Lapis Route Grammar

Based on https://github.com/leafo/lapis/blob/v1.5.0/lapis/router.moon#L158-L190 . The grammar below has been modified for clarity, but it should not have changed the pattern itself.

```
route             <- (chunk / literal)+

chunk             <- var / splat / optional

var               <- (':' alpha alpha_num*) class?
class             <- '[' [^]]+ ']'
splat             <- '*'

optional          <- '(' optional_route ')'
optional_route    <- (chunk / optional_literal)+
optional_literal  <- (!')' !chunk .)+

literal           <- (!chunk .)+

alpha             <- [a-zA-Z_]
alpha_num         <- [a-zA-Z_0-9]
```

Current Status of Error Reporting in Lapis Route Parser

Technically, there is no way of producing an error. Failures to properly match `vars` or `optionals` will simply be treated as `literals` instead. However, these failures are unlikely to be intentional, and so it would be better for them to be treated as errors instead.

Annotated Lapis Route Grammar

[pattern]^label means that the label is thrown upon failure of the pattern (not to be confused with character classes, which do not have the label).

```
route             <- (chunk / literal)+

chunk             <- var / splat / optional

var               <- (':' [alpha alpha_num*]^mis_name) class?
class             <- '[' [[^]]+]^mis_class [']']^mis_close_bracket
splat             <- '*'

optional          <- '(' [optional_route]^mis_route [')']^mis_close_paren
optional_route    <- (chunk / optional_literal)+
optional_literal  <- (!')' !chunk .)+

literal           <- (!chunk .)+

alpha             <- [a-zA-Z_]
alpha_num         <- [a-zA-Z_0-9]
```

Labels and Error Messages with Examples

1. **mis_name** - missing a variable name after the ':'
    a. /foo/:
    b. /foo/:/bar
2. **mis_class** - missing a character class after the '['
    a. /foo/:v[
    b. /foo/:v[]
3. **mis_close_bracket** - missing the closing ']'
    a. /foo/:v[%d
4. **mis_route** - missing an optional route after the '('
    a. /foo(
    b. /foo()/bar
5. **mis_close_paren** - missing the closing ')'
    a. /foo(/bar
    b. /foo(/:v(/bar)

Error Reporting & Recovery

Due to the fact that most routes are only a single line long, it was decided that error recovery would not be necessary. Error reporting prints the line and a pointer to the column with the error followed by the error message on the line below. It is assumed that the route is only a single line.

Lapis PostgreSQL Clause Grammar

Based on https://github.com/leafo/lapis/blob/v1.5.0/lapis/db/postgres.moon#L235-L288 . The grammar below has been modified for clarity, but it should not have changed the pattern itself.
Note: All string literal matches are case-insensitive.

```
grammar           <- ws* joins? clause*

joins             <- &start_join join_tuple+
join_tuple        <- start_join join_body
join_body         <- (balanced_parens / strings / !start_join !keyword .)+

start_join        <- join_type 'join'
join_type         <- ('natural' ws*)? (('inner' / outer_join_type) ws*)?
outer_join_type   <- ('left' / 'right' / 'full') (ws* 'outer')?

clause            <- keyword clause_content
clause_content    <- (balanced_parens / strings / !keyword .)+

balanced_parens   <- '(' (balanced_parens / strings / !')' .)*  ')'

keyword           <- (('order' ws+ 'by') / ('group' ws+ 'by')
                   / 'where' / 'having' / 'limit' / 'offset') ws*

string            <- single_string / double_string
single_string     <- "'" ("''" / !"'" .)* "'"
double_string     <- '"' ('""' / !'"' .)* '"'

alpha             <- [a-zA-Z_]
alpha_num         <- [a-zA-Z_0-9]
word              <- alpha_num+

ws                <- [ \t\r\n]
```

Current Status of Error Reporting in Lapis PostgreSQL Clause Parser

The parser will inform the user when parsing fails including the clause attempted to be parsed. However, the user is not given additional information about the error position or the specific error.

Annotated Lapis PostgreSQL Clause Grammar

[pattern]label means that the label is thrown upon failure of the pattern (not to be confused with character classes, which do not have the label). See the next page for more info.

```
grammar           <- ws* joins? clause*

joins             <- &start_join join_tuple+
join_tuple        <- start_join [join_body]exp_join_body
join_body         <- (balanced_parens / strings / !start_join !keyword .)+

start_join        <- join_type 'join'
join_type         <- ('natural' ws*)? (('inner' / outer_join_type) ws*)?
outer_join_type   <- ('left' / 'right' / 'full') (ws* 'outer')?

clause            <- keyword [clause_content]exp_clause
clause_content    <- (balanced_parens / strings / !keyword .)+

balanced_parens   <- '(' (balanced_parens / strings / !')' .)*  [')']mis_close_paren

keyword           <- (('order' ws+ ['by']exp_oby) / ('group' ws+ ['by']exp_gby)
                     / 'where' / 'having' / 'limit' / 'offset') ws*

string            <- single_string / double_string
single_string     <- "'" ("''" / !"'" .)* ["'"]mis_close_squote
double_string     <- '"' ('""' / !'"' .)* ['"']mis_close_dquote

alpha             <- [a-zA-Z_]
alpha_num         <- [a-zA-Z_0-9]
word              <- alpha_num+

ws                <- [ \t\r\n]
```

Labels and Error Messages with Examples

1. **exp_join_body** - expected a body after 'JOIN'
    a. join
    b. JOIN where id = 1
2. **exp_clause** - expected a clause after the keyword
    a. WHERE id = 1 LIMIT
    b. WHERE id = 1 LIMIT OFFSET 10
3. **mis_close_paren** - missing the closing ')'
    a. JOIN (t1, t2 GROUP BY date HAVING sum(qty) > 100
4. **exp_oby** - expected a 'BY' after 'ORDER'
    a. ORDER date, qty DESC
5. **exp_gby** - expected a 'BY' after 'GROUP'
    a. GROUP date
6. **mis_close_squote** - missing the closing single quote
    a. WHERE name = "John AND age > 18
7. **mis_close_dquote** - missing the closing double quote
    a. WHERE name = 'John AND age > 18

Error Reporting & Recovery

Similar to the route parser, error recovery was not implemented. Error reporting prints the line followed by the error message with the position of the error.

Limitations and Other Possible Improvements

- The errors of the route parser are only reported when trying to visit any page. It would be better if this could be done upon running the server. In practice, this is likely to be a non-issue since one would usually test at least one page before deployment anyways.
- Although there may be multiple routes with errors, only one of them will be reported. Lapis could be modified further to record these errors instead, but this is likely to be unnecessary since it is not often the case that many routes will have errors as the routes are often tested incrementally.
- The second error message of the clause parser is quite generic. It would be better if the actual keyword was specified.