

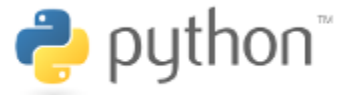
EITB NAHIERAN: ASKATU BIDEOAK API HONEN BIDEZ

by Mikel Larreategi



CodeSyntax

CodeSyntax



EITB APIA? ZERGATIK? (I)

- Zergatik ez?
- Gurea ere bada, ezta?
- XXI. mendean gaude jada...

ZERGATIK? (II)

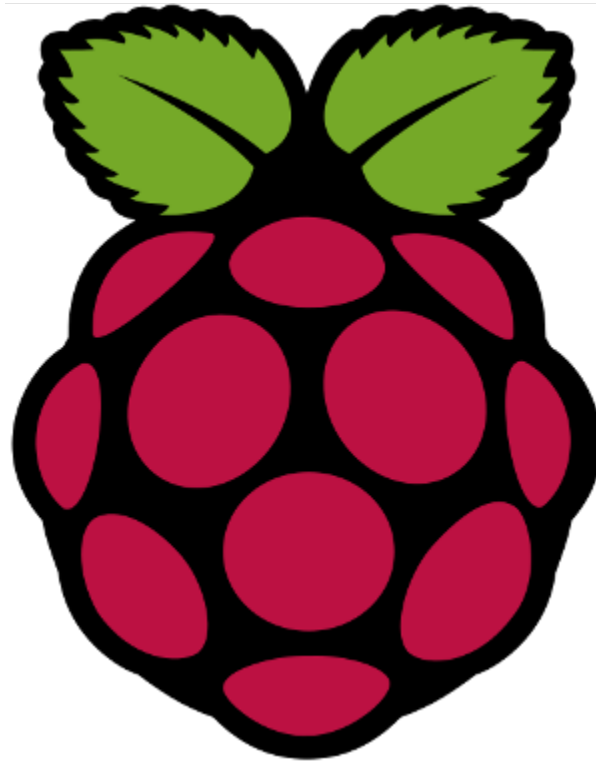
- Webgunea infumablea iruditzen zaidalako
- Ez dudalako lortzen ganoraz ezer ikustea
- Nabigatzailea kolgatzen didalako
- Flash erabiltzen duelako eta askotan ez zaidalako kargatzeko



Beno, badago beste arrazoi bat...

RASPBERRY PI

- Raspberry PI bat erosi nuen



KODI (LEHEN XBMC)



INSTALATU KODI

Deskargatu zure sistemarako

KODI (LEHEN XBMC)

- Youtube
- PlexBMC
- OpenSubtitles
- ...
- [tvalacarta](#)

EITB ETA TVALACARTA?

- Ba bai, tvalacartak bazeukan EITB ikusteko aukera, baina ez zebilen
- Webgune zaharrerako prestatuta zegoen
- Screenscraping egiten zuen

```

def mainlist(item):
    logger.info("[eitb.py] mainlist")
    itemlist=[]

    url = 'http://www.eitb.tv/es/'

    # Descarga la página
    data = scrapertools.cachePage(url)
    patron = "#CENSORED REGEX#"
    matches = re.compile(patron,re.DOTALL).findall(data)
    if DEBUG: scrapertools.printMatches(matches)

    for id,titulo,titulo2 in matches:
        scrapedtitle = titulo
        if titulo!=titulo2:
            scrapedtitle = scrapedtitle + " - " + titulo2
        scrapedurl = "http://www.eitb.tv/es/get/playlist/"+id
        scrapedthumbnail = ""
        scrapedplot = ""
        if (DEBUG): logger.info("title=["+scrapedtitle+"], url=["+sc

        itemlist.append( Item(channel=CHANNELNAME, title=scrapedtitl

    return itemlist

```

Programen zerrenda lortzeko kodea

```

def episodios(item):
    logger.info("[eitb.py] episodios")
    itemlist=[]

    # Descarga la página
    data = scrapertools.cachePage(item.url)
    logger.info(data)
    episodios_json = load_json(data)
    if episodios_json == None : episodios_json = []

    itemlist = []
    for video in episodios_json['videos']:
        scrapedthumbnail = video['thumbnailURL']
        if scrapedthumbnail is None:
            scrapedthumbnail = ""
        logger.info("scrapedthumbnail="+scrapedthumbnail)
        scrapedtitle = video['name']#.encode("utf-8",errors="ignore")
        #scrapedtitle = unicode( scrapedtitle , "iso-8859-1" , error

        scrapedplot = video['shortDescription']#.encode("utf8","igno
        try:
            scrapedtitle = video['customFields']['name_c']#.encode("
            scrapedplot = video['customFields']['shortdescription_c'
        except:
            pass
        scrapedurl = "http://www.eitb.tv/es/#/video/"+str(video['id']
        if (DEBUG): logger.info("title=["+scrapedtitle+"], url=["+sc
        itemlist.append( Item(channel=CHANNELNAME, title=scrapedtitle

