



PHP i5

#2

By SIRI

: uso del toolkit per IBMi





## ■ Agenda

- ✓ API's
- ✓ spool
- ✓ database



# I5 TOOLKIT: api's references

## Connection management

- i5\_connect
- i5\_close
- i5\_adopt\_authority
- i5\_error
- i5\_errno
- i5\_errormsg

## Command calls

- i5\_command

## Program calls

- i5\_program\_prepare
- i5\_program\_prepare\_PCML
- i5\_program\_call
- i5\_program\_close

## Data retrieval

- i5\_fetch\_array
- i5\_fetch\_assoc
- i5\_fetch\_object
- i5\_fetch\_row
- i5\_info
- i5\_field\_len
- i5\_field\_name
- i5\_field\_scale
- i5\_field\_type
- i5\_list\_fields
- i5\_num\_fields
- i5\_result

## Native file access

- i5\_open
- i5\_addnew
- i5\_edit
- i5\_delete
- i5\_cancel\_edit
- i5\_setvalue
- i5\_update
- i5\_range\_from
- i5\_range\_to
- i5\_range\_clear
- i5\_data\_seek
- i5\_seek
- i5\_bookmark
- i5\_free\_file
- i5\_new\_record
- i5\_update\_record
- i5\_get\_keys

# I5 TOOLKIT: api's references

## System values

- i5\_get\_system\_value

## Data areas

- i5\_data\_area\_prepare
- i5\_data\_area\_receive
- i5\_data\_area\_send
- i5\_data\_area\_close

## Print/Get spooled file

- i5\_spool\_list
- i5\_spool\_list\_read
- i5\_spool\_list\_close
- i5\_spool\_get\_data
- i5\_spool\_from\_file

## Job logs

- i5\_jobLog\_list
- i5\_jobLog\_list\_read
- i5\_jobLog\_list\_close

## Active jobs

- i5\_job\_list
- i5\_job\_list\_read
- i5\_job\_list\_close

## Objects list

- i5\_object\_list
- i5\_object\_list\_read
- i5\_object\_list\_close

## User space

- i5\_userspace\_create
- i5\_userspace\_prepare
- i5\_userspace\_get
- i5\_userspace\_put

## Data Queue

- i5\_dtaq\_prepare
- i5\_dtaq\_receive
- i5\_dtaq\_send
- i5\_dtaq\_close

# I5 TOOLKIT: i5\_connect()

i5\_connect(string server, string user, string password[, array options])

ritorna: un valido collegamento o [false] in caso di errore

Parametri:

nome del server (o IP)

Username

Password

opzioni

Esempio:

```
<?php
$conn = i5_connect ( "1.2.3.4", "MYUSER", "MYPWD" );
if (! $conn)
    die ( i5_errormsg () );
echo "Connection OK!<br>\n";
i5_close ( $conn );
```

# I5 TOOLKIT: i5\_connect()

Opzioni di connessione

**I5\_OPTIONS\_JOBNAME**

nome lavoro (nome iSeries di default)

**I5\_OPTIONS\_SQLNAMING**

abilita l'uso del punto (.) o slash (/) nella sintassi SQL

**I5\_OPTIONS\_DECIMALPOINT**

abilita l'uso del punto o virgola come separatore decimale

**I5\_OPTIONS\_CODEPAGEFILE**

abilita l'uso di uno specifico code page (CCSID)

**I5\_OPTIONS\_ALIAS**

(in caso di più iSeries abilita la connessione attraverso il nome connessione per differenziare da altro iSeries già collegato)

**I5\_OPTIONS\_INITLIBL**

indica la libreria iniziale in \*LIBL

# I5 TOOLKIT: i5\_command()

## **i5\_connect()**

connessione al sistema iSeries indicando nome sistema,  
utente e password di collegamento

