Αυτοματοποιημένη Διαχείριση Συστημάτων

Διάλεξη 8η

Θωμάς Καμαλάκης

Χαροκόπειο Πανεπιστήμιο Αθηνών

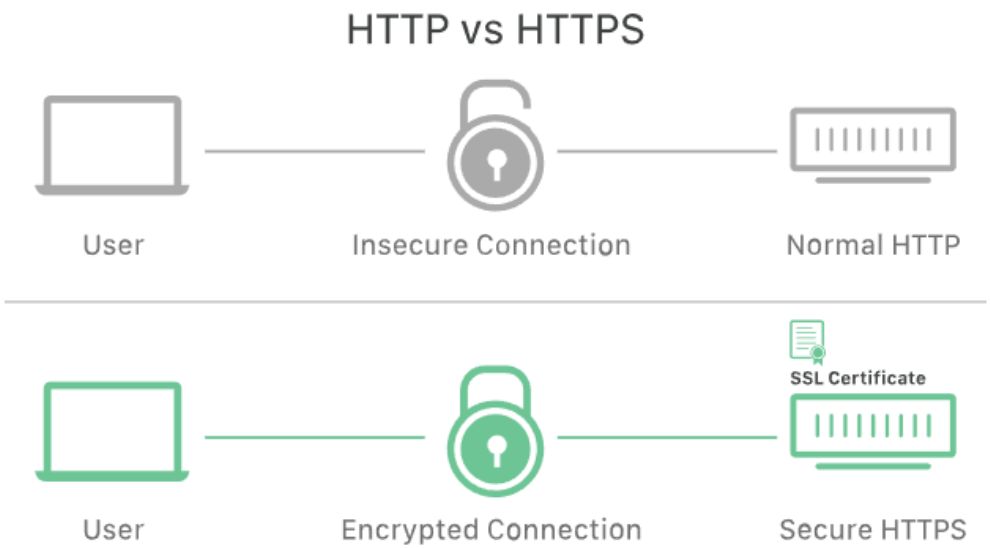
Οκτώβριος 2022

Εισαγωγή

- Για να χρησιμοποιήσουμε το API του Budibase θα χρειαστούμε την βιβλιοθήκη requests της Python.
- Είναι builtin.
- Μας βοηθάει να στείλουμε HTTP requests σε κάποιο URL.
- Εκεί συνήθως ζει ένα micro-service που μας απαντάει.
- Στο budibase, το URL είναι της μορφής http://192.168.56.102:10000/api/public/v1/

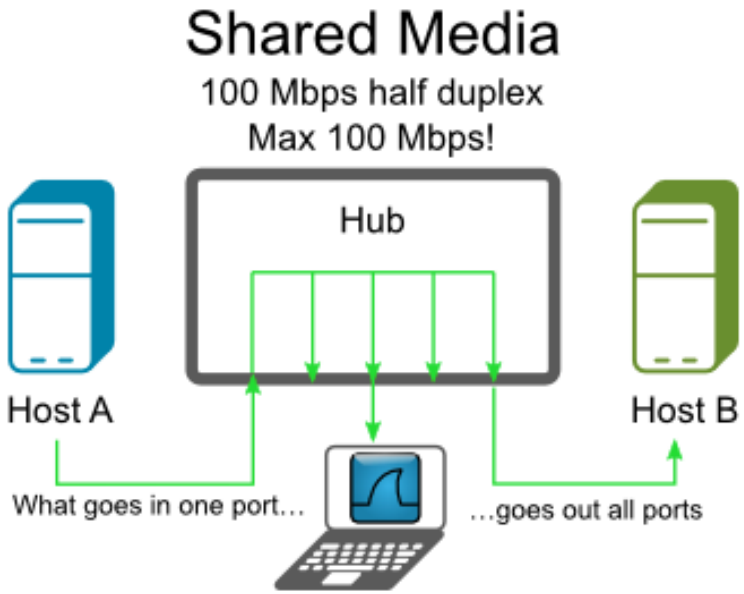
Προσοχή!

