OSS集会

#VRC_OSS集会

のりちゃん　2022/10/27

OSS集会

- みんなで**面白いもの、良いもの**を作りたい。
- プログラムを書くこと以外でもこの場で意見を出し合って、
  プロジェクトに参加できる場所にしたい。
- ソースコードを見ながらみんなで
  「ああでもない、こうでもない」って言いたい😂
- のりちゃんが火炙りにされつつ、**みんなで勉強できる**場所にしたい😄

OSS集会 前回の復習

Webページを動画プレイヤーで見れるようにしたい！

Webページ →URL→ Selenium → スクロール＆スクリーンショット

→画像を動画に変換→GCPにアップロード→動画のURLをWebページに返す

→URLを動画プレイヤーにセット

OSS集会 前回の復習

Webページを動画プレイヤーで見れるようにしたい！

Webページ →URL→ Selenium → スクロール＆スクリーンショット

→画像を動画に変換→GCPにアップロード→**ブラウザに動画を表示**

→URLを動画プレイヤーにセット

OSS集会 前回の復習と進捗

一般向けとOSS集会用どっちなの？
→ どっちもだけど、OSS集会向けにする！

READMEは初めにしっかり書きましょう！
→ 頑張る…

クラウド周りを切り離したほうがプロジェクトに参加しやすい。
→ 3つに分裂した…😅

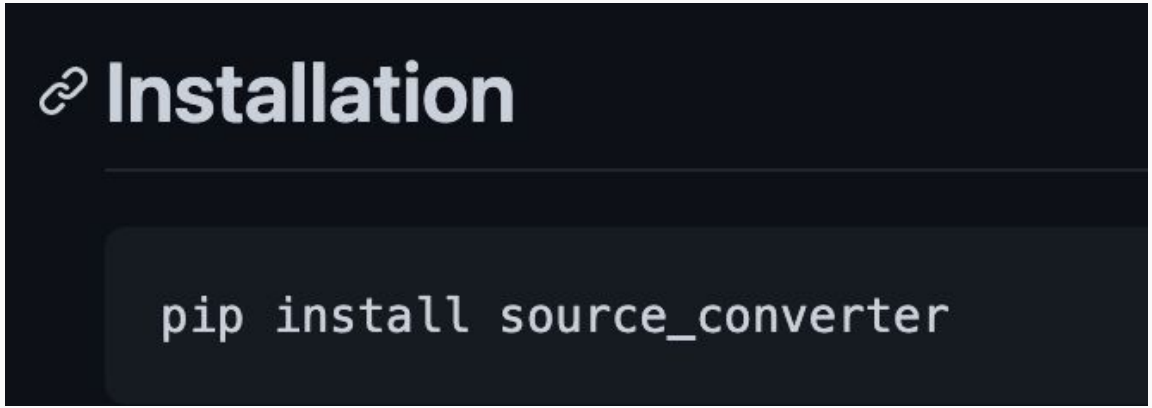
Twitterみたいな無限スクロールは大丈夫？
→最大値を設定するよ😀

進捗報告 Source Converter

GitHubからプロジェクトをZIPでダウンロードしてきて解凍。

指定のファイルをHTMLファイル化するモジュール。

https://github.com/noricha-vr/source_converter



進捗報告 Source Converter

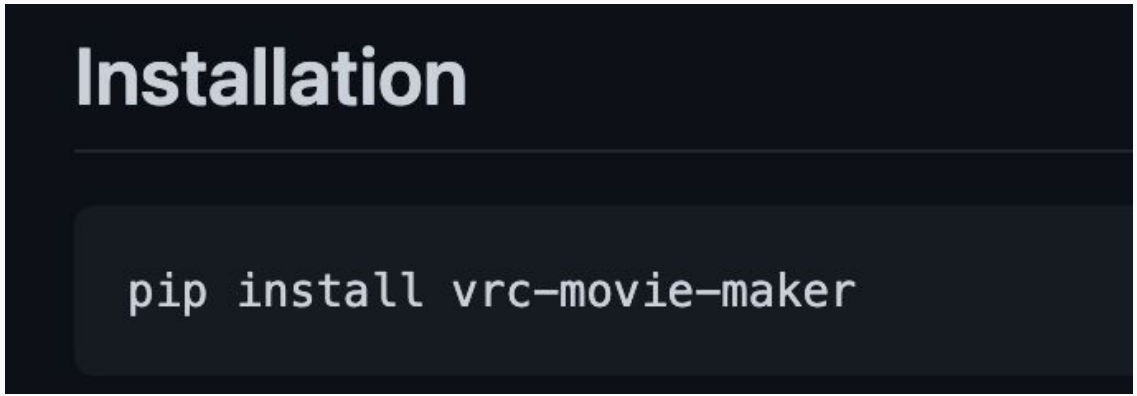
使い方

```
from github_downloader import GithubDownloader
from source_converter import SourceConverter

# Select the repository and file types.
url = "https://github.com/noricha-vr/source_converter"
targets = ['README.md', '*.py', ]

# Download the repository.
project_name = url.split("/")[-1]
folder_path = GithubDownloader.download_github_archive_and_unzip_to_file(url, project_name)
project_path = GithubDownloader.rename_project(folder_path, project_name)

# Convert the source codes to html files.
source_converter = SourceConverter('default')
html_file_path = source_converter.project_to_html(project_path, targets)
```

進捗報告 MovieMaker

URL、HTML、画像から動画を作るモジュール

