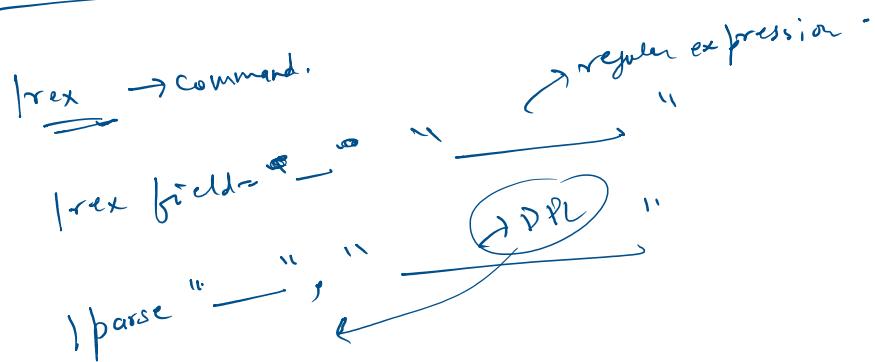


## DPL :- Dynatrace Pattern language-



## Dynatrace patter language:-

① Describe the pattern with matcher  
 → pattern that matches or certain type of data.

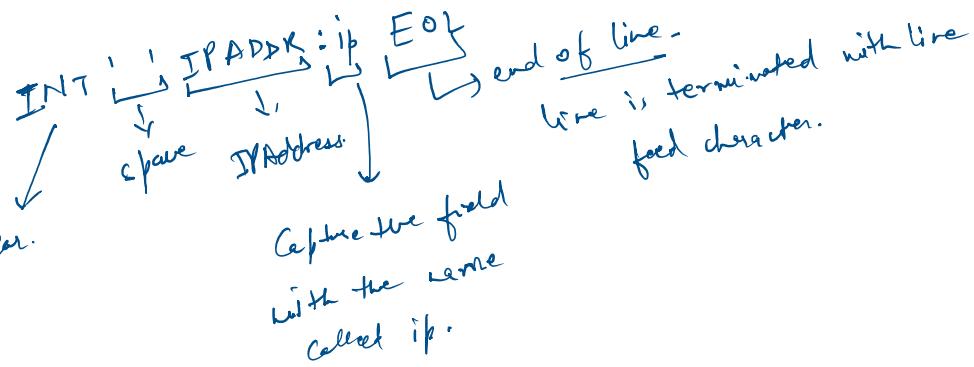
ex (INTEGER or INT) → Integer.

IPADDR → IP Address.

② Parse a record field into multiple output fields for better understanding or further processing.

③ Reshape incoming data for better understanding or further processing.  
 pattern structure work left to right ignore whitespace, linebreak & comments.

L . R



Matching vs parsing:- you don't necessarily need all the data element in the input data for analysis.

Matching vs Parsing:- you -  
input data for analysis.

Time field → HTTPDATE → timefield  
SPACE → " "  
L → Line Date.

Matcher Structure  
DPL :- DPL consist of one or more matcher exp.  
Separate the white space or commas or newline.

- (1) Built in.
- (2) liberal expression
- (3) character group.
- (4) Reference to another pattern expression.

Matching Group:-

- (1) Seq group → Ordered.
  - (2) Alternative group → list matches to choose from.
  - (3) Array → parse repeated data elements as composite type
  - (4) Structure → parse data as numeric value.
  - (5) Env group → match string to numeric value.
  - (6) Iron → parse Iron structure.
- DPL Architect:      use:
- (1) extract field from record
  - (2) create right data pattern to same line in developing DPL pattern.

Base Dataset

Match preview  
Dataset.

... . look ahead.

Kue

DPL Modifier

lookaround-

>> → positive look ahead.  
!>> → negative look ahead.  
<< → positive look behind.  
!<< → negative look behind.

optional modifier " ? "

field is missing →

date, severity,

+ 7

{  
}

f

venue, code  
a -

d e -

i - X

line del. ↗ field name  
LD ? : we name -  
optional modifier

DPL Literal Expression:-

"---" or '---'

a  
aa } → "a" + EOL.

