



Ruby 101

Intermediate Ruby: Single Entry View

Intermediate Ruby:

- Display a single line:
- Organized for readability
- All data visible
- Informs user how to return



Intermediate Ruby:

```
view.rb
class LogEntryView < BasicView
  def quittance?
    false
  end
  ...
end
```



view.rb

```
def display_log_file
  entry = log_file.log_entries[log_file.log_entry_index]
  clear_display
  set_cursor
  print red(center("Displaying Entry #: #{log_file.log_entry_index + 1}"))
  set_cursor 3, 1
  print "Time:           \t#{entry.time_stamp.to_s}\n"
  print "IP:             \t#{entry.ip_address.to_s}\n"
  print "Request:          \t#{entry.request}\n"
  print "Response:         \t#{entry.response_code}\n"
  print "File Size:        \t#{entry.file_size} \n"
  print "HTTP Referer:     \t#{entry.http_referer}\n"
  print "User Agent:       \t#{entry.user_agent}\n"
  print "\e[J"
  set_cursor $stdin.winsize[0], 1
  print red("Esc to return to List View")
end
```



Intermediate Ruby:

view.rb

```
class LogListView
  ...
  def update
    ...
    print red("Esc to exit, up/down to move, 's' to sort or filter, Return
to see the full entry");
  end
  ...
end
```



Intermediate Ruby:

controller.rb

```
case user_input
  when "\n" , "\r"
    #carriage return / new line received
    case @current_view.class.to_s
      when "FileDialogView"
        file_dialog_select
      when "SortFilterView"
        apply_sort_filter
      when "LogListView"
        display_log_entry
    end
  ...
else
  case @current_view.class.to_s
    when "LogListView"
      sort_select if user_input == "s"
    when "LogEntryView"
      escape_log_entry if user_input == "\e"
```



Intermediate Ruby:

controller.rb

```
def display_log_entry
  @current_view = LogEntryView.new
  @current_view.display @log_file
end
def escape_log_entry
  @current_view = LogListView.new
  @current_view.display @log_file
end
```