https://github.com/noricha-vr/MovieMaker



進捗報告 MovieMaker

使い方

From web page URL.

```python
from movie_maker import MovieMaker, MovieConfig

# Please set the url, browser size and scroll.
url = "https://www.google.com/"
width = 1280
height = 720
limit_height = 50000
scroll_each = 200
movie_config = MovieConfig(url, width, height, limit_height, scroll_each)

# create movie
movie_maker = MovieMaker(movie_config)
movie_maker.create_movie()
```

From GitHub repository.

```python
from movie_maker import MovieMaker, MovieConfig

# Please set the repository URL and what types of file you want.
url = 'https://github.com/noricha-vr/source_converter'
targets = ['*.md', '*.py', ]
movie_config = MovieConfig(url, targets=targets)
# create movie
MovieMaker(movie_config).create_github_movie()
```

進捗報告 Screen Capture

Web ページをSeleniumで開く

↓

Seleniumでスクロール＆スクリーンショット

↓

GCSにアップロード
（認証情報がなければローカルの動画ファイルパスを返す）

↓

動画ファイルのURLを返す

進捗報告 Screen Capture

GitHubからリポジトリをダウンロード

↓

指定のファイルをHTML変換

↓（画像ファイルのパスを渡す）

Seleniumでスクロール＆スクリーンショット

↓

GCSにアップロード
（認証情報がなければローカルの動画ファイルパスを返す）

↓

動画ファイルのURLを返す

進捗報告 Screen Capture

https://screen-capture-7fhttuy7sq-an.a.run.app/

検証

どれくらいの維持費がかかるかな…？

23,016px → 5.9MB　1分56秒
20人→ 転送量118MB

Google Cloud からダウンロード19円/1GB
118MB →1回あたり 2.242円 → 1000回で2242円😅

進捗報告 Screen Capture

進捗報告 Screen Capture

画面共有

【クライアント】

1秒毎にデスクトップのスクショを撮って、POSTする。
（https://noricha-converter.com/screen-sharing/）

↓

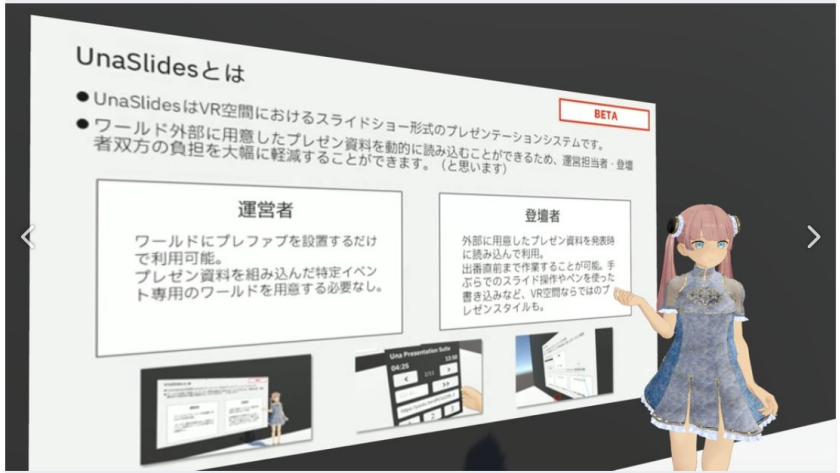
【サーバー】

画像を受け取って、動画化。
動画ファイルを上書きしてURLを固定
（https://noricha-converter.com/screen-sharing/<session_id>/）