DPL Lines & strings:-

① EOL → Line Break.

② CR → Single carriage return character.

③ LD or LDATA → Line Data - Any character until the next non-optimal matches in the scope of a line.

④ SGS → Single Quotes Value.

⑤ DQS → Double Quotes Value

⑥ character group - [char:]  
[0-9]  
!  
{ - }

[{\*} 0-9 {\*} a-z] {4,15} : user name  
username → user name as field name.  
Any character (lowercase, any digit)  
length b/w 4 to 15.

DPL Positional Match char

① Beginning of string [BOS, EOF] → BOS LD: header EOL;

② Middle of string [MOS, MOF] → MOS LD: Name \*

③ End of string [EOS, EOF] → EOS LD: name EOS

DPL Time & date

① ISO 8601 → yyyy-mm-ddTHH:mm:ssZ

② MySQL DATE → dd/MMM/yyyy HH:mm:ssZ

③ JSON/TIMESTAMP → yyyy-MM-ddTHH:mm:ss.sssZ

④ TIMESTAMP → Allows parsing time & date field in any format with millisecond precision

## \* DPL Numeric Data:-

① Boolean - Matches case insensitive, string as True or false.

② float -  $[0-9][+|-]$  (dot ". " separated)

③ Cfloat - same as float but comma separated.

④ Double : Matches floating point no. in the form  $[+|-][0-9]+\cdot[0-9]$

⑤ Double - same as double comma separated.

⑥ INT or INTEGER - Matches ~~integer~~ integral no. in the range of  
-247483648 to 247483647

⑦ HEXINT - Hexadecimal notation -  
 $(-184467\ldots)$  to  $18446794\ldots$ )

⑧ LONG. - ( $-184467\ldots$  to  $18446794\ldots$ )

⑨ HEXLONG - Integral no. in the hexadecimal notation.

## \* DPL Network Data:-

① IPADDR → Matches IPv4 & IPv6 addresses

② IPV4, IPV4ADDR → Matches IPv4 addresses.

↓ result

## (2) IPV4, IPV6

IPV4, IPV6 → Matches IPV6 address

(3) IPV6, IPV6ADDR → Matches valid credit card no.  
\* DPL credit card Data -  
ex - CREDITCARD : cc  
↓  
length, formatting, encoding etc.

\* DPL Unique Identifier → Matches valid Universally Unique Identifiers (UUIDs)  
Such as Social security no. Create UUID string parser.

\* DPL Alternative group → Multiple matches exp.  
(Left to right)

(ME | ME ---)  
When matching is done, it will exit if out.  
↓  
lazy match strategy.

\* DPL Array → Array allows parsing repeated seq. of variable no. data element.

The specified pattern is applied repeatedly until:-

① Unmatch occurs

② max. no. of matches has been reached

\* DPL ENUM :- Matching at the predefined string and convert them into respective assigned integer value.

\* DPL ENUM:— Map /  
respective assigned integer value.  
 ENUM {String : integer}  
 O/P → Integer.  
 CIS = true → match the string value case insensitively - Default = false.  
 Default → It's Case Sensitive.  
 CIS = true → its case insensitive.

### DPL Structure

Structure { matcher - expr }

↳ O/P - tuples

Allows capturing any sequence of matcher in tuple datatype.

Tuple → Ordered, immutable collection of elements.

stored, mixed datatype & allows duplicate.

Once it is created, you can't change, add or remove.

### Tomorrow :-

- ① DPL JSON object.
- ② DPL JSON Array.
- ③ DPL JSON Value.

### 8. Notebook

- ④ DPL XML Document.
- ⑤ DPL Key-value pair
- ⑥ DPL Pattern Expression Macros.
- ⑦ SSI - Apache, Winamp - Hands on.