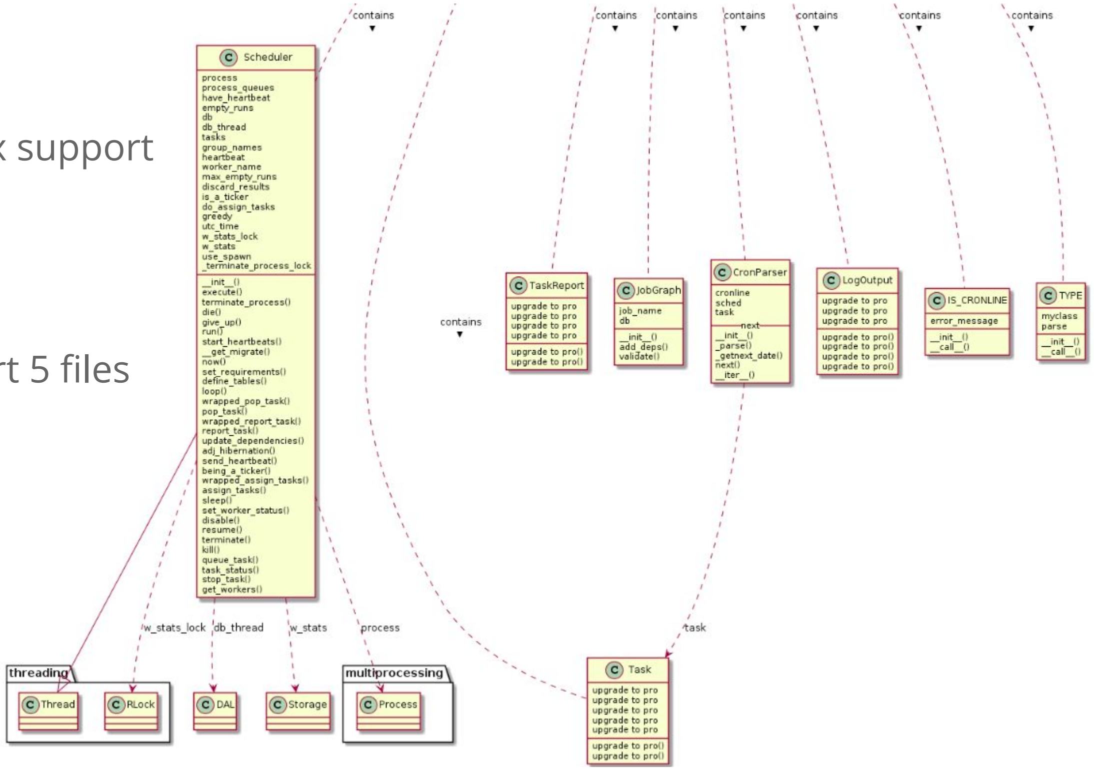
Tool sharing and code reading progress

SED 2019 Team 2
10/31

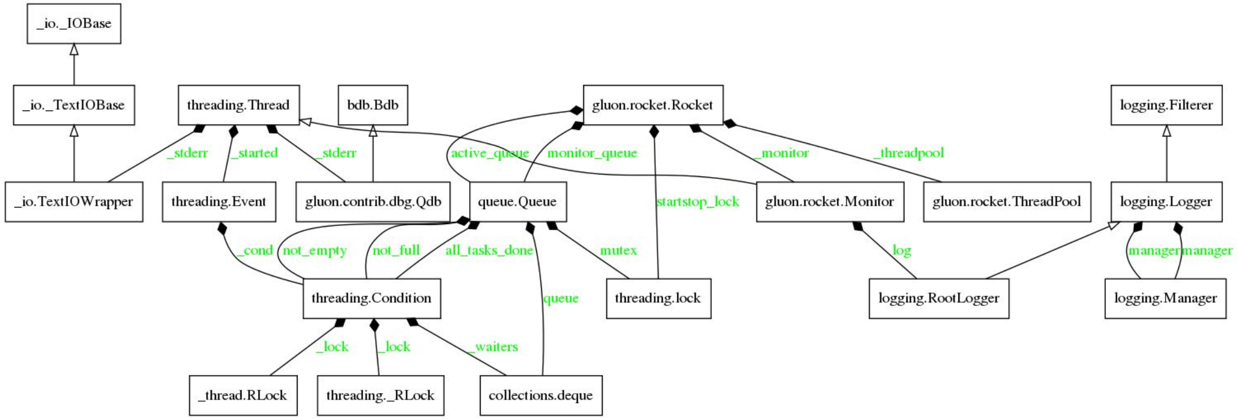
Pynsource

- Advantage
  - Python 3 and Python 2 syntax support
  - Update in 2019 Mar.
  - Can transform into PlantUML
- Disadvantage
  - Basic version can only support 5 files



