

Rappel

- Pour charger un fichier ruby -> ***load 'fichier.rb'*** (attention à être dans le bon répertoire)
- ***HTTParty.get(url)*** permet de récupérer le contenu de la page ***url***
- ***HTTParty.get*** retourne un Objet ayant la méthode ***parsed_response*** qui transforme le JSON de la page en un Hash
- ***pp*** permet d'afficher le contenu d'un object (***pretty print***)
- Vous avez droit au cours, aux tds et internet (INFO-PWS-03-TD-CORRECTION.pdf)
- Code correctement formaté/indenté +2 points bonus

Test

Le test comporte 7 questions pour un total de 25 points.

Si vous bloquez sur une question, vous pouvez passer à la suivante (attention, la première question est indispensable pour continuer...)

Nous travaillerons avec l'API de Facebook (-> <https://developers.facebook.com/docs/graph-api/overview>), mais la documentation requise sera fournie.

Vous allez devoir écrire deux fichiers ruby:

1. ***facebook_client.rb*** contenant une classe permettant de communiquer avec l'API de facebook
2. ***facebook.rb*** contenant le code chargeant et utilisant la classe ci-dessus

Un fichier ***http.rb*** vous est fourni sur <http://1.101.fr>

Ouvrez une fenetre ***irb*** afin de tester votre code.

!!! A la fin du test vous devrez m'envoyer ces deux fichiers à partir de votre boite email
CELENE !!!
-> 0@101.fr

!!! Assurez-vous de pouvoir envoyer des emails dès maintenant !!!

IMPORTANT

Afin d'éviter tout problème réseau, nous utiliserons une **fausse** version de l'API de facebook.
Normalement pour travailler avec une API vous avez utilisé le gem **httparty**, ce gem est **simulé** par le fichier **http.rb**

Les 3 fichiers suivants sont disponibles sur le dossier **p:\\web dev test**

```
# http.rb
# ce fichier comporte une classe HTTParty
# que vous pouvez utiliser via
# HTTParty.get("https://graph.facebook.com/me/posts?fields=id,message,created_time&access_token=TOKEN")
```

```
# facebook_client.rb
load 'http.rb'
# écrivez la classe Facebook::Client ici
```

```
# facebook.rb
load 'facebook_client.rb'
# écrivez le code pour tester votre classe Facebook::Client ici
```

```
# irb
load 'facebook.rb'
```

L'API de Facebook permet de récupérer les posts/messages de l'utilisateur courant (celui associé à l'access token) via la requête suivante:

***https://graph.facebook.com/me/posts?
fields=id,message,created_time&access_token=TOKEN***

TOKEN correspond à l'access token de l'utilisateur courant.

**** Attention, l'API Facebook ne requiert pas de *authorization* (on lui passe déjà un ***TOKEN***)**

Exemple de réponse de l'API Facebook pour */me/posts*

```
{
  "data": [
    {
      "created_time": "2021-10-06T10:32:45+0000",
      "message": "Et encore un autre message",
      "id": "112125381237572_112155004567943"
    },
    {
      "created_time": "2021-10-06T10:32:37+0000",
      "message": "Un autre message",
      "id": "112125381237572_112154947901282"
    },
    {
      "created_time": "2021-10-06T10:06:39+0000",
      "message": "Ceci est un exercice",
      "id": "112125381237572_112140841236026"
    },
    {
      "created_time": "2021-10-06T09:48:35+0000",
      "message": "Hello, how are you?",
      "id": "112125381237572_112130777903699"
    },
    {
      "created_time": "2021-10-06T08:20:04+0000",
      "message": "Premiere publication",
      "id": "112125381237572_112077747909002"
    },
    {
      "created_time": "1982-03-01T08:00:00+0000",
      "id": "112125381237572_112075634575880"
    }
  ],
  "paging": {
    "previous": "...",
    "next": "..."
  }
}
```

1. (5 points) Ecrire une classe encapsulant l'API Facebook (dans ***facebook_client.rb***) pour récupérer les posts/messages de l'utilisateur.

