SPRINT - 3

caloriestracker

|  |  |
| --- | --- |
| Team Id | PNT2022TMID17043 |
| Date | 16.11.2022 |
| Project Name | Ai-powered Nurition Analyzer and Fitness Enthusiasts |

from datetime import datetime

from caloriestracker.objects.meal import MealManager\_from\_sql from caloriestracker.objects.company import CompanyPersonal from caloriestracker.objects.product import ProductPersonal

from caloriestracker.objects.company\_product import CompaniesAndProducts from caloriestracker.text\_inputs import input\_decimal, input\_string

from caloriestracker.mem import MemConsole

from caloriestracker.npyscreen import MealAddApp, BiometricDataAddApp from caloriestracker.contribution import generate\_contribution\_dump,

parse\_contribution\_dump\_generate\_ﬁles\_and\_validates\_them from logging import debug

from sys import exit

def main(): mem=MemConsole() mem.run()

debug(mem.tr("Start mem took {}".format(datetime.now()-mem.inittime)))

if mem.args.ﬁnd!=None: cp=CompaniesAndProducts(mem) cp.ﬁnd\_report(mem.args.ﬁnd)

exit(0)

if mem.args.add\_company==True: name=input\_string("Name of the company: ") o=CompanyPersonal(mem)

o.name=name

o.last=datetime.now() o.obsolete=False o.save() mem.con.commit()

print("CompanySystem added with id={}".format(o.id)) exit(0)

if mem.args.elaborated!=None:

elaborated=mem.data.elaboratedproducts.ﬁnd\_by\_id(mem.args.elaborated) elaborated.needStatus(1)

elaborated.show\_table() exit(0)

if mem.args.add\_meal==True: app = MealAddApp(mem) app.run()

print(app.log) exit(0)

if mem.args.add\_product==True: name=input\_string("Add a name: ") company=mem.data.companies.ﬁnd\_by\_input()

system\_company=None if company==None else company.system\_company amount=input\_decimal("Add the product amount", 100) carbohydrate=input\_decimal("Add carbohydrate amount",0)

protein=input\_decimal("Add protein amount", 0) fat=input\_decimal("Add fat amount", 0)

ﬁber=input\_decimal("Add ﬁber amount", 0) calories=input\_decimal("Add calories amount", 0) o=ProductPersonal(mem)

o.mem=name o.amount=amount o.fat=fat o.protein=protein

o.carbohydrate=carbohydrate o.company=company o.last=datetime.now() o.calories=calories

o.ﬁber=ﬁber o.system\_company=system\_company

#Basic insert not all attributes

o.save() mem.con.commit()

print("Product added with id={}".format(o.id)) exit(0)

if mem.args.add\_biometrics==True: app = BiometricDataAddApp(mem) app.run()

exit(0)

if mem.args.contribution\_dump==True: mem.languages.cambiar("en", "caloriestracker") generate\_contribution\_dump(mem)

exit(0)

if mem.args.parse\_contribution\_dump!=None: mem.languages.cambiar("en", "caloriestracker")

dbversion=mem.con.cursor\_one\_ﬁeld("select value from globals where global='Version'")

ﬁlenameversion=mem.args.parse\_contribution\_dump.replace("caloriestracker\_collabora tion\_", "").replace(".sql", "")

if dbversion==ﬁlenameversion:

parse\_contribution\_dump\_generate\_ﬁles\_and\_validates\_them(mem.con, mem.args.parse\_contribution\_dump)

else:

print("The dump provided can't be parsed due to is from {} version and developer version is {}. Tell the user to update app and resend dump.".format(ﬁlenameversion, dbversion, ))

exit(0)

if mem.args.update\_after\_contribution!=None: mem.languages.cambiar("en", "caloriestracker")

dbversion=mem.con.cursor\_one\_ﬁeld("select value from globals where global='Version'")

ﬁlenameversion=mem.args.update\_after\_contribution[0:12] print(dbversion, ﬁlenameversion)

if dbversion==ﬁlenameversion: mem.con.load\_script(mem.args.update\_after\_contribution) mem.con.commit()

print("Personal data you sent to us, have been integrated in the Calories Tracker system data. Thank you ;)")

else:

print("Your Calories Tracker app should be in {} version to parse this script. Please update to that version because it's {}".format(dbversion, ﬁlenameversion))

exit(0)

user=mem.data.users.ﬁnd\_by\_id(mem.args.users\_id) meals=MealManager\_from\_sql(mem, "select \* from meals where datetime::date=%s

and users\_id=%s", (mem.args.date, user.id)) meals.order\_by\_datetime() meals.show\_table(mem.args.date)