



Exam Next Friday (08/10/2021)

Revise the last this TD correct and the previous one
Revise Ruby Courses

Setup

1. launch the VM
2. launch Sublime Editor
3. launch a terminal
4. Use IRB to test some snippets of code

```
→ hugues$ irb
irb(main):001:0> puts "Hello, world!"
Hello, world!
=> nil
irb(main):002:0> exit
→ hugues $
```

Write the class in a file

```
# user.rb
class User
  # class constants
  ME_NAME = 'Hugues'
  ME_LASTNAME = 'Bernet-Rollande'

  # instance variables
  attr_accessor :firstname, :lastname

  # initialize
  def initialize(firstname:, lastname:)
    self.firstname = firstname
    self.lastname = lastname
  end

  # instance method
  def name
    "#{firstname} #{lastname}"
  end

  # class method
  def self.me
    new(firstname: ME_NAME, lastname: ME_LASTNAME)
  end
end
```

Use it in IRB

```
→ hugues$ irb
irb(main):001:0> load 'user.rb'
=> true
irb(main):002:0> User.new(firstname: "Steve", lastname: "Jobs").name
→ "Steve Jobs"
irb(main):002:0> User.me.name
→ "Steve Jobs"
```

HTTParty

```
gem install httparty
irb
> require 'httparty'
> result = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468" } )
> channels = result.parsed_response['channels']
```

Fetch a list of channels

```
require 'httparty'
authorization = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: authorization } )
channels = response.parsed_response['channels']
for channel in channels
  puts "#{channel['id']}: #{channel['name']}"
end
# => C02E24403SB: general
# ...
```

Fetch a list of messages

```
require 'httparty' # no need to re-include if already loaded
authorization = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
channel = 'C02E24403SB'
response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{channel}", headers: { Authorization: authorization} )
messages = response.parsed_response['messages']
for message in messages
  puts "#{message['ts']}: #{message['text']}"
end
# => 1632912974.000500: Hello
# ...
```

Post a message

```
require 'httparty' # no need to re-include if already loaded
authorization = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
channel = 'C02E24403SB'
text = "Hello World"
response = HTTParty.post("https://slack.com/api/chat.postMessage?channel=#{channel}&text=#{CGI.escape(text)}", headers: { Authorization: authorization} )
message = response.parsed_response['message']
puts "#{message['ts']}: #{message['text']}"
# => 1632912974.000500: Hello World
```


Update a message

```
require 'httparty' # no need to re-include if already loaded
authorization = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
ts = '1632920336.000100'
text = "Hello World, again!"
response = HTTParty.delete("https://slack.com/api/chat.postMessage?channel=#{channel}&text=#{CGI.escape(text)}&ts=#{ts}", headers: { Authorization: authorization} )
message = response.parsed_response['message']
puts "#{message['ts']}: #{message['text']}"
# => 1632920410.000300: Hello World
```

Delete a message

```
require 'httparty' # no need to re-include if already loaded
authorization = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
ts = '1631695747.001300'
response = HTTParty.delete("https://slack.com/api/chat.postMessage?channel=#{channel}&ts=#{ts}", headers: { Authorization: authorization} )
success = response.parsed_response['ok']
puts success
# => true
```

Write a slack client Class in a file `slack_client.rb` which allow to access the list of conversation

```
load 'slack_client.rb' # load the class written in the slack_client.rb
client = Slack::Client.new(authorization: "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468")
conversations = client.get_conversations
# puts conversations.inspect # optional to "inspect" the content of conversations
conversations.each do |conversation|
  puts "#{conversation['id']}: #{conversation['name']}"
end
# => C02E24403SB: general
# ...
```

```
# slack_client.rb
require 'httparty'
module Slack
  class Client
    attr_accessor :authorization

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization })
      response.parsed_response['channels']
    end
  end
end
```

Add a default client initialize with the correct authorization.

