrt -lm*

---

## Chapter 2

# DESCRIPTION

Function converts JSON formatted text in *buffer* to VIEW buffer. JSON should be formatted in two levels. In first level there is single field named as VIEW name, and in second level JSON field names matches VIEW field name. If multiple occurrences/array of values is present in JSON, then it is converted in VIEW field array elements. The function in case of success returns XATMI (tpcalloc'ed) VIEW buffer, which is initialised to NULL values and then recovered from JSON.

For example following JSON:

```
{
  "MYVIEW2":
  {
    "tshort1":100,
    "tlong1":5,
    "tchar1":"",
    "tfloat1":1.10000,
    "tdouble1":0.000000,
    "tstring1":["HELLO WORLD", "Enduro/X", "HELLO WORLD"],
    "tarray1":"SGVsbG8gd29ybGQ="
  }
}
```

Can be loaded into following VIEW:

```
VIEW MYVIEW2
#type  cname      fbname      count  flag    size    null
short   tshort1    -           1      -       -       -
long    tlong1     -           1      -       -       -
char    tchar1     -           1      -       -       -
float   tfloat1    -           1      -       -       -
double  tdouble1   -           1      -       -       -
string  tstring1   -           3      -       15      -
carray  tarray1    -           1      -       10      -
END
```

Note that **carray** (binary data) fields should be present in base64 encoding in JSON buffer.

## Chapter 3

# RETURN VALUE

On success, **tpcall()** return pointer to XATMI allocated VIEW buffer; on error, NULL is returned, with **tperrno** set to indicate the error.

## Chapter 4

# ERRORS

Note that **tpsterror()** returns generic error message plus custom message with debug info from last function call.

**TPEINVAL** Invalid JSON or invalid base64 encoding, *view* is NULL or *buffer* is NULL.

**TPEOTYPE** Invalid type specified to function - *view* name specified in JSON message is not found in view files.

**TPESYSTEM** VIEW sub-system error, JSON sub-system error or Enduro/X environment not configured.

**TPEOS** System failure occurred during serving. See logs i.e. user log, or debugs for more info. In case of insufficient memory this error will be generated too.

## Chapter 5

# EXAMPLE

See `atmitest/test040_typedview/atmict40.c` for sample code.

## Chapter 6

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)

## Chapter 7

## SEE ALSO

[tpviewtojson\(3\)](#) [viewfile\(5\)](#) [viewc\(8\)](#) [ex\\_env\(5\)](#)

## **Chapter 8**

# **COPYING**

© Mavimax, Ltd