LinuxAcademy.com

Ruby 101

Intermediate Ruby: Log Parsing Program

Log Parser: Basic sketch of features:

1) Read apache log files
2) provide ways to sort and/or filter log entries
3) Display matching entries and scroll through them

Project Organization: MVC

Model / View / Controller

Model:

- Organizes Data and Handles Logic

View:

- Displays the data for the user

Controller:

- Handles user interaction
- Figures out what program logic to use
- Figures out what view to display
- Fetches the correct data from the model
- Sends it along to the appropriate view

Model Classes:

- Log entry: One line of a log file
  - Time Stamp
  - Remote IP Address / name
  - Local IP Address / name
  - Requested path
  - response code
- Log File:
  - File date
  - File name
  - File path
  - Array of Log Entries
  - Read File Method
  - Sort Data Method
  - Filter Data Method

View:

- File Dialog
- Table of Log Entries
- Menu:
    - Open a New File
- Sort / Filter
    - Columns and Direction
    - Field search

Controller:

- Keep track of current view
- Monitor up/down left/right keys
- Monitor for return key

File Layout:

- model.rb
- view.rb
- controller.rb
- logparser.rb

logparser.rb

```
require './model.rb'
require './view.rb'
require './controller.rb'
```

model.rb

```
class LogFile
end

class LogEntry
end
```

Intermediate Ruby:

view.rb

```
class BasicView
end
class FileDialogView < BasicView
    def display
        puts "Select an Apache log file."
    end
end
class LogListView < BasicView
end
class SortFilterView < BasicView
end
```

controller.rb

```
class LogParserController

    def initialize
        @log_file = LogFile.new
        @current_view = FileDialogView.new
    end
    def run
        while true do
            @current_view.display
            break
        end
    end
end
```

logparser.rb

```
...

@controller = LogParserController.new
@controller.run
```

Intermediate Ruby:

Iterative Programming:

Often called *Agile*, stresses the importance of building with multiple short interations with demonstratable results at each stage

DRY:

Do not Repeat Yourself: Make code reusable in different parts of the program

MVC:

Seems abstract at first: but helps organize the code
Used in many frameworks such as Rails
Requires a little extra code when working from scratch, so benefits are not immediate at the beginning
Just keep working along and feel free to ask questions.