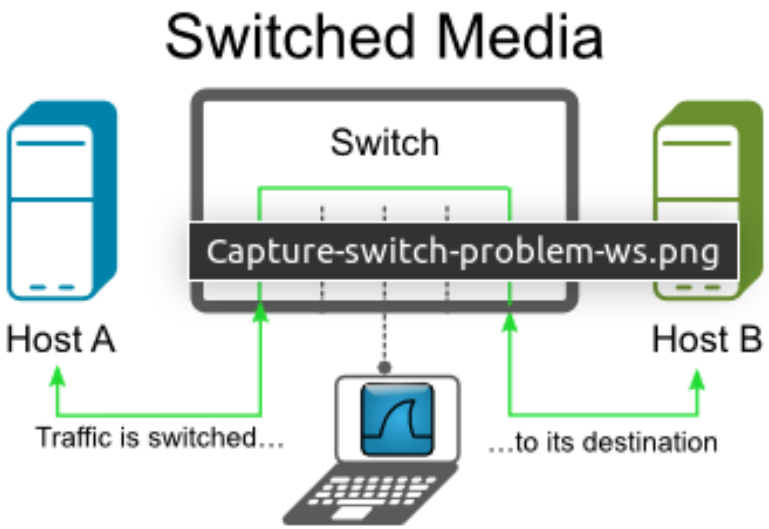
- Εμείς θα χρησιμοποιήσουμε http αλλά δεν είναι ασφαλές!
- Δεν θα δουλεύαμε έτσι σε production περιβάλλον.
- Εκεί θα έπρεπε να χρησιμοποιήσουμε https.



Προσοχή!

- Με ένα wireshark μπορεί κάποιος να ακούσει τα πακέτα μας.
- Βέβαια στα σύγχρονα ενσύρματα δίκτυα έχουμε switched αρχιτεκτονική.





Υλοποίηση

```
2  API_KEY = '9b97a3f4557422d1a8eb4afc826dc653-35
       c033e048e478935d278bdc8d15773e17c4ef3b57c38c640746c9e0a0054578f0718d1006281b55'
3  URL = 'http://192.168.56.102:10000/api/public/v1'
4  APP_MAP = {
5      'HRMS' : 'app_7a2fcd762e6a47d19ed0a5557296417d'
6  }
7  TABLE_IDS = {
8      'personnel' : 'ta_a4fd098cc1dd4a4cac9a889d4ccef86d'
9  }
10
11 import requests
12 import json
13 class budiapi:
14
15     def __init__(self, api_key = API_KEY, master_url = URL, app_map = APP_MAP, table_ids = TABLE_IDS):
16         self.api_key = api_key
17         self.master_url = master_url
18         self.app_map = app_map
19         self.table_ids = table_ids
```

Listing: budi.py

Υλοποίηση

```
22      def post_headers(self, app_id = None):
23          headers = {"Accept": "application/json",
24                     "Content-Type": "application/json",
25                     "x-budibase-api-key" : self.api_key }
26          if app_id is not None:
27              headers['x-budibase-app-id'] = app_id
28
29          return headers
```

Listing: budi.py

Υλοποίηση

```
32      def get_table_info(self, table_name, app_name):
33          table_id = self.table_ids[ table_name ]
34          app_id = self.app_map[ app_name ]
35          url = self.master_url + '/tables/' + table_id
36          headers = self.post_headers( app_id = app_id )
37          response = requests.get(url, headers = headers)
38          return json.loads(response.text)[ 'data' ]
```

Listing: budi.py

Υλοποίηση

```
41      def get_table_data(self, table_name, app_name):
42          table_id = self.table_ids[ table_name ]
43          app_id = self.app_map[ app_name ]
44          url = self.master_url + '/tables/' + table_id + '/rows/search'
45          headers = self.post_headers( app_id = app_id )
46          response = requests.post(url, headers = headers)
47          return json.loads(response.text)[ 'data' ]
```