- Tester votre classe via le code:

```
# facebook.rb
load 'facebook_client.rb'
client = Facebook::Client.new(access_token: "TOKEN")
posts = client.get_posts
pp posts
# `get_posts` DOIT retourner la liste des messages/posts sous la forme d'un tableau de hash
# voir le slide suivant pour le contenu de `posts`
```

```
# => posts
[
  {
    "created_time": "2021-10-06T10:32:45+0000",
    "message": "Et encore un autre message",
    "id": "112125381237572_112155004567943"
  },
  {
    "created_time": "2021-10-06T10:32:37+0000",
    "message": "Un autre message",
    "id": "112125381237572_112154947901282"
  },
  {
    "created_time": "2021-10-06T10:06:39+0000",
    "message": "Ceci est un exercice",
    "id": "112125381237572_112140841236026"
  },
  {
    "created_time": "2021-10-06T09:48:35+0000",
    "message": "Hello, how are you?",
    "id": "112125381237572_112130777903699"
  },
  {
    "created_time": "2021-10-06T08:20:04+0000",
    "message": "Premiere publication",
    "id": "112125381237572_112077747909002"
  },
  {
    "created_time": "1982-03-01T08:00:00+0000",
    "id": "112125381237572_112075634575880"
  }
]
```


1. (2 points) Mettez à jour ***facebook.rb*** pour afficher la liste des messages sous la forme:

CREATED_TIME: MESSAGE
CREATED_TIME: MESSAGE

...

Exemple

2021-10-06T08:20:04+0000: Premiere publication
2021-10-06T09:48:35+0000: Hello, how are you?

...

3.(2 points) Modifier votre classe ***facebook_client.rb*** pour ajouter une methode de classe ***default*** permettant de récupérer un client par default (avec un access token pré-renseigné)

- Tester votre classe via le code:

```
# facebook.rb  
load 'facebook_client.rb'  
client = Facebook::Client.default  
posts = client.get_posts  
pp posts
```

L'API Facebook permet aussi de récupérer des informations sur l'utilisateur courant via la requête

***https://graph.facebook.com/me?
fields=id,email,first_name,last_name&access_token=TOKEN***

- Attention, notez que la réponse n'inclut PAS de clé ***data*** mais directement les champs de l'utilisateur! *

Exemple de réponse de l'API Facebook pour ***/me***

```
{  
  "id": "112125381237572",  
  "email": "0@101.fr",  
  "first_name": "Hugues",  
  "last_name": "Bernet-Rollande"  
}
```

1. (3 points) Modifier votre classe ***facebook_client.rb*** pour ajouter une methode d'instance ***get_user_greeting*** retournant les informations de l'utilisateur formatées dans une chaine de caractères sous la forme "Hello, FIRSTNAME LASTNAME, you're email is EMAIL".
- Tester votre classe via le code:

```
# facebook.rb
load 'facebook_client.rb'
client = Facebook::Client.default
user_greeting = client.get_user_greeting
puts user_greeting
# => "Hello, Hugues Bernet-Rollande, you're email is 0@101.fr"
```

Pour tester si une chaîne de caractères en contient une autre, vous pouvez utiliser la méthode suivante:

```
"Ceci est un test".include?("test")  
# => true  
"Ceci est un test".include?("exercice")  
# => false
```

Pour définir une méthode prenant un argument optionnel/facultatif vous pouvez

```
def method(argument: nil)  
  if argument  
    puts "argument #{argument} fourni"  
  else  
    puts "argument non fourni"  
  end  
end  
puts method  
# => argument non fourni  
puts method(argument: '123')  
# => argument 123 fourni
```

1. (5 points) Modifier la méthode ***get_posts*** de votre classe ***facebook_client.rb*** pour supporter un argument nommé facultatif permettant de ne retourner QUE la liste des posts/messages qui contiennent un mot-clé (***keyword***) spécifique
- Tester votre classe via le code:

```
# facebook.rb
load 'facebook_client.rb'
client = Facebook::Client.default
posts = client.get_posts(keyword: "message")
pp posts
# contenu de posts dans le prochain slide
```

