**Fichier modules/cragr/module.py**

*diff --git a/modules/cragr/module.py b/modules/cragr/module.py*

*index bf41ad4..d303266 100644*

*--- a/modules/cragr/module.py*

*+++ b/modules/cragr/module.py*

@@ -18,7 +18,7 @@

# along with weboob. If not, see <http://www.gnu.org/licenses/>.

-from weboob.capabilities.bank import CapBank, AccountNotFound

+from weboob.capabilities.bank import CapBank, AccountNotFound, Transfer, TransferError, Recipient

from weboob.tools.backend import Module, BackendConfig

from weboob.tools.ordereddict import OrderedDict

from weboob.tools.value import ValueBackendPassword, Value

@@ -108,5 +108,12 @@ class CragrModule(Module, CapBank):

def iter\_history(self, account):

return self.browser.get\_history(account)

+ def iter\_transfer\_recipients(self, ignored):

+ for account in self.browser.get\_transfer\_accounts().itervalues():

+ recipient = Recipient()

+ recipient.id = account.id

+ recipient.label = account.label

+ yield recipient

+

def transfer(self, account, to, amount, reason=None):

return self.browser.do\_transfer(account, to, amount, reason)

**Fichier modules/cragr/web/browser.py**

*diff --git a/modules/cragr/web/browser.py b/modules/cragr/web/browser.py*

*index ce6d9ed..4847989 100644*

*--- a/modules/cragr/web/browser.py*

*+++ b/modules/cragr/web/browser.py*

@@ -20,12 +20,22 @@

import urllib

import re

+import mechanize

from weboob.deprecated.browser import Browser, BrowserIncorrectPassword

from weboob.tools.date import LinearDateGuesser

+from weboob.capabilities.bank import Transfer, TransferError

+from selenium.webdriver.common.by import By

+from selenium.webdriver.support import expected\_conditions as EC

+from selenium.webdriver.support.ui import WebDriverWait

+from selenium.common.exceptions import NoSuchElementException

+from selenium.webdriver.support.ui import Select

+from shutil import rmtree

+

+from datetime import datetime

from .pages import HomePage, LoginPage, LoginErrorPage, AccountsPage, \

- SavingsPage, TransactionsPage, UselessPage, CardsPage

+ TransferPage, SavingsPage, TransactionsPage, UselessPage, CardsPage

\_\_all\_\_ = ['Cragr']

@@ -36,12 +46,14 @@ class Cragr(Browser):

ENCODING = 'ISO-8859-1'

PAGES = {'https?://[^/]+/': HomePage,

+ 'https?://[^/]+/particuliers.html': HomePage,

'https?://[^/]+/stb/entreeBam': LoginPage,

'https?://[^/]+/stb/entreeBam\?.\*typeAuthentification=CLIC\_ALLER.\*': LoginPage,

'https?://[^/]+/stb/entreeBam\?.\*pagePremVisite.\*': UselessPage,

'https?://[^/]+/stb/entreeBam\?.\*Interstitielle.\*': UselessPage,

'https?://[^/]+/stb/entreeBam\?.\*act=Tdbgestion': UselessPage,

'https?://[^/]+/stb/entreeBam\?.\*act=Synthcomptes': AccountsPage,

+ 'https?://[^/]+/stb/entreeBam\?.\*act=Virementssepa': TransferPage,

'https?://[^/]+/stb/entreeBam\?.\*act=Synthepargnes': SavingsPage,

'https?://[^/]+/stb/.\*act=Releves.\*': TransactionsPage,

'https?://[^/]+/stb/collecteNI\?.\*sessionAPP=Releves.\*': TransactionsPage,

@@ -83,6 +95,7 @@ class Cragr(Browser):

# On the homepage, we get the URL of the auth service.

url = self.page.get\_post\_url()

+ self.login\_url = url

if url is None:

raise self.WebsiteNotSupported()

@@ -135,6 +148,9 @@ class Cragr(Browser):

# we can deduce the URL to "savings" accounts from the regular accounts one

self.savings\_url = re.sub('act=([^&=]+)', 'act=Synthepargnes', self.accounts\_url, 1)

+ # we can deduce the URL to "transfer" from the regular accounts one

+ self.transfer\_url = re.sub('act=([^&=]+)', 'act=Virementssepa', self.accounts\_url, 1)

+

def get\_accounts\_list(self):

accounts\_list = []

# regular accounts

@@ -188,3 +204,172 @@ class Cragr(Browser):

yield tr

url = self.page.get\_next\_url()

+

+ def get\_transfer\_accounts(self):

+ if not self.is\_on\_page(TransferPage):

+ self.location(self.transfer\_url)

+

+ assert self.is\_on\_page(TransferPage)

+ return self.page.get\_transfer\_target\_accounts()

+

+ def selenium\_start(self):

+ # To avoid ImportError during e.g. building modules list.

+ from selenium import webdriver

+

+ WIDTH = 1920

+ HEIGHT = 10000 # So that everything fits...