↓↑ 1秒毎に動画を更新

【VRChat】動画プレイヤー

1秒毎にリロードできるプレイヤーがあります！！🎉



うな技研

UnaSlides - VRChat向けプレゼンシステム

♡ 152

無料ダウンロード ¥ 0
ダウンロード商品

無料ダウンロード
UnaSlides_0.4.1.zip (4.51 MB)

無料ダウンロード
UnaSlides_0.5.zip (4.56 MB)

❤開発支援❤（内容は同じ） ¥ 500
ダウンロード商品

カートに入れる

進捗報告 Screen Capture

ScreenCapture URL Image Desktop GitHub

ScreenCapture

VRChatにデスクトップを共有しよう！

1.このコードをターミナルに貼り付ける

```
while true;
do
    screencapture -x -t jpg /tmp/desktop.jpg
    curl -X POST -F "file=@/tmp/desktop.jpg" -H 'X-Token: 99ea0660-edad-450a-99f9-424154640073' http://0.0.0.
    sleep 3
done
```

2.このURLをVRChatの動画プレイヤーに貼り付ける
http://0.0.0.0:8080/api/desktop/99ea0660-edad-450a-99f9-424154640073

質問タイム

教えて！

強強エンジニア！

PowerShellだと…

全然シンプルじゃなかった(^o^;

post_screenshot.ps1 Raw

```powershell
function Get-ScreenCapture
{
    param(
    [Switch]$OfWindow
    )
    begin {
        Add-Type -AssemblyName System.Drawing
        $jpegCodec = [Drawing.Imaging.ImageCodecInfo]::GetImageEncoders() |
            Where-Object { $_.FormatDescription -eq "JPEG" }
    }
    process {
        Start-Sleep -Milliseconds 250
        if ($OfWindow) {
            [Windows.Forms.Sendkeys]::SendWait("%{PrtSc}")
        } else {
            [Windows.Forms.Sendkeys]::SendWait("{PrtSc}")
        }
        Start-Sleep -Milliseconds 250
        $bitmap = [Windows.Forms.Clipboard]::GetImage()
        $ep = New-Object Drawing.Imaging.EncoderParameters
        $ep.Param[0] = New-Object Drawing.Imaging.EncoderParameter ([System.Drawing.Imaging.Encoder]::Quality, [long]100)
        $screenCapturePathBase = "$pwd\ScreenCapture"
        $bitmap.Save("${screenCapturePathBase}.jpg", $jpegCodec, $ep)
    }
}




POST

##########
# パラメータ
##########
$uri = "http://192.168.1.1"
$filePath = $pwd\ScreenCapture.jpg
##########
# Body
##########
$fileName = [System.IO.Path]::GetFileName($filePath)
$boundary = '----Boundary'
$tempFile = './tempfile'
$UTF8woBOM = New-Object "System.Text.UTF8Encoding" -ArgumentList @($false)

$sw = New-Object System.IO.StreamWriter($tempFile, $false, $UTF8woBOM)
$sw.Write("--$boundary`nContent-Disposition: form-data; name=`"up_file`"; filename=`"$fileName`"`n")
$sw.Write("Content-Type: application/octet-stream`n`n")
$sw.Close()

$fs = New-Object System.IO.FileStream($tempFile, [System.IO.FileMode]::Append)
$bw = New-Object System.IO.BinaryWriter($fs)
$fileBinary = [System.IO.File]::ReadAllBytes($filePath)
$bw.Write($fileBinary)
$bw.Close()

$sw = New-Object System.IO.StreamWriter($tempFile, $true, $UTF8woBOM)
$sw.Write("`n--$boundary--`n")
$sw.Close()

##########
# Post
##########
Invoke-RestMethod -Method POST -Uri $uri -ContentType "multipart/form-data; boundary=$boundary" -InFile $tempFile
Remove-Item $tempFile
```