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversations.each do |conversation|
  puts "#{conversation['id']}: #{conversation['name']}"
end
# => C02E24403SB: general
# ...
```

```
# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization } )
      response.parsed_response['channels']
    end
  end
end
```

Add messages fetching to your Slack Client

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversation_id = conversations.first['id']
messages = client.get_messages(conversation_id: conversation_id)
# puts messages.inspect
messages.each do |message|
  puts "#{message['id']}: #{message['text']}"
end
# => 1632912974.000500: Hello
# => 1632912975.000500: World
```

```

# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization } )
      response.parsed_response['channels']
    end

    def get_messages(conversation_id:)
      response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{conversation_id}", headers: { Authorization: authorization } )
      response.parsed_response['messages']
    end
  end
end
end

```


Add message posting to your slack client

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversation_id = conversations.first['id']
message = client.post_message(conversation_id: conversation_id, text: 'Hello')
puts "#{message['id']}: #{message['text']}"
# => 1632912974.000500: Hello
```

```

# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization } )
      response.parsed_response['channels']
    end

    def get_messages(conversation_id:)
      response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{conversation_id}", headers: { Authorization: authorization } )
      response.parsed_response['messages']
    end

    def post_message(conversation_id:, text:)
      encoded_test = CGI.escape(text)
      response = HTTParty.post("https://slack.com/api/chat.postMessage?limit=50&channel=#{conversation_id}&text=#{encoded_test}", headers: { Authorization: authorization } )
      response.parsed_response['message']
    end
  end
end

```

Add message updating to your slack client

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversation_id = conversations.first['id']
message = client.post_message(conversation_id: conversation_id, text: 'Hello')
puts message.inspect
puts "#{message['id']}: #{message['text']}"
# => 1632912974.000500: Hello
message = client.update_message(conversation_id: conversation_id, message_id: message['ts'], text: 'World')
puts message.inspect
puts "#{message['id']}: #{message['text']}"
# => 1632912974.000500: World
```

```

# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization } )
      response.parsed_response['channels']
    end

    def get_messages(conversation_id:)
      response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{conversation_id}", headers: { Authorization: authorization } )
      response.parsed_response['messages']
    end

    def post_message(conversation_id:, text:)
      encoded_test = CGI.escape(text)
      response = HTTParty.post("https://slack.com/api/chat.postMessage?limit=50&channel=#{conversation_id}&text=#{encoded_test}", headers: { Authorization: authorization } )
      response.parsed_response['message']
    end

    def update_message(conversation_id:, message_id:, text:)
      encoded_test = CGI.escape(text)
      response = HTTParty.post("https://slack.com/api/chat.update?limit=50&ts=#{message_id}&channel=#{conversation_id}&text=#{encoded_test}", headers: { Authorization: authorization } )
      response.parsed_response['message']
    end
  end
end

```

Add message deleting to your slack client

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversation_id = conversations.first['id']
message = client.post_message(conversation_id: conversation_id, text: 'Hello')
puts message.inspect
puts "#{message['id']}: #{message['text']}"
# => 1632912974.000500: Hello
success = client.delete_message(conversation_id: conversation_id, message_id: message['ts'])
puts success
```

```

# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization } )
      response.parsed_response['channels']
    end

    def get_messages(conversation_id:)
      response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{conversation_id}", headers: { Authorization: authorization } )
      response.parsed_response['messages']
    end

    def post_message(conversation_id:, text:)
      encoded_test = CGI.escape(text)
      response = HTTParty.post("https://slack.com/api/chat.postMessage?limit=50&channel=#{conversation_id}&text=#{encoded_test}", headers: { Authorization: authorization } )
      response.parsed_response['message']
    end

    def update_message(conversation_id:, message_id:, text:)
      encoded_test = CGI.escape(text)
      response = HTTParty.put("https://slack.com/api/chat.update?limit=50&ts=#{message_id}&channel=#{conversation_id}&text=#{encoded_test}", headers: { Authorization: authorization } )
      response.parsed_response['message']
    end

    def delete_message(conversation_id:, message_id:)
      response = HTTParty.delete("https://slack.com/api/chat.delete?limit=50&ts=#{message_id}&channel=#{conversation_id}", headers: { Authorization: authorization } )
      response.parsed_response['ok']
    end
  end
end

```

Define a new class to encapsulate the conversations and messages Hashes

