Name- Vikas Bhartiya

Email- vikas6050@gmail.com

Website:- www.vikas6050.com

Mob- +91-9956961814

Good Day Sir,

I have answered below question with best of my ability and sincerely. Please find below.

Assignment: -

1. A developer is assigned a task to scrape 1 lakh website pages from a directory site, while scrapping he is facing such hcaptcha, which are placed to stop people from scrapping As a project Coordinator suggest ways to solve this problem.

   Ans:- Dear Sir there are most of the website which does not allow the web scarping so without use of the measures the scrapping will be blocked there are few method which I have identified to prevent blocking.

   A. Rotating the IP address- Multiple request coming from the same IP address can lead us to block so we need to use the multiple address. Create a pool of IP that can be use random one for each request. There are several method by which we can rotate the IP address e.g. VPN, TOR, free proxies, shared proxies etc.
   B. We can also scrap the website by making the virtual machine on cloud as these have different ip address and if machine get blocked we can create the new one.
   C. Website checks if you are the real browser, A real browser is necessary in most of the cases to scrap the data. There are libraries to automatically control such browser like Puppeteer, Selenium and Playwright.

D. There are the many low cheap paid services available in the market to solve the captcha problem like Octoparse, GSA Captcha Breaker, 2Captcha, Anti-Captcha, DeathByCaptcha etc

2. Our client has around 10k LinkedIn people profiles, he wants to know the estimated income range of these profiles. Suggest ways on how to do this?

   Ans:- First we have scrap the linkedin profile using the BeautifulSoup and Selenium web driver. Then using the soup.find or selenium driver.find_elemens we have to get the data of name profie id, company, year of experience and job stream and salary.

   Convert all the data into pandas dataframe

   After that we have to groupby LinkedIn profile by their job stream and country, aggregate their work experience as mean and the salary(if they provided in the Linkedin).

   If the salary is not provided the we have to see the average salary from the LinkedIn/job/salary section.

3. We have a list of 1L company names, need to find LinkedIn company links of these profiles, how to go about this?

   Ans:- company name- Accenture LinkedIn URL- https://www.linkedin.com/company/accenture/

   company name- american-express LinkedIn URL- https://www.linkedin.com/company/american-express/

   company name- radiansys-inc LinkedIn URL- https://www.linkedin.com/company/radiansys-inc/

   So here I have taken few company and their URL and can see the pattern on their URL.

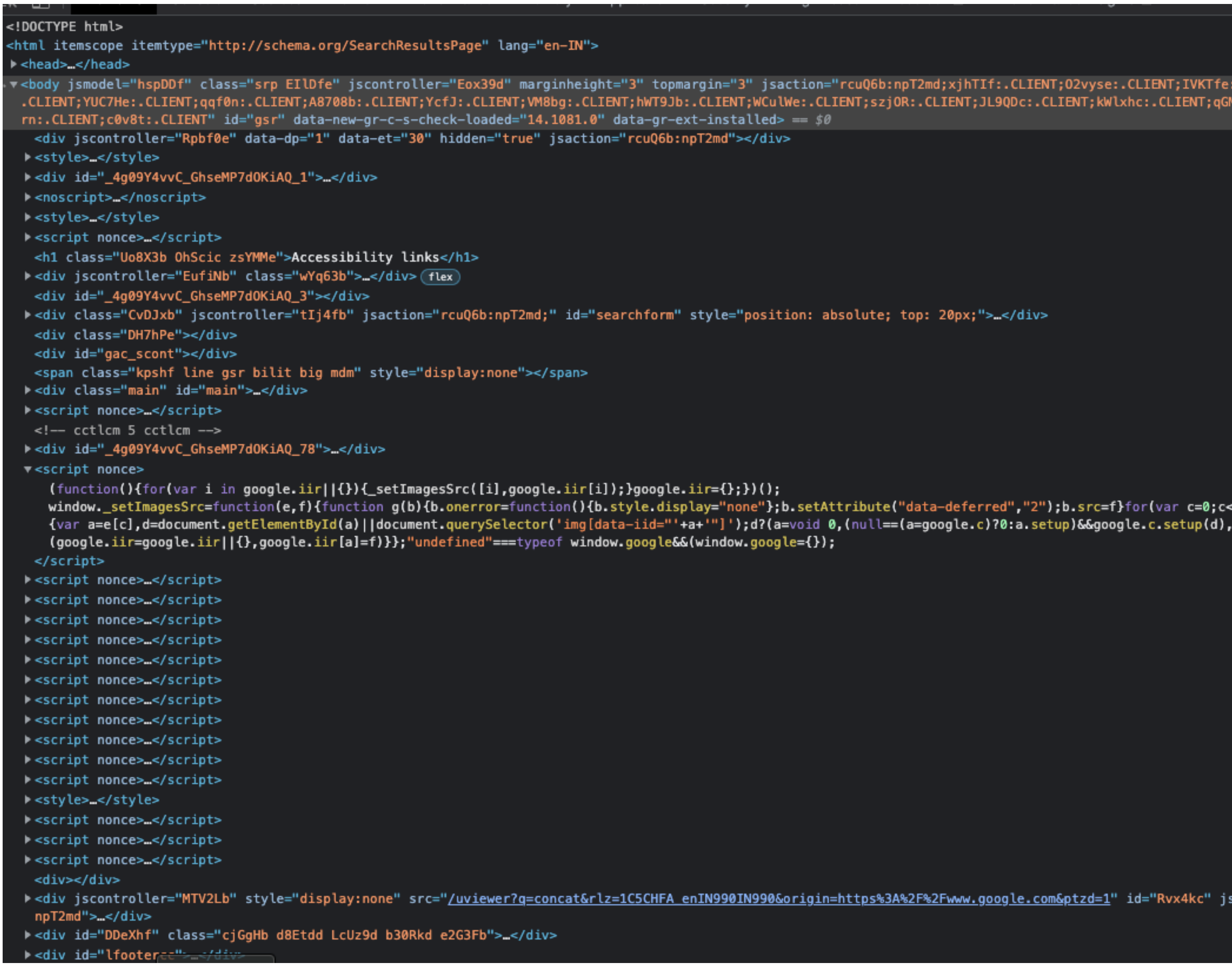
   So with the help of pattern I can conclude that