```



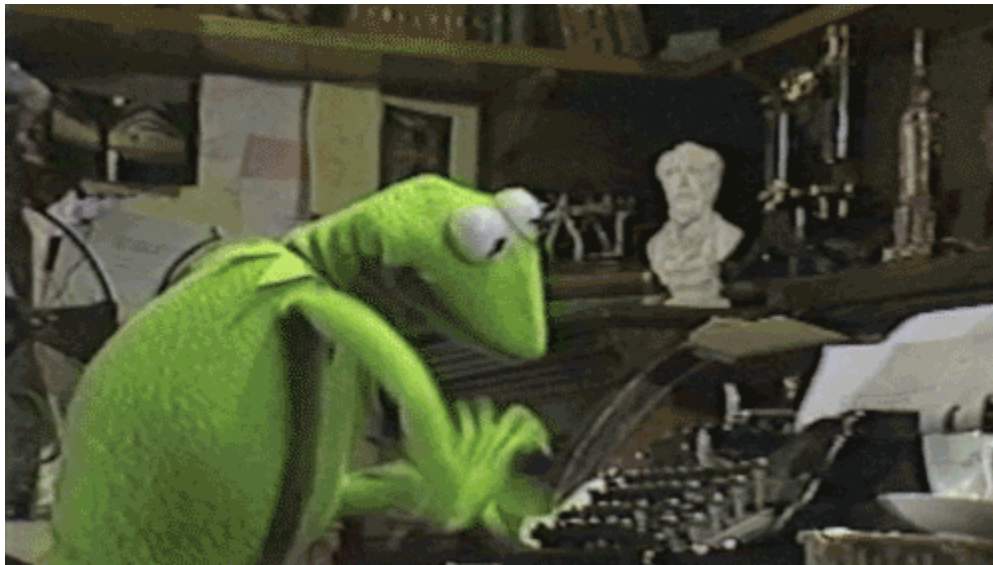
```
def get_video_url( page_url , premium = False , user="" , password=""  
    logger.info("[eitb.py] get_video_url(page_url='%s')" % page_url)  
  
    data = scrapertools.cache_page(page_url)  
  
    player_id = scrapertools.get_match(data, '<param name="playerID"  
    logger.info("player_id="+player_id)  
  
    player_key = scrapertools.get_match(data, '<param name="playerKey"  
    logger.info("player_key="+player_key)  
  
    # id del contenido, viene en la URL  
    # http://www.eitb.tv/es/#/video/1628880837001  
    video_player = scrapertools.get_match(page_url, 'http://www.eitb.  
    logger.info("video_player="+video_player)  
  
    response = get_rtmp( player_key , video_player , page_url , play
```

Bideo baten helbidea lortzeko "marabilak"