+

+ prof = webdriver.FirefoxProfile()

+ self.\_browser = webdriver.Firefox(prof)

+ self.\_browser.set\_window\_size(WIDTH, HEIGHT)

+

+ url= self.login\_url

+ data = {'CCPTE': self.username.encode(self.ENCODING),

+ 'canal': 'WEB',

+ 'hauteur\_ecran': 768,

+ 'largeur\_ecran': 1024,

+ 'liberror': '',

+ 'matrice': 'true',

+ 'origine': 'vitrine',

+ 'situationTravail': 'BANCAIRE',

+ 'typeAuthentification': 'CLIC\_ALLER',

+ 'urlOrigine': self.page.url,

+ 'vitrine': 0,

+ }

+ self.\_browser.get(url+'?'+urllib.urlencode(data))

+

+ imgmap = {}

+ self.\_browser.implicitly\_wait(3)

+

+ for td in self.\_browser.find\_elements\_by\_xpath('//table[@id="pave-saisie-code"]/tbody/tr/td'):

+ a = td.find\_element\_by\_xpath('.//a')

+ num = a.get\_attribute("text").strip()

+ if num.isdigit():

+ imgmap[num] = int(a.get\_attribute("tabindex")) - 1

+

+ hidden1 = self.\_browser.find\_element\_by\_name('CCCRYC')

+ value1 = ','.join(['%02d' % imgmap[c] for c in self.password])

+ self.\_browser.execute\_script("arguments[0].value = '%s';"%value1,hidden1)

+ hidden2 = self.\_browser.find\_element\_by\_name('CCCRYC2')

+ value2 = '0' \* len(self.password)

+ self.\_browser.execute\_script("arguments[0].value = '%s';"%value2,hidden2)

+

+ self.\_browser.execute\_script("javascript:Valid()")

+ try:

+ element = WebDriverWait(self.\_browser, 10).until(

+ EC.presence\_of\_element\_located((By.ID, "btn-deconnexion"))

+ )

+ except:

+ self.selenium\_finish()

+

+ def selenium\_finish(self):

+ prof\_dir = self.\_browser.firefox\_profile.profile\_dir

+ self.\_browser.close()

+ del self.\_browser

+ rmtree(prof\_dir)

+

+ def do\_transfer(self, account, to, amount, reason=None):

+ """

+ Transfer the given amount of money from an account to another,

+ tagging the transfer with the given reason.

+ """

+

+ self.selenium\_start()

+

+ # access the transfer page

+

+ transfer\_page\_unreachable\_message = u'Could not reach the transfer page.'

+ transfer\_page\_processerror\_message = u'Error while processing the transfer.'

+ #self.\_browser.get(self.absurl('/'))

+ self.selenium\_accounts\_url = self.\_browser.current\_url

+ self.selenium\_transfer\_url = re.sub('act=([^&=]+)', 'act=Virementssepa', self.selenium\_accounts\_url, 1)

+ self.\_browser.get(self.selenium\_transfer\_url)

+

+ try:

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//\*[text() ="Compte bénéficiaire :"]'))

+ )

+ except:

+ self.selenium\_finish()

+ print(transfer\_page\_unreachable\_message)

+ raise

+

+ # separate euros from cents

+ amount\_euros = int(amount)

+ amount\_cents = int((amount \* 100) - (amount\_euros \* 100))

+

+

+ try:

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//\*[@name ="VIR\_VIR1\_FR3\_LE"]'))

+ )

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//\*[@name ="VIR\_VIR1\_FR3\_LB"]'))

+ )

+ except:

+ self.selenium\_finish()

+ print(transfer\_page\_processerror\_message)

+ raise

+

+ for option in Select(self.\_browser.find\_element\_by\_name('VIR\_VIR1\_FR3\_LE')).options:

+ if account in option.text:

+ Select(self.\_browser.find\_element\_by\_name('VIR\_VIR1\_FR3\_LE')).select\_by\_visible\_text(option.text)

+

+ for option in Select(self.\_browser.find\_element\_by\_name('VIR\_VIR1\_FR3\_LB')).options:

+ if to in option.text:

+ Select(self.\_browser.find\_element\_by\_name('VIR\_VIR1\_FR3\_LB')).select\_by\_visible\_text(option.text)

+ amount1 = self.\_browser.find\_element\_by\_name('T3SEF\_MTT\_EURO')

+ self.\_browser.execute\_script("arguments[0].value = '%s';"%amount\_euros,amount1)

+ amount2 = self.\_browser.find\_element\_by\_name('T3SEF\_MTT\_CENT')

+ self.\_browser.execute\_script("arguments[0].value = '%02d';"%amount\_cents,amount2)

+

+ #click on first step

+ self.\_browser.execute\_script("javascript:verif('Confirmer')")

+

+ try:

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//\*[@name ="VICrt\_CDDOOR"]'))

+ )

+ except:

+ self.selenium\_finish()

+ print(transfer\_page\_processerror\_message)

+ raise

+

+ reason\_elm = self.\_browser.find\_element\_by\_name('VICrt\_CDDOOR')

+

+ if not reason is None:

+ self.\_browser.execute\_script("arguments[0].value = '%s';"%reason,reason\_elm)

+

+ submit\_date = datetime.now()