   1. Change the company column name to lower case.

2. If there is space replace with “-“ in between to concat in single company name.
3. Now **concat** company name at last with the url https://www.linkedin.com/company/ to get the company URL.

4. How to identify list of companies whose tech stack is built on Python. Give names of 5 companies if possible, by your suggested approach

Ans:- 1. For this in the chrome we have to click the 3 dot at the upper right corner and go to the more tool/developer tool. In this we have read the script. From the script we can conclude whether company using the python or not. In the below screenshot I have taken from the google website and I can see Google uses the Python.



```
<!DOCTYPE html>
<html itemscope itemtype="http://schema.org/SearchResultsPage" lang="en-IN">
 ▶<head>…</head>
 ▼<body jsmodel="hspDDf" class="srp EIlDfe" jscontroller="Eox39d" marginheight="3" topmargin="3" jsaction="rcuQ6b:npT2md;xjhTIf:.CLIENT;O2vyse:.CLIENT;IVKTfe
  .CLIENT;YUC7He:.CLIENT;qqf0n:.CLIENT;A8708b:.CLIENT;YcfJ:.CLIENT;VM8bg:.CLIENT;hWT9Jb:.CLIENT;WCulWe:.CLIENT;szjOR:.CLIENT;JL9QDc:.CLIENT;kWlxhc:.CLIENT;qGM
  rn:.CLIENT;c0v8t:.CLIENT" id="gsr" data-new-gr-c-s-check-loaded="14.1081.0" data-gr-ext-installed> == $0
    <div jscontroller="Rpbf0e" data-dp="1" data-et="30" hidden="true" jsaction="rcuQ6b:npT2md"></div>
   ▶<style>…</style>
   ▶<div id="_4g09Y4vvC_GhseMP7dOKiAQ_1">…</div>
   ▶<noscript>…</noscript>
   ▶<style>…</style>
   ▶<script nonce>…</script>
    <h1 class="Uo8X3b OhScic zsYMMe">Accessibility links</h1>
   ▶<div jscontroller="EufiNb" class="wYq63b">…</div> flex
    <div id="_4g09Y4vvC_GhseMP7dOKiAQ_3"></div>
   ▶<div class="CvDJxb" jscontroller="tIj4fb" jsaction="rcuQ6b:npT2md;" id="searchform" style="position: absolute; top: 20px;">…</div>
    <div class="DH7hPe"></div>
    <div id="gac_scont"></div>
    <span class="kpshf line gsr bilit big mdm" style="display:none"></span>
   ▶<div class="main" id="main">…</div>
   ▶<script nonce>…</script>
    <!-- cctlcm 5 cctlcm -->
   ▶<div id="_4g09Y4vvC_GhseMP7dOKiAQ_78">…</div>
   ▼<script nonce>
      (function(){for(var i in google.iir||{})(_setImagesSrc([i],google.iir[i]);}google.iir={};})();
      window._setImagesSrc=function(e,f){function g(b){b.onerror=function(){b.style.display="none"};b.setAttribute("data-deferred","2");b.src=f}for(var c=0;c<
      {var a=e[c],d=document.getElementById(a)||document.querySelector('img[data-iid="'+a+'"]');d?(a=void 0,(null==(a=google.c)?0:a.setup)&&google.c.setup(d),
      (google.iir=google.iir||{},google.iir[a]=f)}};"undefined"===typeof window.google&&(window.google={});
    </script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<style>…</style>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
   ▶<script nonce>…</script>
    <div></div>
   ▶<div jscontroller="MTV2Lb" style="display:none" src="/uviewer?q=concat&rlz=1C5CHFA_enIN990IN990&origin=https%3A%2F%2Fwww.google.com&ptzd=1" id="Rvx4kc" js
    npT2md">…</div>
   ▶<div id="DDeXhf" class="cjGgHb d8Etdd LcUz9d b30Rkd e2G3Fb">…</div>
   ▶<div id="lfooter">…</div>
```