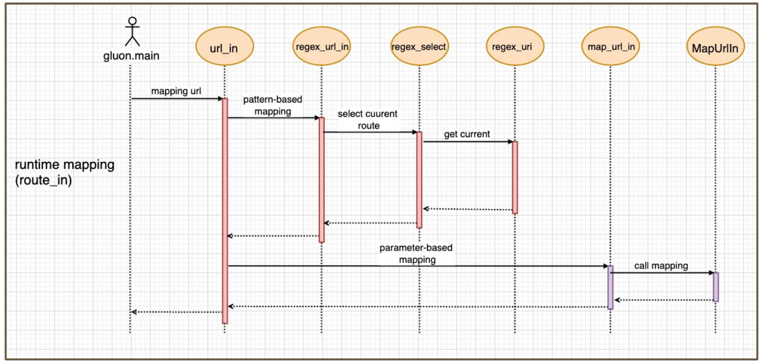
Pyreverse

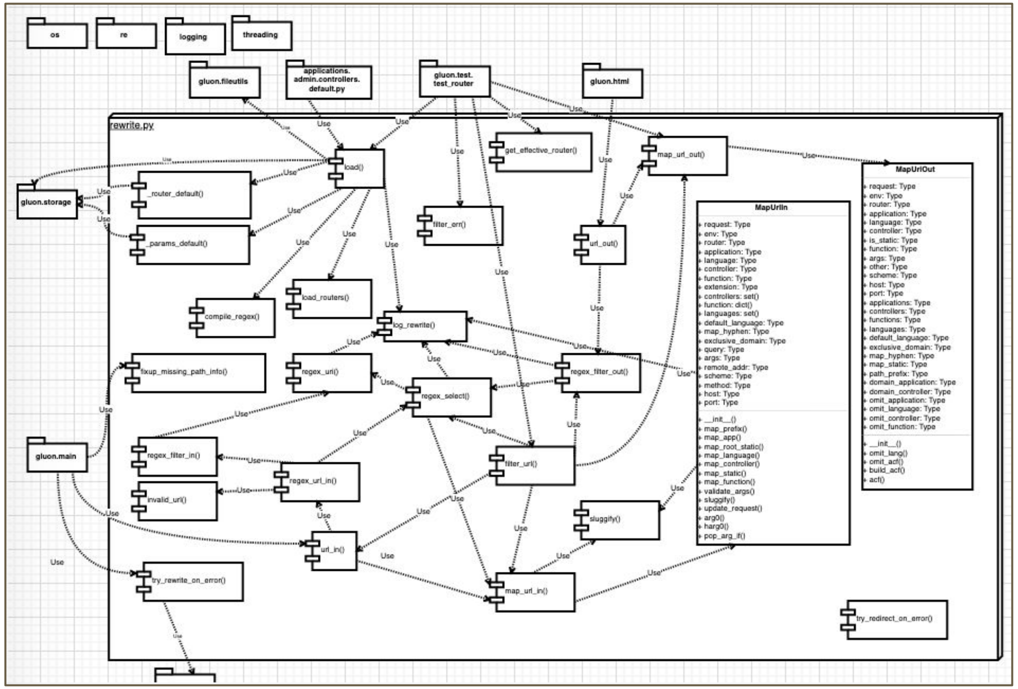
- Advantage
  - Free & open-source
- Disadvantage
  - Fail to detect some dependency
  - Not well-maintained. Produce errors when analyzing some specific files.



draw.io

- Advantage
  - can draw class diagram
  - can draw sequence diagram
  - co-editing
- Disadvantage
  - draw diagram by hand
  - very time-consuming



