

Application Résanet – Mission 3

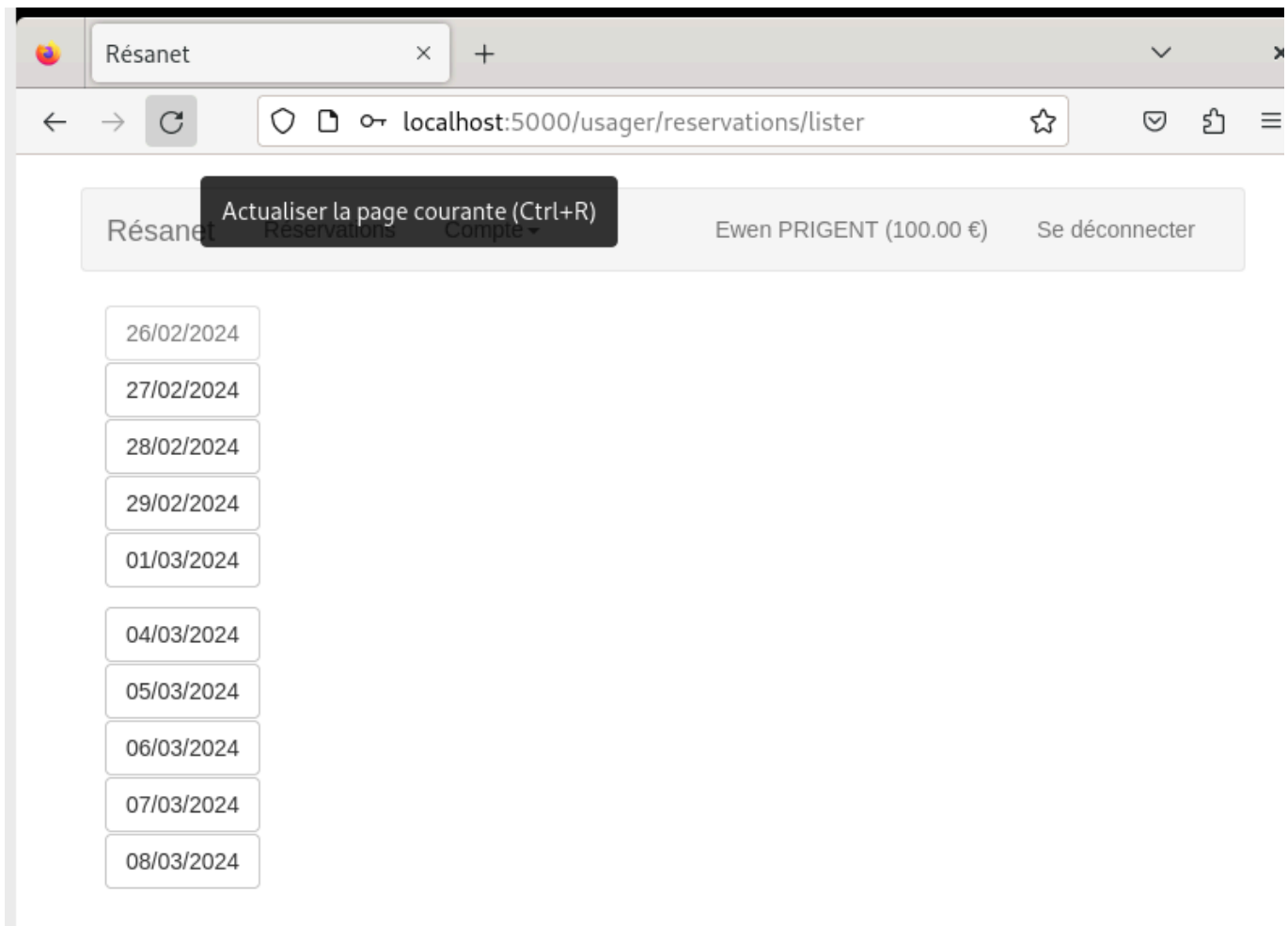
Prototype – Analyse de l'existant

"Consulter ses réservations" (UC2)

Le cas d'utilisation UC2 correspond à la consultation des réservations par l'utilisateur. Ce cas est décrit dans le cahier des charges.