## **i5\_command()**

chiamata a comando di sistema operativo

## **i5\_close()**

chiusura collegamento ad iSeries

# I5 TOOLKIT: i5\_command()

**i5\_command(string command[, array inputs, array outputs, resource connection])**

**ritorna: valore booleano se chiamata con successo**

## **Parametri:**

**nome comando secondo la sintassi "iSeries parameters".**

**Array dei parametri di input (è preferibile immettere tutto nella stringa del comando)**

**array dei valori di ritorno**

**connessione ad iSeries**



# I5 TOOLKIT: i5\_command()

```
<?php
i5_command ( "rtvjoba",
            array (),
            array( "curlib" => "curl",
                  "user" => "utente",
                  "usrlibl" => "userlib",
                  "syslibl" => array ("syslib",
                                     "char(165)")
                )
            );
print "User : $utente<br>";
print "User library : $userlib<br>";
print "System libs list : $syslib<br>";
print "Current library : $curl<br>";
```

# I5 TOOLKIT: call a programma

## `i5_connect()`

connessione al sistema iSeries indicando nome sistema, utente e password di collegamento

## `i5_program_prepare()`

impostazione di chiamata a programma RPG/COBOL con il setting dei parametri in una array

## `i5_program_call()`

chiamata effettiva a programma su iSeries passando i parametri e ricevendo eventuali valori di ritorno

## `i5_close()`

chiusura collegamento ad iSeries

# I5 TOOLKIT: i5\_program\_prepare()

`i5_program_prepare(string name[, array description][, resource connection])`

ritorna: resource connection se apertura con successo, [false] in caso di errore

## Parametri:

nome del programma

se si tratta di “service program” il nome va tra parentesi  
Lib/Service\_Program(PROC)

descrizione del paramtetri in formato php (attraverso array)

connessione a iSeries

# I5 TOOLKIT: i5\_program\_prepare()

L'array di valori è solitamente così definito:

nome del campo

tipo di campo che può essere:

I5\_TYPE\_CHAR

I5\_TYPE\_INT

I5\_TYPE\_PACKED

I5\_TYPE\_ZONED

I5\_TYPE\_FLOAT

I5\_TYPE\_BYTE

I5\_TYPE\_STRUCT

lunghezza

per CHAR, BYTE - numero

per PACKED, ZONED - numero,dec

per STRUCT - array che contiene la definizione della struttura

per INT, FLOAT - ignorato

Input/outpt

I5\_IN (default)

I5\_OUT

Count (opzionale) - ripetizione del campo nell'array

# I5 TOOLKIT: i5\_program\_prepare()

Preparazione dei parametri

i parametri vanno impostati attraverso un'array indicando, per ogni elemento il tipo e la lunghezza

Esempio:

```
$desc = array (  
    array ("name"=>"city", "io"=>I5_IN,  
          "type" => I5_TYPE_CHAR,"length" => "15"),  
    array ("name"=>"zip", "io"=>I5_IN,  
          "Type" => I5_TYPE_CHAR,"length" => "5"),  
    array ("name"=>"result", "io"=>I5_IN|I5OUT,  
          "type" => I5_TYPE_CHAR,"length" => "40")  
);
```

# I5 TOOLKIT: i5\_program\_call()

`i5_program_call(resource program, array params[, array retvals])`

ritorna: valore Booleano se chiamata con successo

## Parametri:

nome programma

array di valori come da definizione di “i5\_program\_prepare”

array di valori di output

# I5 TOOLKIT: i5\_program\_call()

```
$prog = i5_program_prepare ( "DEMOPGM" );  
    $result=i5_program_call($prog,  
        $param_in,  
        $param_out  
    );  
  
if (! $result)  
    die("Program call failed<br>");  
print "result is ".$param_out['valore']."<br>";
```