2016 URTEAN HORRELA IBILI BEHAR IZATEA ERE TB PUBLIKO BATEKIN...



BA NIK EGINGO DUT



Kontsola HTML CSS Script-ak DOM Sarea Cookie-ak					
Garbitu Jarraitu All HTML CSS JavaScript XHR Images Plugins Media Fonts					
URLa	Egoera	Protokoloa	Domelnu	Tamaina	Urruneko IPa
+ GET jquery.js	200 OK	http	eitb.tv	76.8 KB	84.53.132.35:80
+ GET eitb.constants.js	200 OK	http	eitb.tv	507 B	84.53.132.35:80
+ GET comun.js	200 OK	http	eitb.tv	20.0 KB	84.53.132.35:80
+ GET jquery.cookie.js	200 OK	http	eitb.tv	3.0 KB	84.53.132.35:80
+ GET jcookies.js	200 OK	http	eitb.tv	519 B	84.53.132.35:80
+ GET shadowbox.js	200 OK	http	eitb.tv	60.4 KB	84.53.132.35:80
+ GET cintillo.js	200 OK	http	eitb.tv	49 B	84.53.132.35:80
+ GET jquery.scrollTo.js	200 OK	http	eitb.tv	2.2 KB	84.53.132.35:80
+ GET jquery.serialScroll.js	200 OK	http	eitb.tv	2.0 KB	84.53.132.35:80
+ GET jquery.mouseWheel.js	200 OK	http	eitb.tv	1.3 KB	84.53.132.35:80
+ GET jquery.hoverIntent.minif	200 OK	http	eitb.tv	1.4 KB	84.53.132.35:80
+ GET jquery.tipsy.js	200 OK	http	eitb.tv	4.3 KB	84.53.132.35:80
+ GET jqbrowser-compressed.j	200 OK	http	eitb.tv	2.6 KB	84.53.132.35:80
+ GET video.js	200 OK	http	eitb.tv	174.2 KB	84.53.132.35:80
+ GET videojs.plugin.promos.js	200 OK	http	eitb.tv	15.7 KB	84.53.132.35:80
+ GET vast-client.js	200 OK	http	eitb.tv	68.3 KB	84.53.132.35:80
+ GET videojs.vast.js	200 OK	http	eitb.tv	18.4 KB	84.53.132.35:80
+ GET videojs.ga.min.js	200 OK	http	eitb.tv	2.5 KB	84.53.132.35:80
+ GET MSPlayerVPAIDManager.	200 OK	http	eitb.tv	20.7 KB	84.53.132.35:80
+ GET yumecb.js	200 OK	http	eitb.tv	20.3 KB	84.53.132.35:80
+ GET playerberria.js	200 OK	http	eitb.tv	15.0 KB	84.53.132.35:80
+ GET player_page.js	200 OK	http	eitb.tv	29.6 KB	84.53.132.35:80
+ GET menu.js	200 OK	http	eitb.tv	3.5 KB	84.53.132.35:80
+ GET video.js	200 OK	http	eitb.tv	23.4 KB	84.53.132.35:80
+ GET buscador.js	200 OK	http	eitb.tv	14.7 KB	84.53.132.35:80



Utzi baina....

youtube-dl ezagutu nuen arte

YOUTUBE-DL

- Youtubetik bideoak deskargatzeko aplikazioa
- Python pakete bat da
- Python paketeetatik erabili daiteke
- Eta ez da youtuberako bakarrik, EITBrekin funtzionatzen du



KODEA YOUTUBE-DL-REKIN

```
def get_video_url( page_url , premium = False , user="" , password=""  
    logger.info("[eitb.py] get_video_url(page_url='%s')" % page_url)  
  
    ydl = youtube_dl.YoutubeDL({'outtmpl': u'%(id)s%(ext)s'})  
    result = ydl.extract_info(page_url, download=False)  
    video_urls = []  
    if 'formats' in result:  
        for entry in result['formats']:  
            if 'http' in entry['format']:  
                video_urls.append([safe_unicode(entry['format']).encode('utf-8')])  
                logger.info('Append: {}'.format(entry['url']))  
  
    return video_urls
```



**ETA API BAT EGITEN
BADUT?**



EITB APIA

- Pyramid-en eginda
- Framework arin-arina
- Zuk aukeratu guztia (DB, template, ...)



PyramidTM

APIA PYRAMID-EN

- Misterio handirik ez
- 3 URL:
 - programa guztiak
 - programa baten saioak
 - saioa
- Plone-ren API lanean inspiratuta

ESAN ETA EGIN

Nabigatu APIan
Kodea ikusi

```
@view_config(route_name='programs', renderer='prettyjson')
def programs(request):
    """get all information about all the programs.
    How: scrap the website and look for the javascript links.
    """
    data = requests.get(EITB_FRONT_PAGE_URL)
    matches = re.compile(EITB_EPISODE_LIST_REGEX, re.DOTALL).findall(
        data.text

    result = {
        '@context': 'http://www.w3.org/ns/hydra/context.jsonld',
        '@id': request.route_url('programs'),
        '@type': 'TV',
        'parent': {},
    }

    results = []

    for id, title1, title2 in matches:
        scrapedtitle = title1
        if title1 != title2:
            scrapedtitle = scrapedtitle + " - " + title2

        results.append({
            '@id': request.route_url('playlist', playlist_id=id),
            '@type': 'Playlist',
            'title': scrapedtitle,
            'description': '',
        })
```

```
} )
```

```
result['member'] = results
```

```
return result
```

```

@view_config(route_name='playlist', renderer='prettyjson')
def playlist(request):
    """ get all the information about the given program.
        How: get the information from a pseudo-api
    """
    playlist_id = request.matchdict['playlist_id']

    result = {
        '@context': 'http://www.w3.org/ns/hydra/context.jsonld',
        '@id': request.route_url('playlist', playlist_id=playlist_id),
        '@type': 'Playlist',
        'parent': request.route_url('programs'),
    }

    playlist_url = EITB_PLAYLIST_BASE_URL.format(playlist_id)
    data = requests.get(playlist_url)
    playlist_data = data.json()
    web_medias = playlist_data.get('web_media')
    del playlist_data['web_media']

    playlist_data['member'] = []
    for web_media in web_medias:
        item = {
            '@id': create_internal_video_url(
                playlist_data.get('name_playlist'),
                playlist_data.get('id_web_playlist'),
                web_media.get('NAME_FILE')
            )
        }

