



# Ruby 101

## Intermediate Ruby: Exceptions Revisited

## Intermediate Ruby:

If the user tries to load a non-conforming log file...

If the user doesn't have sufficient permissions...

If the user types weird stuff into the Filter fields...



## Intermediate Ruby:

We can catch the errors in the controller and inform the user.

We could display the message on the last line: where the information message goes.



## Errors.rb

```
class NotAnApacheAccessLog < RuntimeError
end
class NoFileAccess < RuntimeError
end
class NoDirAccess < RuntimeError
end
class InvalidDate < RunTimeError
end
```



logparser.rb

```
require 'io/console'
require './model.rb'
require './view.rb'
require './controller.rb'
require './errors.rb'
```

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

Intermediate Ruby:



controller.rb

```
def file_dialog_select
  begin
    case @log_file.select_directory_or_load_file
    when :directory
      @current_view.update @log_file
    when :file
      @current_view = LogListView.new
      @current_view.display @log_file
    end
    rescue NotAnApacheAccessLog
      @current_view.notice "File does not conform to Access Log
pattern"
    rescue NoFileAccess, NoDirAccess
      @current_view.notice "File or Directory Access Not Permitted"
    end
  end
end
```

## Intermediate Ruby:

model.rb

```
def cd path
  begin
    if Dir.exist?(path)
      @file_path = path
      @directory = Dir.new(@file_path)
      @directory_index = 0
      @list_start = 0
      true
    else
      false
    end
  rescue Errno::EACCES
    @file_path.gsub! /([^\/*\?])$/, ""
    raise NoDirAccess
  end
end
```



## Intermediate Ruby:

model.rb

```
def load_file
  begin
    if File.file?(@file_path + @directory.entries[@directory_index])
      @file_name = @directory.entries[@directory_index]
      log_array = IO.readlines(@file_path + @file_name)
      log_array.each_with_index do |log, index|
        @log_entries[index] = LogEntry.new log
      end
      @log_entry_index = 0
      @list_start = 0
      true
    else
      false
    end
  rescue Errno::EACCES
    raise NoFileAccess
  end
end
```



## Intermediate Ruby:

```
model.rb
def initialize row = nil
  if row
    begin
      row.gsub! /\t/, "    "
    rescue ArgumentError
      raise NotAnApacheAccessLog
    end
    match_data = parse_row row
    if match_data
      set_properties match_data
    else
      raise NotAnApacheAccessLog
    end
  end
end
```

IP Address:

ipaddr raises *IPAddr::InvalidAddressError*

```
def apply_sort_filter
  begin
    @log_file.log_entries = []
    @log_file.select_directory_or_load_file
    @log_file.sort_filter.apply_selections @log_file
    @current_view = LogListView.new
    @current_view.display @log_file
  rescue IPAddr::InvalidAddressError
    @current_view.notice "Please input a valid IP
address"
  end
end
```

Date:

...

```
if matches[5] != "**"  
  log_file.log_entries.select! do | entry |  
    entry.time_stamp.sec == matches[5].to_i
```

```
  end
```

```
end
```

```
else
```

```
  raise InvalidDate
```

```
end
```

```
end
```

Date:

...

```
rescue IPAddr::InvalidAddressError
  @current_view.notice "Please input a valid IP address"
rescue InvalidDate
  @current_view.notice "Please input a date and time 'MM-DD
HH:MM:SS ' "
end
```

```
class BasicView
```

```
...
```

```
  def notice message
```

```
    set_cursor $stdin.winsize[0], 1
```

```
    print "\e[K" + red(message)
```

```
  end
```

```
end
```