- spool

- ✓ lista spool
- ✓ lettura di uno spool





# I5 TOOLKIT: gestione spool

`i5_spool_list()`

crea una lista di spool di una certa outq o tutte

`i5_spool_list_read()`

legge gli spool di una outq

`i5_spool_get_data()`

legge i dati dallo spool

`i5_spool_list_close()`

libera risorse

# I5 TOOLKIT: i5\_spool\_list()

`i5_spool_list([array description][, resource connection])`

ritorna: resource connection se ok, altrimenti [false]

parametri:

- username che ha creato il job
- outq
- userdata (altri possibili parametri in secondo il formato os400)
- resource connection dell'iSeries

# I5 TOOLKIT: i5\_spool\_list\_read()

`i5_spool_list_read(resource spool_list)`

ritorna successivo spool nella lista, o [false] se lista vuota

il formato dello spool sarà SPLF0300.

vedi documentazione al seguente link:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/index.jsp?topic=/apis/QUSLSPL.htm>

*(in pratica quello che torna il comando WRKSPLF)*

## parametri:

resource connection ricevuta da `i5_spool_list`

# I5 TOOLKIT: i5\_spool\_get\_data()

```
i5_spool_get_data(string SPLFNAME, string JOBNAME,  
                 string USERNAME, integer JOBNBR, integer SPLFNBR [,string  
                 filename])
```

ritorna: valore stringa, [false] se la chiamata non ha esito positivo

parametri:

- nome del file di spool
- nome del job che ha creato lo spool
- username del job che ha creato lo spool
- numero del job che ha creato lo spool
- ID del file di spool nella coda (valore di ritorno da outq\_read)
- nome di file (IFS) che conterrà il risultato, se non indicato, il risultato sarà ritornato in stringa

# I5 TOOLKIT: i5\_spool\_list\_close()

`i5_spool_list_close(resource spool_list)`

**ritorna:** valore boolean se chiamata con successo

**parametri:**

resource connection ricevuta da `i5_spool_list`

# Esercizio 1

```
<?php
$spool = i5_spool_list ( array ("username" => "utente" ),
                        $conn );
if ($spool) {
    while ( $a = i5_spool_list_read ( $spool ) ) {
        echo "<br>=====<br>";
        var_dump ( $a ); // per debug
        $data = i5_spool_get_data ( $a ['SPLFNAME'],
                                    $a ['JOBNAME'],
                                    $a ['USERNAME'],
                                    $a ['JOBNBR'],
                                    $a ['SPLFNBR'] );

        var_dump ( $data );
    }
    i5_spool_list_close ( $spool );
} else
echo "Niente spool oggi.";
```

# Esercizio 1/bis

....per scrivere meno codice.....

```
<?php
for($line=strtok($result,"\n"); $line; $line=strtok("\n")) {
    echo ("  
<br>$line");
}
```

....e più righe di commento.....:)

## Esercizio 2

Creare lo script PHP che produca su una pagina html l'output del comando WRKACTJOB

Nota:

*Il comando non restituisce parametri diretti*

help:

- eseguire il comando con output su spool
- leggere lo spool e portarlo in html
- cancellare lo spool



# I5 TOOLKIT: area dati

`i5_data_area_create()`

crea un'area dati indicandone le dimensioni

`i5_data_area_read()`

legge il contenuto di un'area dati

`i5_data_area_write()`

sovrascrive il contenuto di un'area dati

`i5_data_area_delete()`

cancella un'area dati

# I5 TOOLKIT: i5\_data\_area\_create()

`i5_data_area_create(string nome, int size,  
resource connection)`

**ritorna:** valore boolean se chiamata con successo

**parametri:**

nome dell'area dati

lunghezza

resource connection

# I5 TOOLKIT: i5\_data\_area\_read()

```
i5_data_area_read(string nome, int pos.inizio,  
int num.byte, resource connection)
```

ritorna: contenuto dell'area dati se chiamata con successo,  
altrimenti, false  
(false anche nel caso l'offset sia errato)