Like google I have also checked other company using the Python script – Facebook, Quora, Netflix, Dropbox and Instagram.

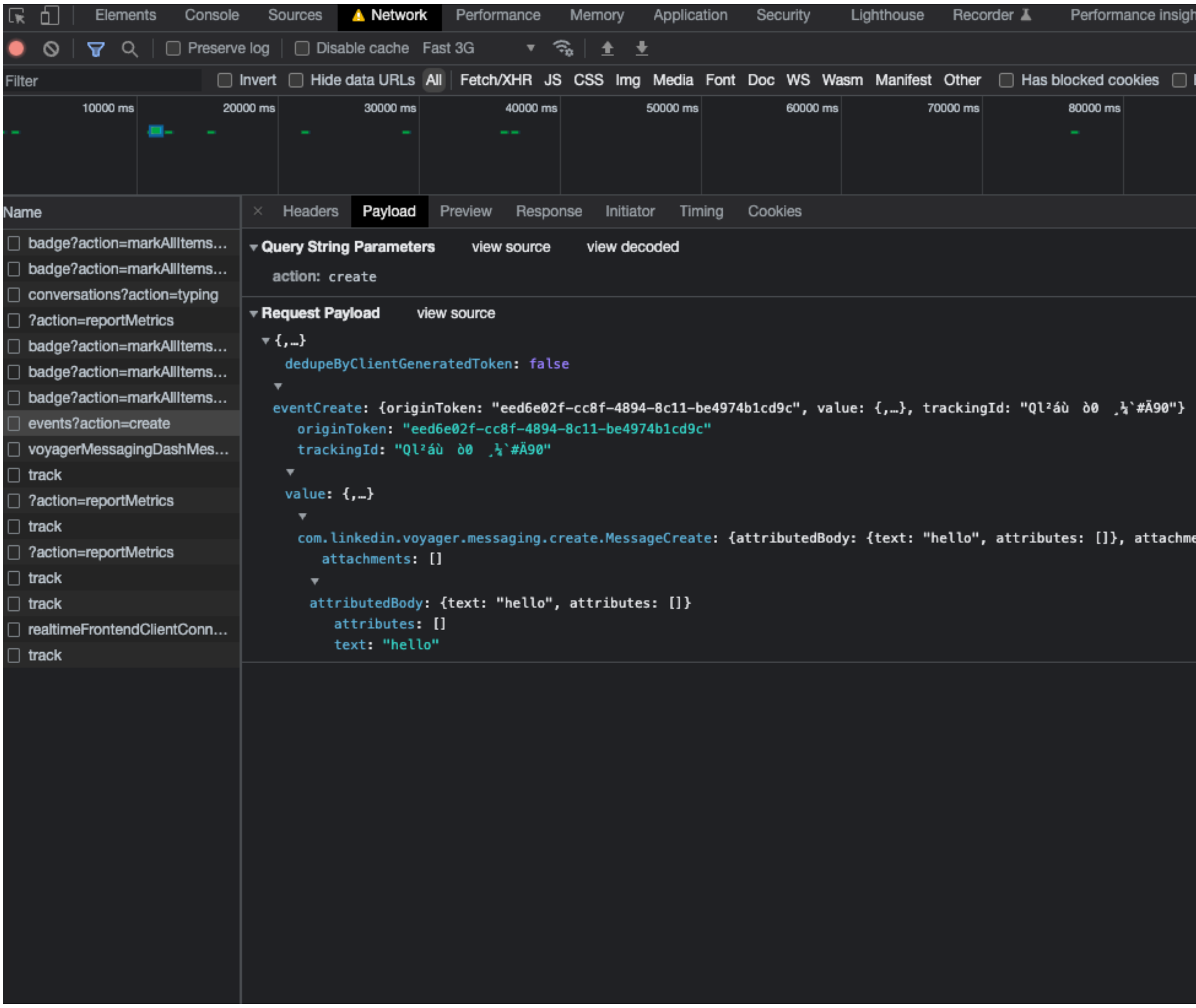
5. Need to find an API, through which we can send linkedin messages to other linkedin users.