=> *posts*

```
[  
  {  
    "created_time": "2021-10-06T10:32:45+0000",  
    "message": "Et encore un autre message",  
    "id": "112125381237572_112155004567943"  
  },  
  {  
    "created_time": "2021-10-06T10:32:37+0000",  
    "message": "Un autre message",  
    "id": "112125381237572_112154947901282"  
  }  
]
```

L'API facebook permet de récupérer les commentaires (***comments***) d'un post/message en particulier en utilisant la requête

***https://graph.facebook.com/POST_ID/comments?
access_token=TOKEN***

avec ***POST_ID*** étant l'id d'un post/message

Exemple de réponse de l'API Facebook pour */POST_ID/comments*

```
{
  "data": [
    {
      "created_time": "2021-10-06T10:32:19+0000",
      "from": {
        "name": "Hugues Bernet-Rollande",
        "id": "112125381237572"
      },
      "message": "Salut les kids",
      "id": "112077747909002_112154734567970"
    },
    {
      "created_time": "2021-10-06T10:32:27+0000",
      "from": {
        "name": "Hugues Bernet-Rollande",
        "id": "112125381237572"
      },
      "message": "Comment ca va?",
      "id": "112077747909002_112154774567966"
    }
  ],
  "paging": {
    "cursors": {
      "before": "WTI5dGJXVnVkRjlqZAFhKemlzSTZANVEV5TVRVME56TTBOVFkzT1Rjd09qRTJNek0xTVRZAek5EQT0ZD",
      "after": "WTI5dGJXVnVkRjlqZAFhKemlzSTZANVEV5TVRVME56YzBOVFkzT1RZAMk9qRTJNek0xTVRZAek5EYz0ZD"
    }
  }
}
```

1. (3 points) Modifier votre classe ***facebook_client.rb*** pour ajouter une méthode ***get_comments(post_id:)*** qui permet de retourner les commentaires d'un message en particulier
- Tester votre classe via le code:

```
# facebook.rb
load 'facebook_client.rb'
client = Facebook::Client.default
posts = client.get_posts
post = posts.first
comments = client.get_comments(post_id: post['id'])
# contenu de `comments` sur le prochain slide
```

```
# => comments
[
  {
    "created_time": "2021-10-06T10:32:19+0000",
    "from": {
      "name": "Hugues Bernet-Rollande",
      "id": "112125381237572"
    },
    "message": "Salut les kids",
    "id": "112077747909002_112154734567970"
  },
  {
    "created_time": "2021-10-06T10:32:27+0000",
    "from": {
      "name": "Hugues Bernet-Rollande",
      "id": "112125381237572"
    },
    "message": "Comment ca va?",
    "id": "112077747909002_112154774567966"
  }
]
```

Pour obtenir le nombre d'éléments dans un tableau, utilisez la méthode ***count***

```
tableau = ['a', 'b', 'c']
```

```
puts "tableau contient #{tableau.count} éléments"
```

```
# => tableau contient 3 éléments
```

1. (5 points) Modifier votre classe ***facebook_client.rb*** pour ajouter une méthode ***display_feed*** qui affiche un résumé de toutes les informations de l'utilisateur (l'ordre des messages importe peu)

- Tester votre classe via le code:

```
# facebook.rb  
load 'facebook_client.rb'  
client = Facebook::Client.default  
client.show_feed
```

Hello, Hugues Bernet-Rollande, you're email is 0@101.fr

You've created 8 posts:

- *'Premiere publication' with 2 comment(s)*
- *'Hello, how are you?' with 2 comment(s)*
- *'Ceci est un exercice' with 2 comment(s)*
- *'Un autre message' with 2 comment(s)*
- *'Et encore un autre message' with 2 comment(s)*