```

```
        web_media.get('NAME_ES'),

        web_media.get('ID_WEB_MEDIA'),
        request=request,
    ),
    '@type': 'Episode',
    'title': web_media.get('NAME_ES'),
    'description': web_media.get('SHORT_DESC_ES', ''),
}
playlist_data['member'].append(item)
del playlist_data['id']

result.update(playlist_data)
return result
```

```
@view_config(route_name='episode', renderer='prettyjson')
def episode(request):
    """ Get all the information and the video links from a given episode
        How: use youtube-dl to get the information
    """
    episode_url = request.matchdict['episode_url']

    url = EITB_VIDEO_BASE_URL + episode_url

    playlist_title, playlist_id, video_title, video_id = episode_url.split('/')
    result = {
        '@context': 'http://www.w3.org/ns/hydra/context.jsonld',
        '@id': request.route_url('episode', episode_url=episode_url),
        '@type': 'Episode',
        'parent': request.route_url('playlist', playlist_id=playlist_id)
    }

    ydl = youtube_dl.YoutubeDL({'outtmpl': '%(id)s%(ext)s'})
    video_data = ydl.extract_info(url, download=False)

    result.update(video_data)

    return result
```

```

def clean_title(title):
    """slugify the titles using the method that EITB uses in
    the website:
    - url: http://www.eitb.tv/resources/js/comun/comun.js
    - method: string2url
    """
    translation_map = {
        'À': 'A', 'Á': 'A', 'Â': 'A', 'Ã': 'A', 'Ä': 'A', 'Å': 'A',
        'È': 'E', 'É': 'E', 'Ê': 'E', 'Ë': 'E',
        'Ì': 'I', 'Í': 'I', 'Î': 'I', 'Ï': 'I',
        'Ò': 'O', 'Ó': 'O', 'Ô': 'O', 'Ö': 'O',
        'Ù': 'U', 'Ú': 'U', 'Û': 'U', 'Ü': 'U',
        'Ñ': 'N', '?': '', '¿': '', '!': '',
        '¡': '', ':': '', '_': '-', '°': '',
        'ª': 'a', ',': '', '.': '', '(': '',
        ')': '', '@': '', ' ': '-', '&': ''
    }

    val = title.upper()
    for k, v in translation_map.items():
        val = val.replace(k, v)
    return val.lower()

```

ETA HONEKIN ZER?

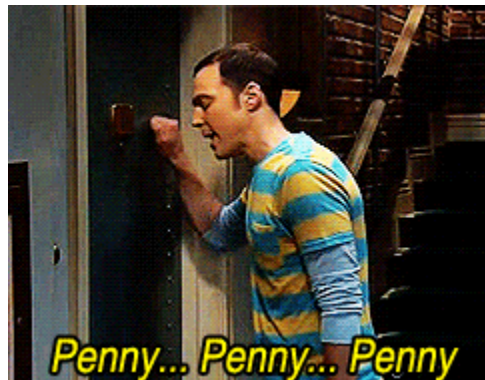
- Egin nahi duzuna
- Telebista ikusteko beste modu batzuk daude orain
- Kodi bat da, baina beste mila daude (chromecast, SmartTV, ...)
- Eman sarrera bideotekara era garbi baten
- EITBk ez badu egiten, besteok egingo dugu

INGURUKOEK BADUTE APIA

Eta garatzaileok gauzak egiteko aukera dugu

- RTVE
- BBC
- PBS
- NPR

Jendeak eskatzen badu egin beharko dute. Denok
joan beharko gara EITBra eskatzera



ESKERRIK ASKO

INFO GEHIAGO

APIa 

<https://still-castle-99749.herokuapp.com>

Kodea 

<https://github.com/erral/eitbapi>

Aurkezpena 

<https://erral.github.io/eitbapi>

Twitterren 

[@erralin](#)

GALDERARIK?



ETA EZ AHAZTU...

```
$ pip install gazta
```

CodeSyntax