Saé15:Traitement de donné

CLAIRE Klayce

QOUDAD Bilal

Tout d’abord nous avons convertis le ics en csv :

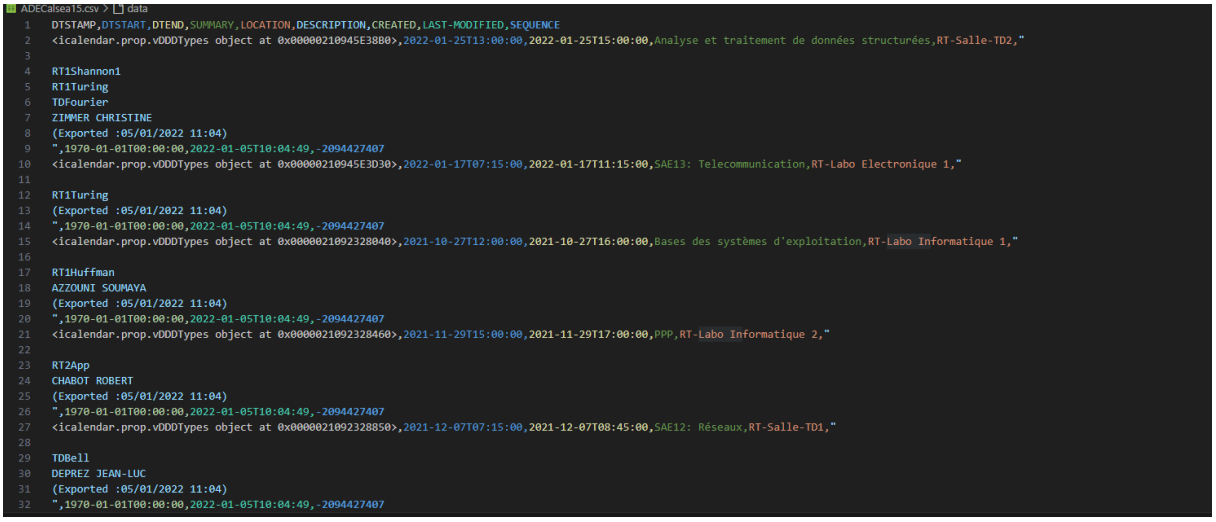
1. **Lecture du fichier ICS :** Le script commence par ouvrir le fichier ICS spécifié (ADECal (1).ics) en mode lecture binaire. Il utilise la bibliothèque icalendar pour lire le contenu du calendrier.
2. **Écriture dans le fichier CSV :** Ensuite, le script ouvre un fichier CSV spécifié (ADECal-sea15.csv) en mode écriture. Il définit les noms de champ pour les en-têtes du fichier CSV dans la liste fieldnames.
3. **Extraction des informations de l'événement :** Le script parcourt les composants du calendrier de type VEVENT (événements) et extrait certaines informations telles que la date de début (DTSTART), la date de fin (DTEND), le résumé (SUMMARY), l'emplacement (LOCATION), la description (DESCRIPTION), la date de création (CREATED), la date de dernière modification (LAST-MODIFIED), etc.
4. **Formatage des champs de date :** Pour les champs de date spécifiques (comme DTSTART, DTEND, CREATED, LAST-MODIFIED), le script formate les valeurs au format 'YYYY-MM-DDTHH:mm:ss'.
5. **Écriture dans le fichier CSV :** Les données extraites sont ensuite écrites dans le fichier CSV en utilisant la classe DictWriter de la bibliothèque CSV.
6. **Affichage du succès :** Enfin, le script affiche un message indiquant que la conversion du fichier ICS vers le fichier CSV est terminée avec succès.

Le code présente une utilisation de la bibliothèque icalendar pour manipuler des fichiers ICS et utilise la bibliothèque CSV pour la manipulation des fichiers CSV. Le résultat final est un fichier CSV contenant les informations extraites des événements du calendrier ICS.

Voici le code :

```
import icalendar
import csv

def ics_to_csv(ics_file, csv_file):
    # Ouvrir le fichier ICS
    with open(ics_file, 'rb') as ics_file:
        cal = icalendar.Calendar.from_ical(ics_file.read())

    # Ouvrir le fichier CSV pour l'écriture
    with open(csv_file, 'w', newline='', encoding='utf-8') as csv_file:
        fieldnames = ["DTSTAMP", "DTSTART", "DTEND", "SUMMARY", "LOCATION",
"DESCRIPTION", "CREATED", "LAST-MODIFIED", "SEQUENCE"]
        csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)

        # Écrire l'en-tête CSV
        csv_writer.writeheader()

        # Parcourir les composants du calendrier
        for event in cal.walk('VEVENT'):
            event_data = {}

            for field in fieldnames:
                if field == "DTSTART" or field == "DTEND" or field ==
"CREATED" or field == "LAST-MODIFIED":
                    # Formater les champs de date
                    event_data[field] = event.get(field).dt.strftime('%Y-%m-
%dT%H:%M:%S')
                else:
                    event_data[field] = str(event.get(field, ''))

            # Écrire chaque ligne dans le fichier CSV
            csv_writer.writerow(event_data)

    print(f"La conversion de {ics_file} vers {csv_file} est terminée avec
succès.")

# Exemple d'utilisation avec le nom de fichier spécifique
ics_to_csv('ADECal (1).ics', 'ADECal-sea15.csv')
```

