s4gpy

Release 0.0.8

S4G Team

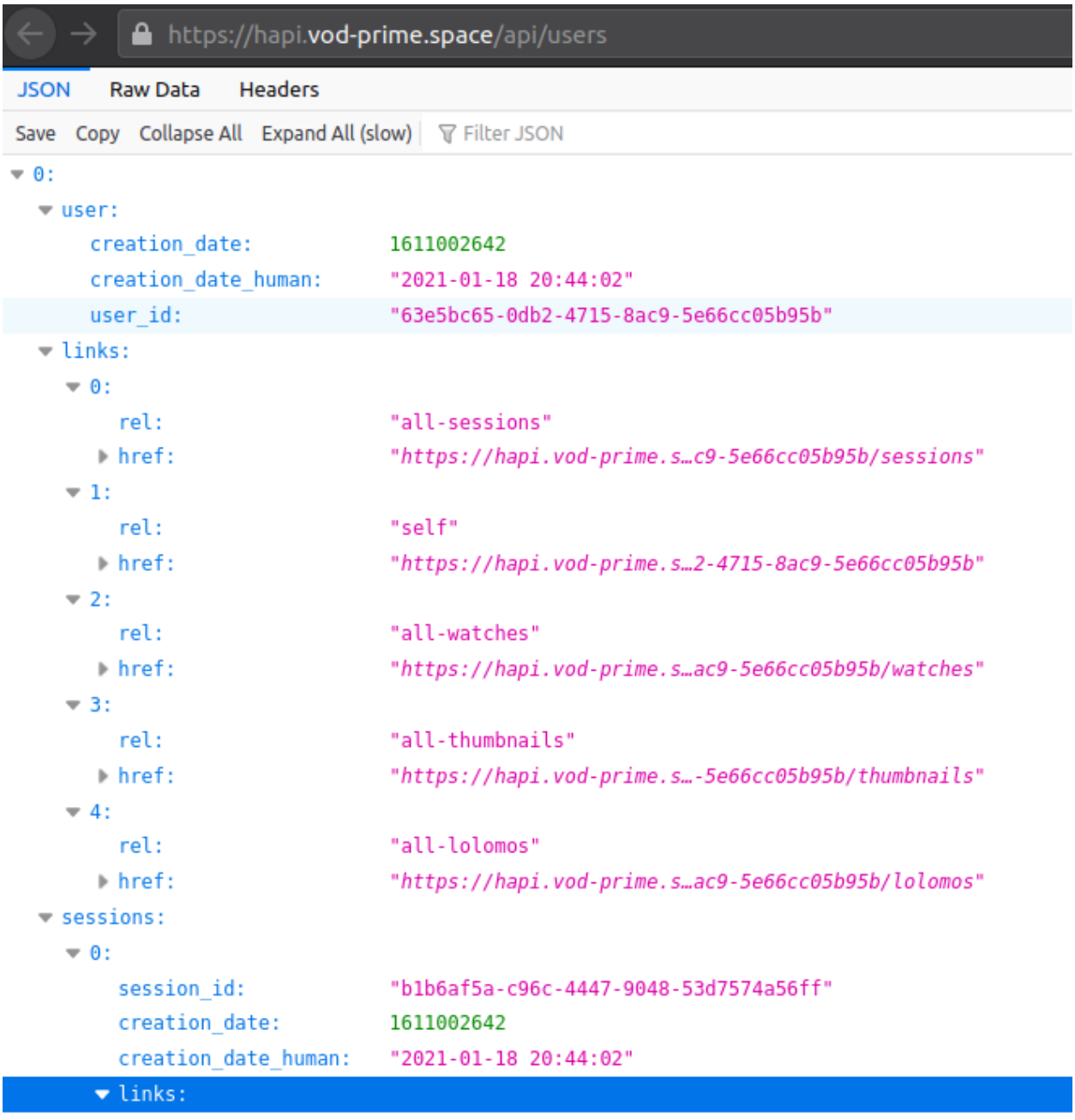
Apr 08, 2021

PROJECT DOCUMENTATION:

This client api can be used to perform analysis of S4G dataset with Python3.

It closely mirros the HATEAOS from the Rest API so that when requesting data, the `links` information are automatically converted to functions that can be called from the API.

For example with the user API



you can get a Python version with the following syntax:

```
from s4gpy.s4gpy import S4GAPI
#first register with your vod-prime.space credentials
api=S4GAPI("foo","bar")
#then get a user API object
user_api=api.get_user_api()
```

(continues on next page)

(continued from previous page)

```
#get the users from the API
for u in user_api.get_users():
    #for each user, follow the all-thumbnails link by calling the all_thumbnails()␣
↪function.
    for t in u.all_thumbnails()["thumbnails"]:
        print(f"{u.user.user_id};{t.row};{t.col};{t.video_id};{t.timestamp}")
```

this code prints out the users' id, and some metadata on thumbnails that were proposed to her.

```
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;4;1;81074110;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;4;0;80994082;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;6;0;81277950;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;5;2;80232398;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;5;3;80234304;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;4;3;80095697;1610984583.0
1f52213c-f31d-4dc9-bac5-35464b2ff1b9;5;0;80025678;1610984584.0
```