Ans:- Dear Sir for this I have tried following

1. I have sent the message to the LinkedIn user.
2. Then I right click and then went on inspect.
3. From there to networking.

4. Cleared all the previous history.
5. Now sent the message again.
6. We see that event generated with the api
7. Also I can see the payload of the page and its showing that even of message.

   Please refer the attached screenshots.



Name

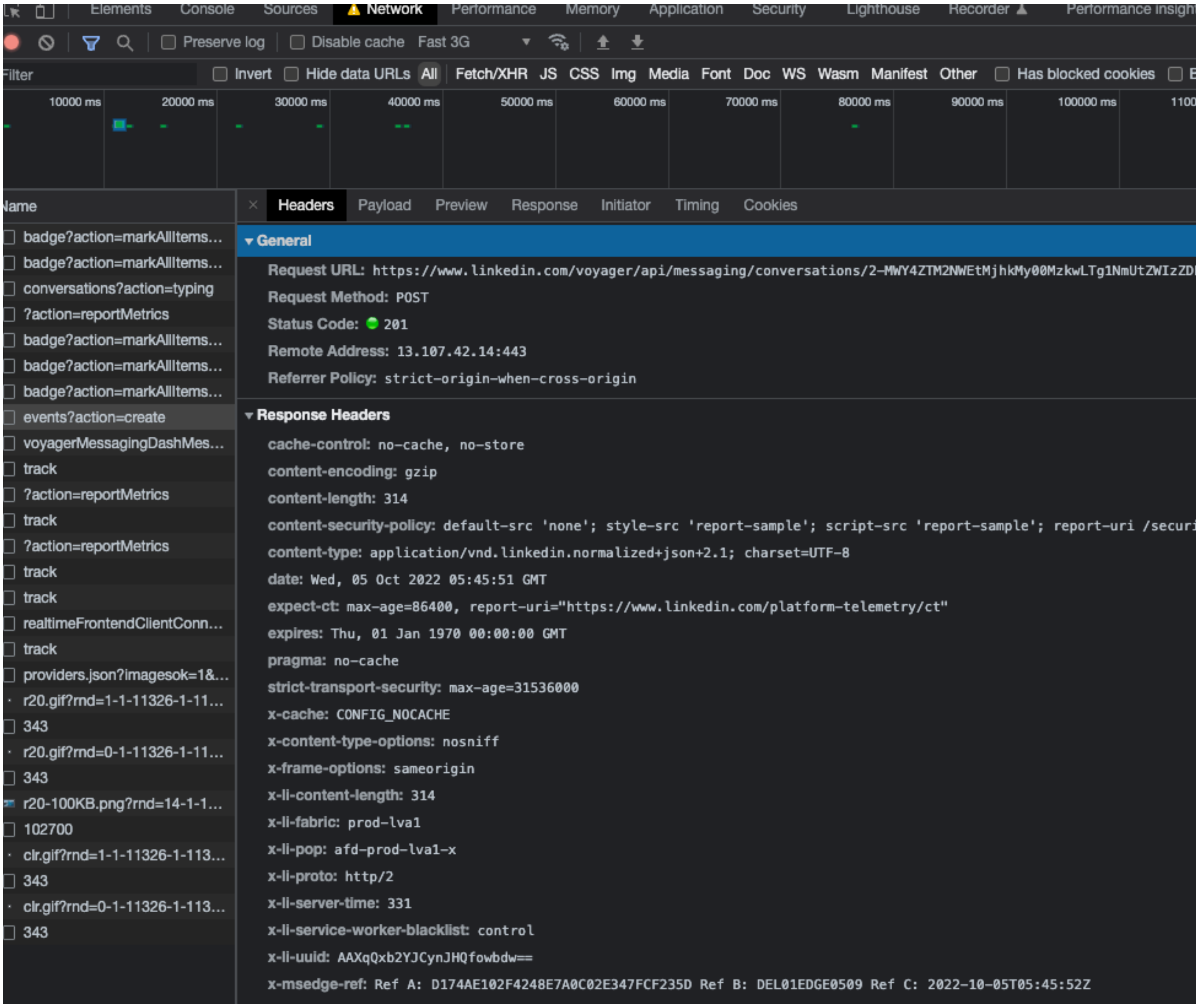
- badge?action=markAllItems...
- badge?action=markAllItems...
- conversations?action=typing
- ?action=reportMetrics
- badge?action=markAllItems...
- badge?action=markAllItems...
- badge?action=markAllItems...
- events?action=create
- voyagerMessagingDashMes...
- track
- ?action=reportMetrics
- track
- ?action=reportMetrics
- track
- track
- realtimeFrontendClientConn...
- track