Voici le résultat avec les données trier :



```
ADECalsea15.csv > data
1   DTSTAMP,DTSTART,DTEND,SUMMARY,LOCATION,DESCRIPTION,CREATED,LAST-MODIFIED,SEQUENCE
2   <icalendar.prop.vDDDTypes object at 0x00000210945E38B0>,2022-01-25T13:00:00,2022-01-25T15:00:00,Analyse et traitement de données structurées,RT-Salle-TD2,"
3
4   RT1Shannon1
5   RT1Turing
6   TDFourier
7   ZIMMER CHRISTINE
8   (Exported :05/01/2022 11:04)
9   ",1970-01-01T00:00:00,2022-01-05T10:04:49,-2094427407
10  <icalendar.prop.vDDDTypes object at 0x00000210945E3D30>,2022-01-17T07:15:00,2022-01-17T11:15:00,SAE13: Telecommunication,RT-Labo Electronique 1,"
11
12  RT1Turing
13  (Exported :05/01/2022 11:04)
14  ",1970-01-01T00:00:00,2022-01-05T10:04:49,-2094427407
15  <icalendar.prop.vDDDTypes object at 0x0000021092328040>,2021-10-27T12:00:00,2021-10-27T16:00:00,Bases des systèmes d'exploitation,RT-Labo Informatique 1,"
16
17  RT1Huffman
18  AZZOUNI SOUMAYA
19  (Exported :05/01/2022 11:04)
20  ",1970-01-01T00:00:00,2022-01-05T10:04:49,-2094427407
21  <icalendar.prop.vDDDTypes object at 0x0000021092328460>,2021-11-29T15:00:00,2021-11-29T17:00:00,PPP,RT-Labo Informatique 2,"
22
23  RT2App
24  CHABOT ROBERT
25  (Exported :05/01/2022 11:04)
26  ",1970-01-01T00:00:00,2022-01-05T10:04:49,-2094427407
27  <icalendar.prop.vDDDTypes object at 0x0000021092328850>,2021-12-07T07:15:00,2021-12-07T08:45:00,SAE12: Réseaux,RT-Salle-TD1,"
28
29  TDBell
30  DEPREZ JEAN-LUC
31  (Exported :05/01/2022 11:04)
32  ",1970-01-01T00:00:00,2022-01-05T10:04:49,-2094427407
```

Maintenant nous avons donc répondu à notre sujet qui était de savoir quand les salles informatiques sont dispos :

1. **Lecture du fichier CSV :** Le script commence par ouvrir un fichier CSV ("ADECalsea15.csv") à l'aide de la bibliothèque CSV de Python. Les données lues sont stockées dans une liste appelée "table".
2. **Utilisation de Pandas :** Ensuite, le script utilise la bibliothèque Pandas pour lire le même fichier CSV et stocker les données dans un objet de type DataFrame, qui offre des fonctionnalités avancées pour le traitement des données tabulaires.
3. **Filtrage des données :** Les données du DataFrame sont filtrées pour ne conserver que les créneaux horaires dont la date de début est postérieure à une date d'analyse spécifiée (utilisation de la fonction parse_dates pour convertir les colonnes de dates) et qui concernent les salles commençant par "RT-Labo Informatique".
4. **Création d'une liste de résultats :** Ensuite, le script crée une liste appelée "resultats" qui stocke les 10 premiers créneaux disponibles pour chaque salle dans la plage de dates spécifiée. Ces résultats sont formatés sous forme de dictionnaires Python.
5. **Conversion en DataFrame :** La liste de résultats est ensuite convertie en un nouveau DataFrame appelé "creneaux_disponibles_df" à l'aide de Pandas.
6. **Affichage des résultats :** Enfin, le script affiche les 10 prochains créneaux disponibles dans les salles "RT-Labo Informatique" en affichant le DataFrame "creneaux_disponibles_df".

Le code utilise des bibliothèques telles que CSV et Pandas pour manipuler les données de manière efficace et présente les résultats sous forme d'un DataFrame tabulaire pour une meilleure lisibilité. Les commentaires dans le code fournissent des explications supplémentaires sur chaque étape.

