Algorithm for file updates in Python

Project description

I am a healthcare company security professional responsible for managing access to restricted content based on employee IP addresses. I am tasked with creating a Python algorithm to update the allow list file by removing any IP addresses listed in the remove list.

Open the file that contains the allow list

The function update_file opens the specified file (import_file) in read mode ("r") using the open() function and assigns it to the variable file.

Read the file contents

Within the update_file function, the contents of the opened file are read using the read() method and stored in the variable ip_addresses.

Convert the string into a list

The string ip_addresses is converted into a list of individual IP addresses using the split() method, splitting it at each whitespace character.

Iterate through the remove list

Each element (IP address) in ip_addresses within the loop is checked against the remove_list. If it exists in the remove_list, it is removed from ip_addresses.

Remove IP addresses that are on the remove list

Each element (IP address) in ip_addresses within the loop is checked against the remove_list. If it exists in the remove_list, it is removed from ip_addresses.

Update the file with the revised list of IP addresses

After removing the ip_addresses list, it is converted back into a string with IP addresses separated by whitespace using the join() method. Then, the file is opened again in write mode ("w") to overwrite its contents with the updated ip_addresses string using the write() method.

Summary

This Python algorithm effectively updates the allow list file by removing any IP addresses listed in the remove list. It opens the file, reads its contents, converts the string into a list of IP addresses, iterates through the remove list to identify and remove matching IP addresses from the allow list, and finally updates the file with the revised list of IP addresses.

```
def update_file(import_file, remove_list):
    # Build `with` statement to read in the initial contents of the file
    with open(import_file, "r") as file:
        # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`
        ip_addresses = file.read()
    # Use `.split()` to convert `ip_addresses` from a string to a list
    ip_addresses = ip_addresses.split()
    # Build iterative statement
    # Name loop variable `element`
    # Loop through `ip_addresses`
    for element in ip_addresses:
        # Build conditional statement
        # If current element is in `remove_list`,
        if element in remove_list:
            # then current element should be removed from `ip_addresses`
            ip_addresses.remove(element)
    # Convert `ip_addresses` back to a string so that it can be written into the text file
    ip_addresses = " ".join(ip_addresses)
    # Build `with` statement to rewrite the original file
    with open(import_file, "w") as file:
        # Rewrite the file, replacing its contents with `ip_addresses`
        file.write(ip_addresses)

# Call `update_file()` and pass in "allow_list.txt" and a list of IP addresses to be removed
update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])

# Build `with` statement to read in the updated file
with open("allow_list.txt", "r") as file:
    # Read in the updated file and store the contents in `text`
    text = file.read()

# Display the contents of `text`
```

```
print(text)
```