どっちがいいとかある？みんなはどっち派？

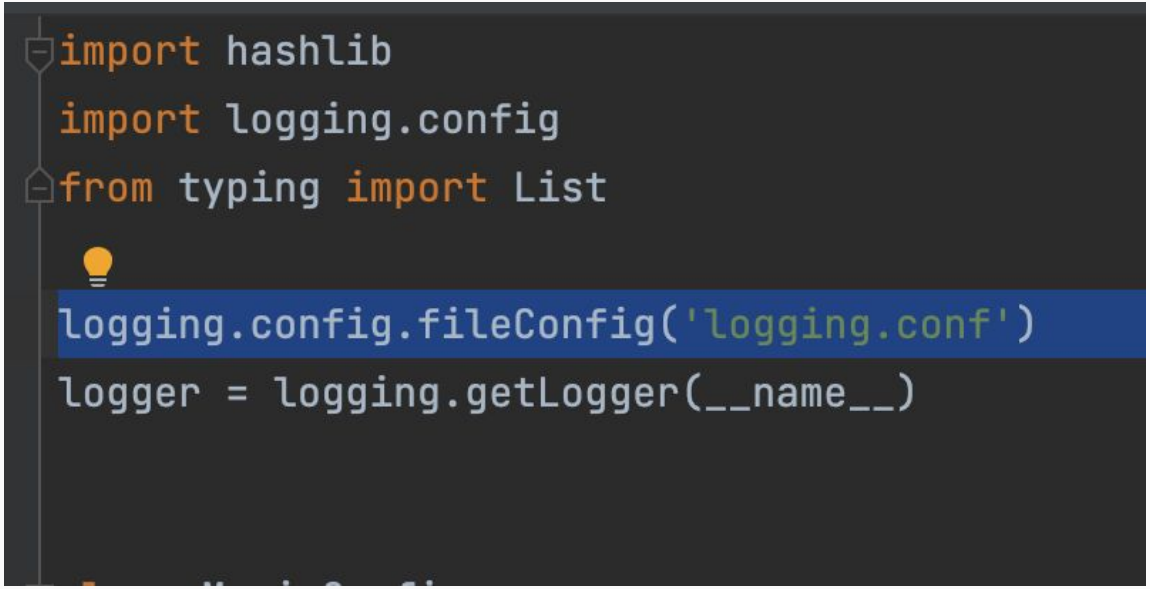
意味はないけど返り値で動画のパスを返す

```
def test_create_github_movie(self, url, targets):
    movie_config = MovieConfig(url, targets=targets)
    movie_path = MovieMaker(movie_config).create_github_movie()
    movie = editor.VideoFileClip(str(movie_path))
```

意味はないので返り値は使わない（持たない）

```
def test_create_github_movie(self, url, targets):
    movie_config = MovieConfig(url, targets=targets)
    MovieMaker(movie_config).create_github_movie()
    movie = editor.VideoFileClip(str(movie_config.movie_path))
```

ログの設定ファイル('logging.conf')って通常時とテストで2つ用意するもの…？



相談

FastAPIを使った時の

HTMLの分割とかコンテキスト（変数）の設定って何を使うの？

jinja が有名…？

MovieMakerを切り離す？

```python
@app.get("/create_movie/")
def create_movie(url: str, width: int = 1280, height: int = 720, limit_height: int =

    if len(url) == 0:
        raise HTTPException(status_code=400, detail="URL is empty.Please set URL.")
    bucket_manager = BucketManager(BUCKET_NAME)
    movie_config = MovieConfig(url, width, height, limit_height, scroll_each)
    if os.path.exists(movie_config.movie_path):
        url = bucket_manager.get_public_file_url(movie_config.movie_path)
        return RedirectResponse(url=url, status_code=303)
    movie_maker = MovieMaker(movie_config)
    movie_maker.create_movie()
    # Upload to GCS
    url = bucket_manager.to_public_url(movie_config.movie_path)
    return RedirectResponse(url=url, status_code=303)
```