Voici le code :

```
import csv
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from datetime import datetime

def creneaux_disponibles(nom_fichier_csv, date_danalyse):
    data = pd.read_csv(nom_fichier_csv, parse_dates=['DTSTART', 'DTEND'])

    # Filtrer les données pour les créneaux après la date d'analyse et pour
les salles "RT-Labo Informatique"
    creneaux_apres_date = data[(data['DTSTART'] >= date_danalyse) &
data['LOCATION'].str.startswith('RT-Labo Informatique')]
    resultats = []

    for salle in creneaux_apres_date['LOCATION'].unique():
        creneaux_salle = creneaux_apres_date[creneaux_apres_date['LOCATION']
== salle].sort_values('DTSTART').head(10)

        for _, creneau in creneaux_salle.iterrows():
            resultat = {
                'Salle': salle,
                'Date de début': creneau['DTSTART'],
                'Date de fin': creneau['DTEND']
            }
            resultats.append(resultat)

    creneaux_disponibles_df = pd.DataFrame(resultats)

    # Afficher les 10 prochains créneaux disponibles
    print("Les 10 prochains créneaux disponibles dans les salles RT-Labo
Informatique :")
    print(creneaux_disponibles_df)

    # Sauvegarder les résultats dans un fichier CSV avec timestamp
    timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
    output_csv_file = f'resultats_creneaux_disponibles_{timestamp}.csv'
    creneaux_disponibles_df.to_csv(output_csv_file, index=False)

    # Créer la frise chronologique
    plot_timeline(creneaux_disponibles_df)

def plot_timeline(data):
    plt.figure(figsize=(10, 6))
    for idx, row in data.iterrows():
        plt.barh(row['Salle'], width=row['Date de fin'] - row['Date de
début'], left=row['Date de début'])
```

```
    plt.xlabel('Heures')
    plt.title('Frise Chronologique des Créneaux Disponibles')
    plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%H:%M'))
    plt.tight_layout()
    plt.show()

creneaux_disponibles('ADECalsea15.csv', pd.to_datetime('2022-01-01'))
```

Voici le résultat :

```
Les 10 prochains créneaux disponibles dans les salles RT-Labo Informatique :
                    Salle       Date de début         Date de fin
0   RT-Labo Informatique 1 2022-01-03 07:15:00 2022-01-03 11:15:00
1   RT-Labo Informatique 1 2022-01-03 13:00:00 2022-01-03 16:00:00
2   RT-Labo Informatique 1 2022-01-04 08:15:00 2022-01-04 11:15:00
3   RT-Labo Informatique 1 2022-01-04 13:00:00 2022-01-04 17:00:00
4   RT-Labo Informatique 1 2022-01-05 07:15:00 2022-01-05 11:00:00
5   RT-Labo Informatique 1 2022-01-05 13:00:00 2022-01-05 16:00:00
6   RT-Labo Informatique 1 2022-01-06 08:15:00 2022-01-06 11:15:00
7   RT-Labo Informatique 1 2022-01-07 08:15:00 2022-01-07 11:15:00
8   RT-Labo Informatique 1 2022-01-10 07:15:00 2022-01-10 11:15:00
9   RT-Labo Informatique 1 2022-01-10 13:00:00 2022-01-10 16:00:00
10  RT-Labo Informatique 2 2022-01-03 13:00:00 2022-01-03 17:00:00
11  RT-Labo Informatique 2 2022-01-04 13:00:00 2022-01-04 17:00:00
12  RT-Labo Informatique 2 2022-01-05 07:15:00 2022-01-05 11:15:00
13  RT-Labo Informatique 2 2022-01-05 13:00:00 2022-01-05 17:00:00
14  RT-Labo Informatique 2 2022-01-06 09:15:00 2022-01-06 11:15:00
15  RT-Labo Informatique 2 2022-01-06 12:30:00 2022-01-06 16:30:00
16  RT-Labo Informatique 2 2022-01-07 07:15:00 2022-01-07 09:15:00
17  RT-Labo Informatique 2 2022-01-07 09:15:00 2022-01-07 11:15:00
18  RT-Labo Informatique 2 2022-01-10 13:00:00 2022-01-10 17:00:00
19  RT-Labo Informatique 2 2022-01-11 07:15:00 2022-01-11 11:15:00
20  RT-Labo Informatique 3 2022-01-03 13:00:00 2022-01-03 16:00:00
21  RT-Labo Informatique 3 2022-01-04 07:15:00 2022-01-04 09:15:00
22  RT-Labo Informatique 3 2022-01-04 09:15:00 2022-01-04 11:15:00
23  RT-Labo Informatique 3 2022-01-04 13:00:00 2022-01-04 17:00:00
24  RT-Labo Informatique 3 2022-01-05 07:15:00 2022-01-05 11:15:00
```