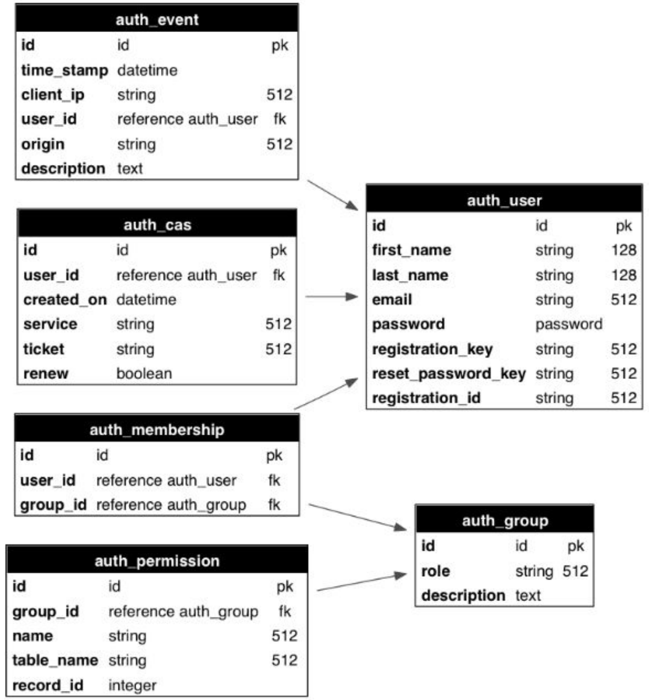


https://draw.io

code reading progress

authapi.py -- Access Control

- Role Based Access Control mechanism (RBAC)
  - Users are not assigned permissions directly, but only acquire them through their role (or roles), management of individual user rights becomes a matter of simply assigning appropriate roles to the user
- The web2py class that implements RBAC is called **Auth**



auth_event		
id	id	pk
time_stamp	datetime	
client_ip	string	512
user_id	reference auth_user	fk
origin	string	512
description	text	

auth_cas		
id	id	pk
user_id	reference auth_user	fk
created_on	datetime	
service	string	512
ticket	string	512
renew	boolean	

auth_membership		
id	id	pk
user_id	reference auth_user	fk
group_id	reference auth_group	fk

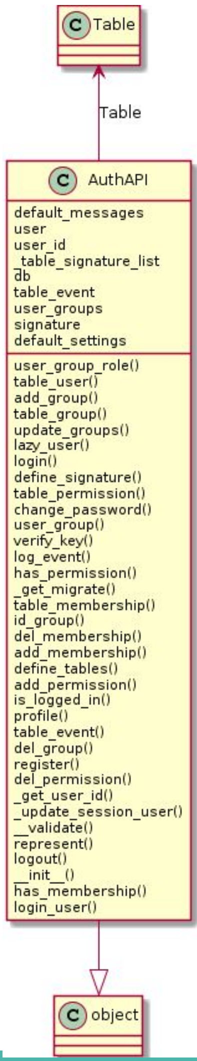
auth_permission		
id	id	pk
group_id	reference auth_group	fk
name	string	512
table_name	string	512
record_id	integer	

auth_user		
id	id	pk
first_name	string	128
last_name	string	128
email	string	512
password	password	
registration_key	string	512
reset_password_key	string	512
registration_id	string	512

auth_group		
id	id	pk
role	string	512
description	text	

authapi.py (cont.)

- # of lines: 1059
  - # of member function: 35
  - # of member variables: 8
- A barebones Auth implementation
- The main Auth functions are designed in a Dict In -> Dict Out logic
- No builtin CSRF protection whatsoever



authapi.py (cont.)