CHAPTER

ONE

INSTALLATION

1.1 From Pypi

```
pip install s4gpy
```

1.2 From source

```
#uninstall first
pip uninstall s4gpy
make build
pip install dist/*.whl
```

CHAPTER

TWO

EXAMPLES

2.1 Direct API

Get the current direct schedule, with metadata from the companion platform-api

```
from s4gpy.s4gpy import S4GAPI
api=S4GAPI(<add your user here>,<add your password here>)
for s in api.get_direct_api().get_direct_schedule():
    try:
        imdb_data=s.content().imdb_id()
        genres="+".join([g["genre"] for g in imdb_data.data.genres])
    except AttributeError: #in case platform.vod-prime.space fucks up things
        print(f"{s.airing_time};{s.video_id};UNKNOWN;UNKNOWN")
        continue
    print(f"{s.airing_time};{s.video_id};{imdb_data.data.title};f{genres}")
```

2.2 User API

Show the row/cols of every watched video for each user

```
from s4gpy.s4gpy import S4GAPI
#create an API entrypoint
api=S4GAPI("foo","bar")
#get the user_api
user_api=api.get_user_api()

#for each user
for u in user_api.get_users():
    #get all the video she watched
    watched_videos=[w.video_id for w in u.all_watches().watches]
    #for all the thumbnails
    for t in u.all_thumbnails()["thumbnails"]:
        #only dump the informations if the user has wached the video
        if t.video_id in watched_videos:
            print(f"{u.user.user_id};{t.row};{t.col};{t.video_id};{t.timestamp}")
```

2.3 Direct API

Get the current direct schedule, with metadata from the companion platform-api

```
from s4gpy.s4gpy import S4GAPI
api=S4GAPI(<add your user here>,<add your password here>)
for s in api.get_direct_api().get_direct_schedule():
    try:
        imdb_data=s.content().imdb_id()
        genres="+".join([g["genre"] for g in imdb_data.data.genres])
    except AttributeError: #in case platform.vod-prime.space fucks up things
        print(f"{s.airing_time};{s.video_id};UNKNOWN;UNKNOWN")
        continue
    print(f"{s.airing_time};{s.video_id};{imdb_data.data.title};f{genres}")
```

2.4 Credential API

Get some credentials for netflix to run a robot run

```
from s4gpy.s4gpy import S4GAPI
api=S4GAPI("foo","bar")
login, password = api.get_credentials_api().get_credentials("netflix")
```

2.5 Company Mapping API

Get/Set the mappings for content and company

```
from s4gpy.s4gpy import S4GAPI
from s4gpy.api.companyapi import CompanyAPI
from s4gpy.s4gsession import S4GSession
api = S4GAPI("foo","bar")

api.get_company_api().push_company("company1_cc_code",name="company 1 name",link=
↪"company 1 link")
api.get_company_api().push_company("company2_cc_code",name="company 2 name",link=
↪"company 2 link")
api.get_company_api().push_content("content_code",["company 1 name","company 2 name"])

for company in api.get_company_api().get_companies():
    print(f"company code {company.company_id} is {company.company().name}")
for content in api.get_company_api().get_contents():
    for company in content.content().companies():
        print(f"content {content.content_id} is produced by {company.company_id}")
```

CHAPTER

THREE

API

3.1 Direct

This API allows pushing and retreiving direct schedule for Netflix

class s4gpy.api.direct.**DirectAPI**(*session*)
This class wraps the direct API

get_direct_schedule()
access direct schedule for netflix fr

Returns netflix direct schedule

3.2 Credentials

This API allows getting credentials for supported service providers, to be used by robots for automatic data scrapping.

class s4gpy.api.credentials.**CredentialAPI**(*session*)
Wraps credential API.

get_credentials(*provider_name*)
Get a tuple containing credentials for the supplied provider.

get_providers()
get a tuple containing the names providers for which we have at least a credential available

3.3 Consommation

This API allows accessing user-generated consumption data.

class s4gpy.api.consoapi.**ConsoAPI**(*session*)
This class wraps the conso-api.

create_direct_schedule(*airing_time*, *video_id*)
Publish or updated a new direct schedule.

3.4 Netflix

This API provides access to content information published in the netflix platform

class s4gpy.api.netflixapi.**NetflixAPI**(*session*)
This class wraps the netflix-api.

get_root()
get the root of the netflix api

3.5 Users

This API provides access to users data.

class s4gpy.api.user.**UserAPI**(*session*)
This class wraps the user API

get_user(*user_id*)
get information about a particular user.

get_users()
returns a list of users.

3.6 Companies

This API allows accessing and pushing data about producing companies for a particular content.

class s4gpy.api.companyapi.**CompanyAPI**(*session*)
This class wraps the company-mapper api.

get_companies()
Get all the companies in the system

get_company(*id*)
retreives information from a particular company, identified by its cc_code

get_content(*id*)
retreives information form a particular content, identified by its cc_code

get_contents()
Get all the contents in the systel

get_root()
Get the root of the company mapping api

push_company(*id*, *name*, *link*, *country=None*)
Publish a new company.

push_content(*id*, *company_names*)
Publish a new content associated with some company names.

updated_company_country(*id*, *country*)
Updates a company country

PYTHON MODULE INDEX

S

INDEX