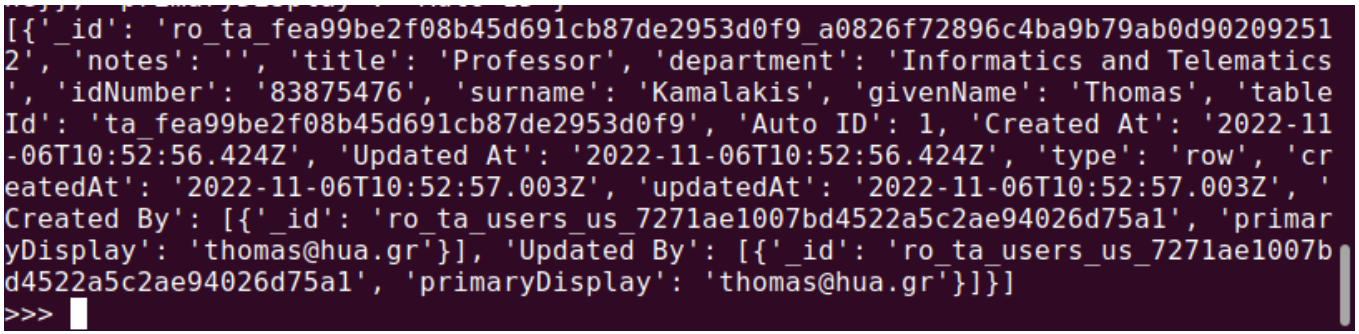
Listing: budi.py

Απλή δοκιμή

```python
from budi import budiapi
b = budiapi()
info = b.get_table_info('personnel', 'HRMS')
print(info)

data = b.get_table_data('personnel', 'HRMS')
print(data)
```

Listing: table.py



Δημιουργία

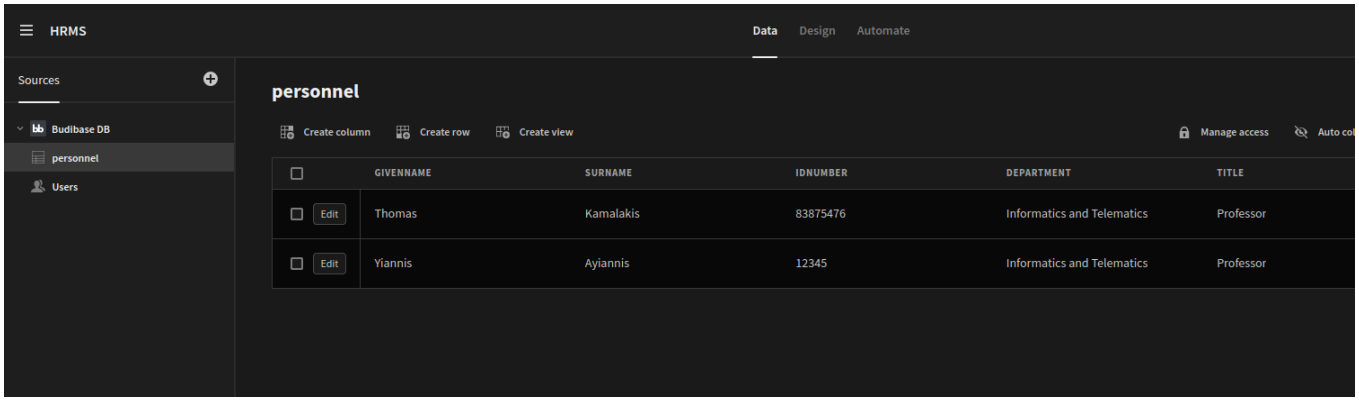
```
50      def create_row(self, table_name, app_name, value_dict):
51          table_id = self.table_ids[ table_name ]
52          app_id = self.app_map[ app_name ]
53          url = self.master_url + '/tables/' + table_id + '/rows'
54          headers = self.post_headers( app_id = app_id )
55          response = requests.post(url, headers = headers, json = value_dict)
56          return json.loads(response.text)[ 'data' ]
```

Listing: budi.py

```
1  from budi import budiapi
2  b = budiapi()
3  r = b.create_row('personnel', 'HRMS',
4                   {'notes': 'nada',
5                    'title': 'Professor',
6                    'department': 'Informatics and Telematics',
7                    'idNumber': '12345',
8                    'surname': 'Ayiannis',
9                    'givenName': 'Yiannis' })
10 print(r)
```

Listing: createrow.py

Δημιουργία




```
thomas@kirklaptop:~/Documents/mscpython/code/lecture8$ python3 -i createrow.py
{'notes': 'nada', 'title': 'Professor', 'department': 'Informatics and Telematic
s', 'idNumber': '12345', 'surname': 'Ayiannis', 'givenName': 'Yiannis', 'tableId
': 'ta_fea99be2f08b45d691cb87de2953d0f9', 'type': 'row', '_id': 'ro_ta_fea99be2f
08b45d691cb87de2953d0f9_2e835f28918a405b8774bd8d74178c10', 'Auto ID': 2, 'Create
d At': '2022-11-13T12:38:31.509Z', 'Updated At': '2022-11-13T12:38:31.509Z', 'Cr
eated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', '_rev': '
2-be9845daefd0459fe36206e3dfd40ac4', 'createdAt': 1667730273312, 'email': 'thoma
s@hua.gr', 'builder': {'global': True}, 'admin': {'global': True}, 'tenantId': '
default', 'status': 'active', 'updatedAt': '2022-11-13T12:18:19.760Z', 'dayPassR
ecordedAt': '2022-11-13T12:18:19.760Z', 'roleId': 'ADMIN', 'tableId': 'ta_users'
}], 'Updated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', '_
rev': '2-be9845daefd0459fe36206e3dfd40ac4', 'createdAt': 1667730273312, 'email':
 'thomas@hua.gr', 'builder': {'global': True}, 'admin': {'global': True}, 'tenan
tId': 'default', 'status': 'active', 'updatedAt': '2022-11-13T12:18:19.760Z', 'd
ayPassRecordedAt': '2022-11-13T12:18:19.760Z', 'roleId': 'ADMIN', 'tableId': 'ta
_users'}]}
>>>
```