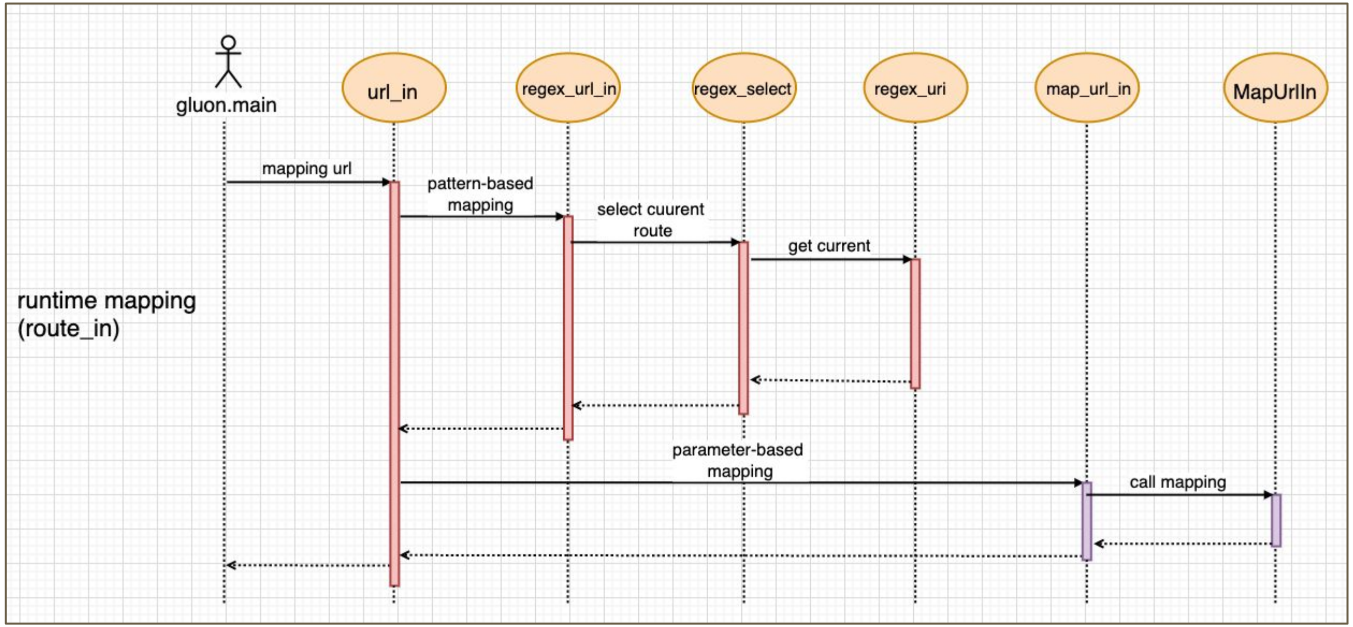
- **def __init__(self, db=None, hmac_key=None, signature=True)**
  - Depend on whether auth info is expired or not to renew the session
- **def define_signature(self)**
  - Initiate a auth_signature table and set the related attributes
- **def define_tables(self, username=None, signature=None, migrate=None, fake_migrate=None)**
  - defines all needed tables

rewrite.py

Notes:

1) only call when developer specify routes.py under their application or infrastructure
2) routes.py can be written in two different patterns
3) use THREAD_LOCAL for threads to deal with each request

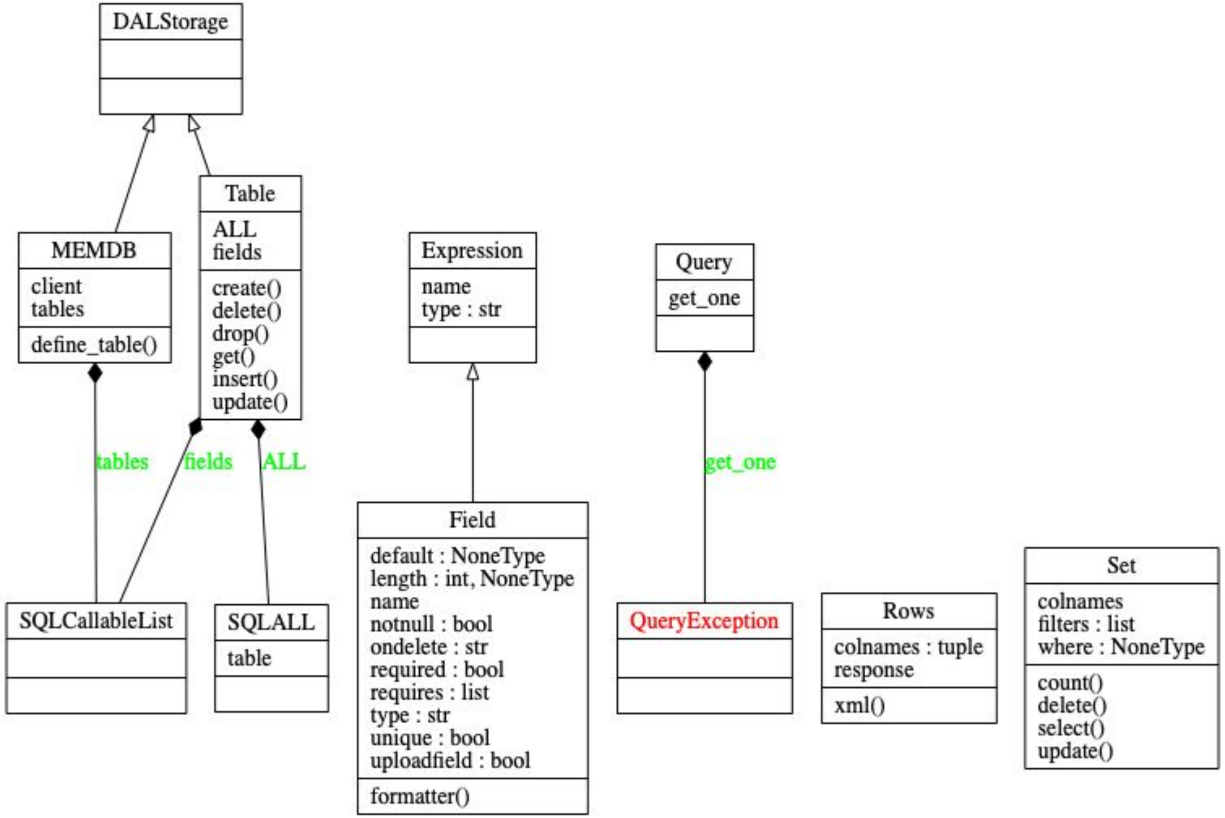
- what it does
  - route_in: (http://domain.com/xyz → http://domain.com/a/b/c)
  - route_out: (http://domain.com/a/b/c → http://domain.com/xyz)
- How it does



