*Task 1: Generate Public Certificate*

The certificate authority can generate a public certificate for you.

Credentials:

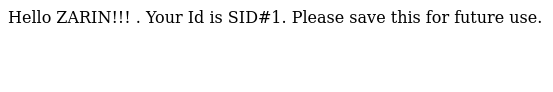
username: lazrig

password: lazrig468

1. Testing Connection to the certificate authority server:
   * [http://129.82.44.147:10200/](http://localhost:10200/)

Call this api from your browser. This api will return a welcome message from the certificate authority(CA) server.

1. Activate your account:
   * [http://129.82.44.147:10200/active-account?username=your\_username&password=](http://localhost:10200/active-account?username=your_username&password=your_password)your\_password
   * For each student there is an account created. The account is in inactive status currently. You need to activate your account by calling this api. To call this api use your eid as username and your csu id as password. In response you will receive your student Id which you will have to use to call further apis. The response will look this this.



In case you forget your student id, you can call this same api again. For future use you have to use the Id after the sub string ‘**SID#’.** In this case My Id is 1**.**

1. Sign CSR:
   * **Run sign-csr.py file**

Download the file from this link: <https://drive.google.com/drive/folders/1jrfoBTpHLLMCwz-wTDpjWwSDtBXqZiB0?usp=drive_link>

After generating your own private key, you need to request the CA server to generate your public certificate. For this you have to create your own certificate signing request(CSR) and send the CSR to the CA server.

You will call the server api by using the above python script. This will download the public certificate in your machine. Remember to put your correct student id in the api call. You can modify the python file if you want/need.

**Deliverables of task 1**: ( should be submitted using the designated dropbox on Canvas)

1- The CSR you created and sent to the CA server

2- The signed Public Key certificate you received from our CA server.

*Certificate Status Verification*

Repository Server hosts a collection of randomly generated certificates. You need to connect to this server to get a list of certificates and verify whether these are expired and/or revoked.

1. Testing Connection to the repository server:
   * http://129.82.44.147:10300/

Call this api from your browser. This api will return a welcome message from the repository server.

1. Get your own certificate Id list to check status from the repository server:
   * http://129.82.44.147:10300/get-cert-list?sid=your\_student\_id

Run this api from your browser. This api will return a list of certificate id for you to verify. As an api parameter you have to put your student id. (No worries if you forgot your student id. You can get it by calling the account activation api again.)

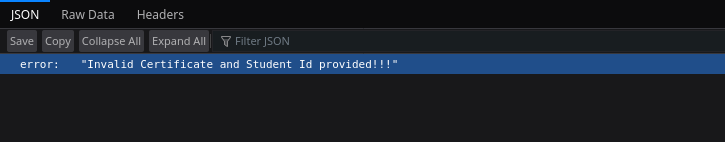
In the response you will receive 10 certificate Ids. You have to further verify these Id’s. If you forget the list, you can just call this same api again. This will return the same set of certificates for you. The response will look like below:



1. Get Certificate Content:
   * http://129.82.44.147:10300//get-certificate?sid=your\_student\_id&cert\_id=CERT6

To verify whether a certificate is expired or not you need to get the contents of the certificate. This api will download the certificate in your machine. From there you can further analyze the certificate.

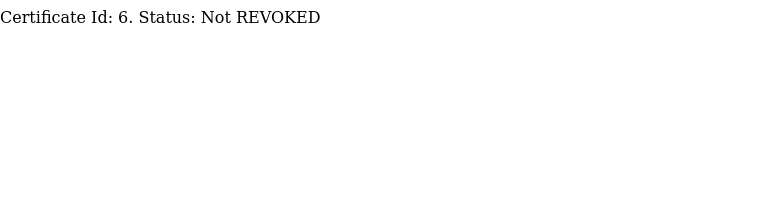
Remember to put the correct student id and certificate id from the previous api, otherwise it will result in error.



1. Verify Revoked Status:
   * [http://](http://localhost:10300/verify-revoked-status?sid=1&cert_id=CERT6)129.82.44.147[:10300/verify-revoked-status?sid=your\_student\_id&cert\_id=CERT6](http://localhost:10300/verify-revoked-status?sid=1&cert_id=CERT6)

A certificate which is not expired can still be revoked. This verification is usually done by calling the certificate authority(CA) server. So you have to call the CA server to get the revocation status of a certificate.

Run this api from your browser. Remember to put the correct student id and certificate id from the previous api, unless it will result in error. In response this will return the revocation status of the certificate. You will get a response like below:



1. Submit Expired Status Verification response:
   * **Run submit\_expired\_answer.py file**

Download the file from this link: <https://drive.google.com/drive/folders/1jrfoBTpHLLMCwz-wTDpjWwSDtBXqZiB0?usp=drive_link>

Copy this code into a python file. Run the file to submit your answer for verifying expired status of the provided certificate list. Your response should be provided as a dictionary with certificate Id being the Key. As value you should put ‘N’ if the certificate has not expired, otherwise ‘Y’.

Remember to put the correct student id. This code will return and print how many of the given certificates you’ve correctly identified their expiration status.

1. Submit Revoked Status Verification response:
   * **Run submit\_revoked\_answer.py file**

Download the file from this link: https://drive.google.com/drive/folders/1jrfoBTpHLLMCwz-wTDpjWwSDtBXqZiB0?usp=drive\_link

Copy this code into a python file. Run the file to submit your answer for verifying expired status of the provided certificate list. Your response should be provided as a dictionary with certificate Id being the Key. As value you should put ‘N’ if the certificate has not expired, otherwise ‘Y’.

Remember to put the correct student id. This code will return and print how many of the given certificates you’ve correctly identified their revocation status.