1- Préparation

1. Exécuter l'application *Résanet*, puis dans un navigateur accéder à l'espace de d'un usager (authentification requise).



2. Relever l'URL de vue courante (liste des réservations de l'utilisateur choisi à la question précédente). **localhost:5000/usager/reservations/lister**

2- Sondes

Les noms des fonctions respectent un formalisme imposé dans toute documentation technique de ce projet : *nomPackage::nomFonction()*.

Exemple :

appRésanet::seDeconnecterUsager() fait référence à la fonction seDeconnecterUsager() du fichier appRésanet.py.

3. Dans le code source du *package* principal (fichier appResanet.py), relever la **route** associée à la fonction listerReservations().
‘/usager/reservations/lister’, methods = [‘GET’]
4. La fonction appResanet::listerReservations() exploite un élément d'un dictionnaire nommé session. À quoi correspond cet élément ?
Dans la fonction listerReservations() l'élément session sert à stocker les données des utilisateurs avec un identifiant de session unique.

L'objectif de cette partie est de positionner des sondes (print) dans le code source de l'application pour tracer son exécution. Deux sondes sont à positionner dans les fonctions :

Une sonde de "début d'exécution" : `print('[START] nomPackage::nomFonction()')`

Une sonde de "fin d'exécution" : `print('[END] nomPackage::nomFonction()')`

Exemple : Sondes positionnées dans la fonction appResanet::seDeconnecterUsager() :

```
def seDeconnecterUsager() :
    print( '[START] appResanet::seDeconnecterUsager()' )
    session.pop( 'numeroCarte' , None )
    session.pop( 'nom' , None )
    session.pop( 'prénom' , None )
    print( '[STOP] appResanet::seDeconnecterUsager()' )
    return redirect( '/' )
```

5. Positionner les sondes de début et de fin d'exécution dans les fonctions énumérées ci-dessous.

appResanet::listerReservations()

```
app.router.add_route( '/usager/reservations/lister' , methods = [ 'GET' ] ,
def listerReservations() :
    print( '[START] appResanet::listerReservations()' )
    tarifRepas = modeleResanet.getTarifRepas( session['numeroCarte'])

    soldeInsuffisant = False
print( '[STOP] appResanet::listerReservations()' )
return render_template( 'vueListeReservations.html' , laSessi
```

modeleResanet::getTarifRepas()

```
def getTarifRepas( numeroCarte ) :
    print( '[START] modeleResanet::getTarifRepas(numeroCarte)' )
    try :
        curseur = getConnexionBD().cursor()
        requete = '''
                                select tarifRepas
                                from Fonction
                                inner join Personnel
                                on Fonction.idFonction = Personnel.idFonction
                                inner join Carte
                                on Personnel.matricule = Carte.matricule
                                where numeroCarte = %s
                                ...
        '''
        curseur.execute( requete , ( numeroCarte , ) )
        enregistrement = curseur.fetchone()

        tarif = 'inconnu'
        if enregistrement != None :
            tarif = enregistrement[ 0 ]
            #print type(tarif)

        curseur.close()
        print( '[STOP]modeleResanet::getTarifRepas(numeroCarte)' )
        return tarif
    except :
        return None
```

modeleResanet::getSolde()

```
def getSolde( numeroCarte ) :
    print('[START] modeleResanet::getSolde(numeroCarte)')
    try :
        curseur = getConnexionBD().cursor()
        requete = '''
                        select solde
                        from Carte
                        where numeroCarte = %s
                    ...

        curseur.execute( requete , ( numeroCarte , ) )
        enregistrement = curseur.fetchone()

        solde = 'inconnu'
        if enregistrement != None :
            solde = enregistrement[ 0 ]
            #print type(solde)

        curseur.close()
        print('[STOP]modeleResanet::getSolde(numeroCarte)')
        return solde

    except :
        return None
```