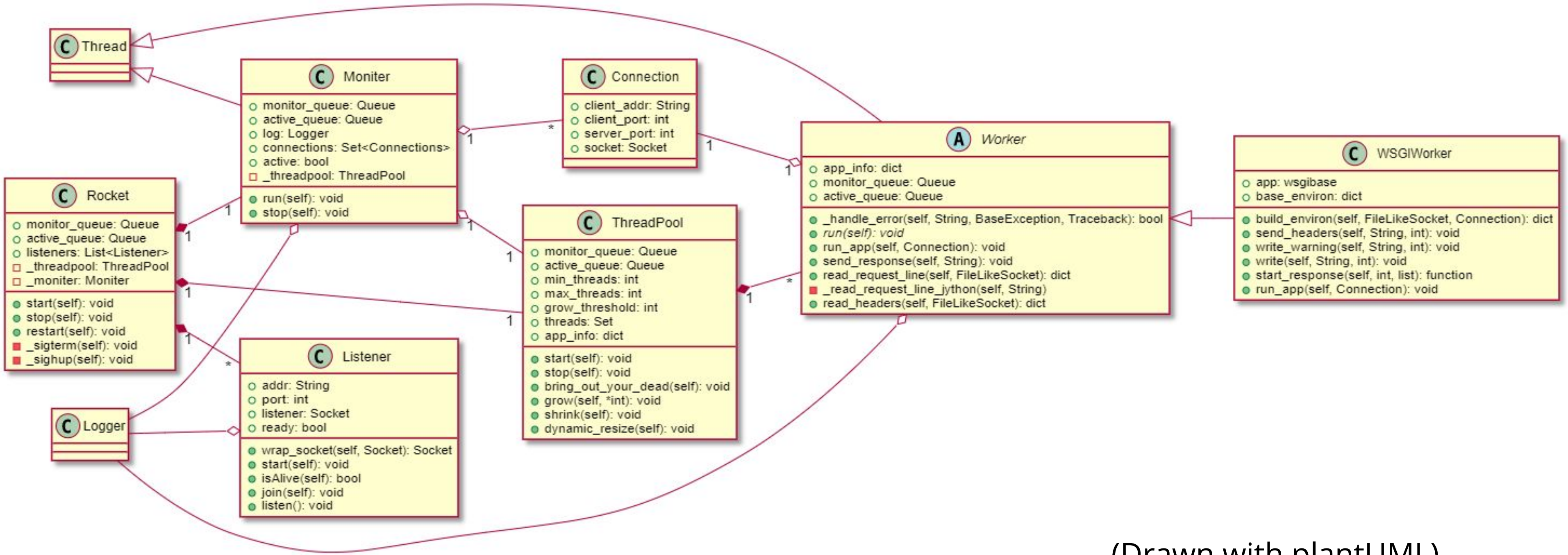
globals.py

- Contain classes for the global used variables
  - global variables and functions
  - current: store thread-specific data as attributes, usage across multiple files
  - classes: Request, Response, Session

memdb.py


DALStorage
MEMDB
client
tables
define_table()
Table
ALL
fields
create()
delete()
drop()
get()
insert()
update()
Expression
name
type : str
Query
get_one
tables
fields
ALL
get_one
Field
default : NoneType
length : int, NoneType
name
notnull : bool
ondelete : str
required : bool
requires : list
type : str
unique : bool
uploadfield : bool
formatter()
SQLCallableList
SQLALL
table
QueryException
Rows
colnames : tuple
response
xml()
Set
colnames
filters : list
where : NoneType
count()
delete()
select()
update()

rocket.py



(Drawn with plantUML)