+

+ self.\_browser.execute\_script("javascript:verif('Confirmer')")

+

+ try:

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//a[text() ="Confirmer"]'))

+ )

+ except:

+ self.selenium\_finish()

+ print(transfer\_page\_processerror\_message)

+ raise

+

+ self.\_browser.execute\_script("javascript:verif('Confirmer')")

+

+ try:

+ element = WebDriverWait(self.\_browser, 3).until(

+ EC.presence\_of\_element\_located((By.XPATH, '//a[text() ="Nouveau virement"]'))

+ )

+ except:

+ self.selenium\_finish()

+ print(transfer\_page\_processerror\_message)

+ raise

+

+ # We now have to return a Transfer object

+ # the final page does not provide any transfer id, so we'll use the submit date

+ transfer = Transfer(submit\_date.strftime('%Y%m%d%H%M%S'))

+ transfer.amount = amount

+ transfer.origin = account

+ transfer.recipient = to

+ transfer.date = submit\_date

+ self.selenium\_finish()

+ return transfer

**Fichier modules/cragr/web/pages.py**

*diff --git a/modules/cragr/web/pages.py b/modules/cragr/web/pages.py*

*index 3daf800..fb37144 100644*

*--- a/modules/cragr/web/pages.py*

*+++ b/modules/cragr/web/pages.py*

@@ -21,10 +21,17 @@ import re

from decimal import Decimal

from weboob.tools.date import parse\_french\_date

-from weboob.capabilities.bank import Account

+from weboob.capabilities.bank import Account, Transfer, TransferError

from weboob.deprecated.browser import Page, BrokenPageError

+from weboob.tools.ordereddict import OrderedDict

from weboob.tools.capabilities.bank.transactions import FrenchTransaction as Transaction

+from lxml import etree

+

+class Account\_perso(object):

+ def \_\_init\_\_(self, id, label):

+ self.id = id

+ self.label = label

class HomePage(Page):

def get\_post\_url(self):

@@ -66,6 +73,32 @@ class UselessPage(Page):

class LoginErrorPage(Page):

pass

+class TransferPage(Page):

+ def get\_transfer\_target\_accounts(self):

+ target\_accounts = OrderedDict()

+ for select in self.document.xpath('//select'):

+ if not select.attrib.get('id', '')=='compte-emmet' and not select.attrib.get('id', '')=='devise':

+ count=select.xpath('count(//option)')

+ j=1

+ for i in range(1,int(count)):

+ opttextraw=select.xpath('//option[%s]/text()'%i)

+ optvalueraw=select.xpath('//option[%s]/@value'%i)

+ for opttext in opttextraw:

+ s=re.search('^\s\*(.\*)\s\*(\w{2,}\d{14,})',opttext)

+ if not (s is None):

+ target\_account\_value=j

+ j=j+1

+ account = Account\_perso(target\_account\_value, s.group(2))

+ target\_accounts[target\_account\_value] = account

+ return target\_accounts

+

+ def is\_transfer\_page(self):

+ """

+ Returns True if the current page appears to be the page dedicated to

+ transfer

+ """

+ link = self.document.xpath('//label[contains(text(), "Compte b")]')

+ return bool(link)

class \_AccountsPage(Page):

COL\_LABEL = 0

@@ -234,6 +267,13 @@ class AccountsPage(\_AccountsPage):

class SavingsPage(\_AccountsPage):

COL\_ID = 1

+ def transfer\_page\_url(self):

+ """

+ Returns the link to the "Virements" page. This function assumes the

+ current page is the saving page

+ """

+ link = self.document.xpath('/html/body//a[@accesskey=1]/@href')

+ return link[0]

**Fichier weboob/applications/boobank/boobank.py**

*diff --git a/weboob/applications/boobank/boobank.py b/weboob/applications/boobank/boobank.py*

*index 10a20a1..95b61d9 100644*

*--- a/weboob/applications/boobank/boobank.py*

*+++ b/weboob/applications/boobank/boobank.py*

@@ -420,10 +420,19 @@ class Boobank(ReplApplication):

self.cached\_format(recipient)

return 0

+ self.logger.debug('from = '+id\_from)

+ for recipient in self.do('iter\_transfer\_recipients', account.id, backends=account.backend):

+ #self.logger.debug('recipientid = '+recipient.id)

+ self.logger.debug('recipientlabel = '+recipient.label)

+ if id\_to == recipient.label:

+ self.logger.debug(recipient.id)

+ id\_to\_orig = id\_to

+ id\_to = str(recipient.id)+'@'+account.backend

+

id\_to, backend\_name\_to = self.parse\_id(id\_to)

if account.backend != backend\_name\_to:

- print("Transfer between different backends is not implemented", file=self.stderr)

+ print("Transfer between different backends is not implemented dd", file=self.stderr)

return 4

try:

@@ -450,7 +459,7 @@ class Boobank(ReplApplication):

return

self.start\_format()

- for transfer in self.do('transfer', account.id, id\_to, amount, reason, backends=account.backend):

+ for transfer in self.do('transfer', id\_from, id\_to\_orig, amount, reason, backends=account.backend):

self.format(transfer)