modeleResanet::getReservationsCarte()

```
def getReservationsCarte( numeroCarte , dateDebut , dateFin ) :
    print('[START] modeleResanet::getReservationsCarte(numeroCarte, dateDebut, dateFin)')
    try:
        curseur = getConnexionBD().cursor()
        requete = '''
                        select dateResa
                        from Reservation
                        where numeroCarte = %s
                        and dateResa >= %s
                        and dateResa <= %s
                    ...

        curseur.execute(requete, ( numeroCarte , dateDebut , dateFin ) )
        enregistrements = curseur.fetchall()

        dates = []
        for unEnregistrement in enregistrements:
            uneDate = '%04d-%02d-%02d' % ( unEnregistrement[0].year , unEnregistrement[0].month ,
            dates.append( uneDate )

        curseur.close()
        print('[STOP] modeleResanet::getReservationsCarte(numeroCarte, dateDebut, dateFin)')
        return dates
```

dateResanet::getDateAujourd'huiISO()

```
def getDateAujourd'huiISO() :
    print('[START]datesResanet::getDateAujourd'huiISO()')
    dateCourante = datetime.datetime.today()
    aujourd'hui = '%04d-%02d-%02d' % ( dateCourante.year , dateCourante.month , dateCourante.day )
    print('[STOP]datesResanet::getDateAujourd'huiISO()')
    return aujourd'hui
```

dateResanet::getDatesPeriodeCouranteISO()

```
def getDatesPeriodeCouranteISO() :
    print('[START]datesResanet::getDatesPeriodeCourantISO()')
    dates = []

    dateAujourd'hui = datetime.datetime.today()
    numJourAujourd'hui = dateAujourd'hui.weekday()

    dateCourante = dateAujourd'hui - datetime.timedelta( numJourAujourd'hui )

    for i in range( 12 ) :
        if i != 5 and i != 6 :
            dateISO = '%04d-%02d-%02d' % ( dateCourante.year , dateCourante.month , dateCourante.day )
            dates.append( dateISO )

            dateCourante = dateCourante + datetime.timedelta( 1 )
    print('[STOP]datesResanet::getDatesPeriodeCourantISO()')
    return dates
```

dateResanet::convertirDateISOversFR()

```
def convertirDateISOversFR( dateISO ) :
    print('[START]datesResanet::convertirDateISOversFR(dateFR)')
    annee , mois , jour = dateISO.split( '-' )
    dateFR = '/'.join( ( jour , mois , annee ) )
    print('[STOP]datesResanet::convertirDateISOversFR(dateFR)')
    return dateFR
```

6. Exécuter de nouveau le cas d'utilisation UC2 puis relever l'ordre d'affichage des sondes.

```
Fichier Edition Affichage Recherche Terminal Aide
* Debugger PIN: 501-207-725
[START] appResanet::listerReservations()
[START] modeleResanet::getTarifRepas(numeroCarte)
[STOP]modeleResanet::getTarifRepas(numeroCarte)
[START] modeleResanet::getSolde(numeroCarte)
[STOP]modeleResanet::getSolde(numeroCarte)
[START] datesResanet::getDateAujourd'huiISO()
[STOP]datesResanet::getDateAujourd'huiISO()
[START] datesResanet::getDatesPeriodeCourantISO()
[STOP]datesResanet::getDatesPeriodeCourantISO()
[START] modeleResanet::getReservationsCarte(numeroCarte, dateDebut, dateFin)
[STOP] modeleResanet::getReservationsCarte(numeroCarte, dateDebut, dateFin)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] datesResanet::convertirDateISOversFR(dateFR)
[STOP]datesResanet::convertirDateISOversFR(dateFR)
[START] appResanet::listerReservations()
```

7. mettre sur github