Query String Parameters

```
action: create
```

Request Payload

```
▼{,…}
  dedupeByClientGeneratedToken: false
 ▼
 eventCreate: {originToken: "eed6e02f-cc8f-4894-8c11-be4974b1cd9c", value: {,…}, trackingId: "Ql²áù  ò0  ¸¼`#Ä90"}
   originToken: "eed6e02f-cc8f-4894-8c11-be4974b1cd9c"
   trackingId: "Ql²áù  ò0  ¸¼`#Ä90"
  ▼
  value: {,…}
   ▼
   com.linkedin.voyager.messaging.create.MessageCreate: {attributedBody: {text: "hello", attributes: []}, attachme
     attachments: []
    ▼
    attributedBody: {text: "hello", attributes: []}
      attributes: []
      text: "hello"
```



```
badge?action=markAllItems...
badge?action=markAllItems...
conversations?action=typing
?action=reportMetrics
badge?action=markAllItems...
badge?action=markAllItems...
badge?action=markAllItems...
events?action=create
voyagerMessagingDashMes...
track
?action=reportMetrics
track
?action=reportMetrics
track
track
realtimeFrontendClientConn...
track
providers.json?imagesok=1&...
r20.gif?rnd=1-1-11326-1-11...
343
r20.gif?rnd=0-1-11326-1-11...
343
r20-100KB.png?rnd=14-1-1...
102700
clr.gif?rnd=1-1-11326-1-113...
343
clr.gif?rnd=0-1-11326-1-113...
343
```

▼ General

```
Request URL: https://www.linkedin.com/voyager/api/messaging/conversations/2-MWY4ZTM2NWEtMjhkMy00MzkwLTg1NmUtZWIzZD
Request Method: POST
Status Code: ● 201
Remote Address: 13.107.42.14:443
Referrer Policy: strict-origin-when-cross-origin
```

▼ Response Headers

```
cache-control: no-cache, no-store
content-encoding: gzip
content-length: 314
content-security-policy: default-src 'none'; style-src 'report-sample'; script-src 'report-sample'; report-uri /secur
content-type: application/vnd.linkedin.normalized+json+2.1; charset=UTF-8
date: Wed, 05 Oct 2022 05:45:51 GMT
expect-ct: max-age=86400, report-uri="https://www.linkedin.com/platform-telemetry/ct"
expires: Thu, 01 Jan 1970 00:00:00 GMT
pragma: no-cache
strict-transport-security: max-age=31536000
x-cache: CONFIG_NOCACHE
x-content-type-options: nosniff
x-frame-options: sameorigin
x-li-content-length: 314
x-li-fabric: prod-lva1
x-li-pop: afd-prod-lva1-x
x-li-proto: http/2
x-li-server-time: 331
x-li-service-worker-blacklist: control
x-li-uuid: AAXqQxb2YJCynJHQfowbdw==
x-msedge-ref: Ref A: D174AE102F4248E7A0C02E347FCF235D Ref B: DEL01EDGE0509 Ref C: 2022-10-05T05:45:52Z
```