## parametri:

nome dell'area dati

posizione iniziale

lunghezza di byte da leggere (-1 tutta l'area dati)

resource connection

*se non indicato l'offset, tutta l'area dati viene letta*

# I5 TOOLKIT: i5\_data\_area\_write()

`i5_data_area_write(string nome, string valore, int pos.iniz.,  
int num.byte, resource connection)`

ritorna: valore boolean se chiamata con successo

## parametri:

nome dell'area dati

valore da sovrascrivere

posizione iniziale

lunghezza di byte da scrivere (-1 tutta l'area dati)

resource connection

*se non indicato l'offset, tutta l'area dati viene  
sovrascritta*

# I5 TOOLKIT: i5\_data\_area\_delete()

`i5_data_area_delete(string nome, resource connection)`

ritorna: valore boolean se chiamata con successo

parametri:

nome dell'area dati

resource connection

# I5 TOOLKIT: i5\_get\_system\_value()

`i5_get_system_value(string nome, resource connection)`

ritorna: valore di sistema se chiamata con successo,  
altrimenti false.

parametri:

nome valore sistema

resource connection



- Database

- ✓ Accesso al DB con toolkit



# I5 TOOLKIT: i5\_open()

`i5_open (string fileName [, int mode ][,resource connection])`.

Ritorna: Handle se apertura con successo, [false] se non corretta

## Parametri:

nome del file (eventualmente libreria)

modo d'uso

I5\_OPEN\_READ - default

I5\_OPEN\_READWRITE

I5\_OPEN\_COMMIT

I5\_OPEN\_SHRRD

I5\_OPEN\_SHRUPD

I5\_OPEN\_SHRNUPD

I5\_OPEN\_EXCLRD

I5\_OPEN\_EXCL

resource connection



# I5 TOOLKIT: i5\_list\_fields()

i5\_list\_fields( resource file [di una open])

ritorna: array con lista campi

Parametri:

resource a apertura file

# I5 TOOLKIT: i5\_seek()

**i5\_seek ( resource file, int/string operator, array keyValue ).**

ritorna: valore booleano se riuscita operazione

Parametri:

resource a connessione

operatori di confronto

I5\_EQ "="

I5\_GT ">"

I5\_LT "<"

I5\_GE ">="

I5\_LE "<="

array di chiavi da confrontare

# I5 TOOLKIT: i5\_fetch\_row()

- array i5\_fetch\_row(resource file [di una open], modo lettura)
- recupero tutto il record da una [resource connection] e restituisce array di singoli valori
- ritorna: array valori record
- Parametri:
  - resource a apertura file
  - modo lettura
    - I5\_READ\_FIRST
    - I5\_READ\_LAST
    - I5\_READ\_NEXT
    - I5\_READ\_PRV

# I5 TOOLKIT: new e update record

**i5\_new\_record(resource file, array data).**

ritorna: valore booleano se corretto

parametri:

resource handle

array di valori in funzione del tracciato record

può essere un'array di valori o un'array di chiavi-valori

**i5\_update\_record(resource file, array data).**

ritorna: valore booleano se corretto

parametri:

resource handle

può essere un'array di valori o un'array di chiavi-valori

# I5 TOOLKIT: esempio

```
<?php
```

```
$file = i5_open ( "API/TESTFILE",i5_OPEN_READWRITE );  
$rec = i5_fetch_row ( $file, i5_READ_FIRST );  
i5_update_record ( $file,  
                  array ("CODE" => "C-02",  
                        "NOM" => "DUPONT",  
                        "TYPE" => 3  
                      )  
                );  
  
i5_new_record ( $file,  
              array ('C-105', 'DUPOND', 'Jean',  
                    'Avenue du Qubec', 'Les Ulis',  
                    3,'FR'  
                  )  
            );
```



# QUESTION TIME ?



Nome \_\_\_\_\_

Cognome \_\_\_\_\_

Data \_\_\_\_\_





ARRIVEDERCI



# TITOLO