```
load 'slack_client.rb'
client = Slack::Client.default_client
conversations = client.get_conversations
conversations.each do |conversation|
  puts "#{conversation.id}: #{conversation.name}" # note the `.id` instead of conversation['id']
end
# => C02E24403SB: general
# ...
conversation = conversations.first
messages = conversation.get_messages
messages.each do |message|
  puts "#{message.id}: #{message.text}"
end
# => 1632912974.000500: Hello
# ...
message = conversation.create_message(text: 'Hello')
puts "#{message.id}: #{message.text}"
# => 1632912974.000500: Hello
message = message.update(text: 'World')
puts "#{message.id}: #{message.text}"
# => 1632912856.000300: World
message.delete
```

```

# slack_client.rb
require 'httparty'
module Slack
  class Client
    DEFAULT_AUTHORIZATION = "Bearer xoxp-2486113197334-2492860403907-2492926538098-76ac2d6b0dcc5d6a24b3c72889355468"
    attr_accessor :authorization

    def self.default_client
      new(authorization: DEFAULT_AUTHORIZATION)
    end

    def initialize(authorization:)
      self.authorization = authorization
    end

    def get_conversations
      response = HTTParty.get('https://slack.com/api/conversations.list?limit=50', headers: { Authorization: self.authorization })
      conversations = response.parsed_response['channels']
      # convert the array of conversation hashes in an array of conversation objects
      conversations.map do |conversation|
        Conversation.new(id: conversation['id'], name: conversation['name'])
      end
    end

    def get_messages(conversation_id:)
      response = HTTParty.get("https://slack.com/api/conversations.history?limit=50&channel=#{conversation_id}", headers: { Authorization: authorization })
      # convert the array of messages hashes in an array of message objects
      messages = response.parsed_response['messages']
      messages.map do |message|
        Message.new(id: message['ts'], text: message['text'], conversation_id: conversation_id)
      end
    end

    def post_message(conversation_id:, text:)
      encoded_text = CGI.escape(text)
      response = HTTParty.post("https://slack.com/api/chat.postMessage?limit=50&channel=#{conversation_id}&text=#{encoded_text}", headers: { Authorization: authorization })
      message = response.parsed_response['message']
      Message.new(id: message['ts'], text: message['text'], conversation_id: conversation_id)
    end

    def update_message(conversation_id:, message_id:, text:)
      encoded_text = CGI.escape(text)
      response = HTTParty.put("https://slack.com/api/chat.update?limit=50&ts=#{message_id}&channel=#{conversation_id}&text=#{encoded_text}", headers: { Authorization: authorization })
      response.parsed_response['message']
      message = response.parsed_response['message']
      Message.new(id: message['ts'], text: message['text'], conversation_id: conversation_id)
    end

    def delete_message(conversation_id:, message_id:)
      response = HTTParty.delete("https://slack.com/api/chat.delete?limit=50&ts=#{message_id}&channel=#{conversation_id}", headers: { Authorization: authorization })
      response.parsed_response['ok']
    end

    class Conversation
      attr_accessor :id, :name

      def initialize(id:, name:)
        self.id = id
        self.name = name
      end

      def get_messages
        client.get_messages(conversation_id: self.id)
      end

      def create_message(text:)
        client.post_message(conversation_id: self.id, text: text)
      end

      private

      def client
        Client.default_client
      end
    end

    class Message
      attr_accessor :id, :text, :conversation_id

      def initialize(id:, text:, conversation_id:)
        self.id = id
        self.text = text
        self.conversation_id = conversation_id
      end

      def update(text:)
        client.update_message(conversation_id: conversation_id, message_id: self.id, text: text)
      end

      def delete
        client.delete_message(conversation_id: conversation_id, message_id: self.id)
      end

      private

      def client
        Client.default_client
      end
    end
  end
end

```