CurrentUsers

July 28, 2020

In [1]:
```
def get_event_date(event):
  return event.date

def current_users(events):
  events.sort(key=get_event_date)
  machines = {}
  for event in events:
    if event.machine not in machines:
      machines[event.machine] = set()
    if event.type == "login":
      machines[event.machine].add(event.user)
    elif event.type == "logout":
      machines[event.machine].remove(event.user)
  return machines

def generate_report(machines):
  for machine, users in machines.items():
    if len(users) > 0:
      user_list = ", ".join(users)
      print("{}: {}".format(machine, user_list))
```

In [2]:
```
class Event:
  def __init__(self, event_date, event_type, machine_name, user):
    self.date = event_date
    self.type = event_type
    self.machine = machine_name
    self.user = user
```

In [3]:
```
events = [
    Event('2020-01-21 12:45:56', 'login', 'myworkstation.local', 'jordan'),
    Event('2020-01-22 15:53:42', 'logout', 'webserver.local', 'jordan'),
    Event('2020-01-21 18:53:21', 'login', 'webserver.local', 'lane'),
    Event('2020-01-22 10:25:34', 'logout', 'myworkstation.local', 'jordan'),
    Event('2020-01-21 08:20:01', 'login', 'webserver.local', 'jordan'),
    Event('2020-01-23 11:24:35', 'login', 'mailserver.local', 'chris'),
]
```

In [4]:
```
current_users(events)
```

```
Out[4]: {'webserver.local': {'lane'},
         'myworkstation.local': set(),
         'mailserver.local': {'chris'}}

In [7]: generate_report(current_users(events))

webserver.local: lane
mailserver.local: chris
```