Αναζήτηση

```
59      def search_rows(self, table_name, app_name, query_dict):
60          table_id = self.table_ids[ table_name ]
61          app_id = self.app_map[ app_name ]
62          url = self.master_url + '/tables/' + table_id + '/rows/search'
63          headers = self.post_headers( app_id = app_id )
64          payload = {'query' : {'string' : query_dict} }
65          response = requests.post(url, headers = headers, json = payload )
66          return json.loads(response.text)[ 'data' ]
```

Listing: budi.py

```
1  from budi import budiapi
2  b = budiapi()
3  r = b.search_rows('personnel', 'HRMS',
4                    {'title': 'Professor' } )
5  print(r)
6
7  r = b.search_rows('personnel', 'HRMS',
8                    {'surname': 'Kamalakis' } )
9  print(r)
```

Listing: searchrow.py

Αναζήτηση


[{'_id': 'ro_ta_fea99be2f08b45d691cb87de2953d0f9_a0826f72896c4ba9b79ab0d902092512', 'notes': '', 'title': 'Profes
sor', 'department': 'Informatics and Telematics', 'idNumber': '83875476', 'surname': 'Kamalakis', 'givenName': 'T
homas', 'tableId': 'ta_fea99be2f08b45d691cb87de2953d0f9', 'Auto ID': 1, 'Created At': '2022-11-06T10:52:56.424Z',
'Updated At': '2022-11-06T10:52:56.424Z', 'type': 'row', 'createdAt': '2022-11-06T10:52:57.003Z', 'updatedAt': '
2022-11-06T10:52:57.003Z', 'Created By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', 'primaryDisp
lay': 'thomas@hua.gr'}], 'Updated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', 'primaryDispla
y': 'thomas@hua.gr'}]}, {'_id': 'ro_ta_fea99be2f08b45d691cb87de2953d0f9_2e835f28918a405b8774bd8d74178c10', 'notes
': 'nada', 'title': 'Professor', 'department': 'Informatics and Telematics', 'idNumber': '12345', 'surname': 'Ayi
annis', 'givenName': 'Yiannis', 'tableId': 'ta_fea99be2f08b45d691cb87de2953d0f9', 'type': 'row', 'Auto ID': 2, 'C
reated At': '2022-11-13T12:38:31.509Z', 'Updated At': '2022-11-13T12:38:31.509Z', 'createdAt': '2022-11-13T12:38:
31.875Z', 'updatedAt': '2022-11-13T12:38:31.875Z', 'Created By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae9
4026d75a1', 'primaryDisplay': 'thomas@hua.gr'}], 'Updated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae940
26d75a1', 'primaryDisplay': 'thomas@hua.gr'}]}]
[{'_id': 'ro_ta_fea99be2f08b45d691cb87de2953d0f9_a0826f72896c4ba9b79ab0d902092512', 'notes': '', 'title': 'Profes
sor', 'department': 'Informatics and Telematics', 'idNumber': '83875476', 'surname': 'Kamalakis', 'givenName': 'T
homas', 'tableId': 'ta_fea99be2f08b45d691cb87de2953d0f9', 'Auto ID': 1, 'Created At': '2022-11-06T10:52:56.424Z',
'Updated At': '2022-11-06T10:52:56.424Z', 'type': 'row', 'createdAt': '2022-11-06T10:52:57.003Z', 'updatedAt': '
2022-11-06T10:52:57.003Z', 'Created By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', 'primaryDisp
lay': 'thomas@hua.gr'}], 'Updated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', 'primaryDispla
y': 'thomas@hua.gr'}]}]
>>>

Ενημέρωση

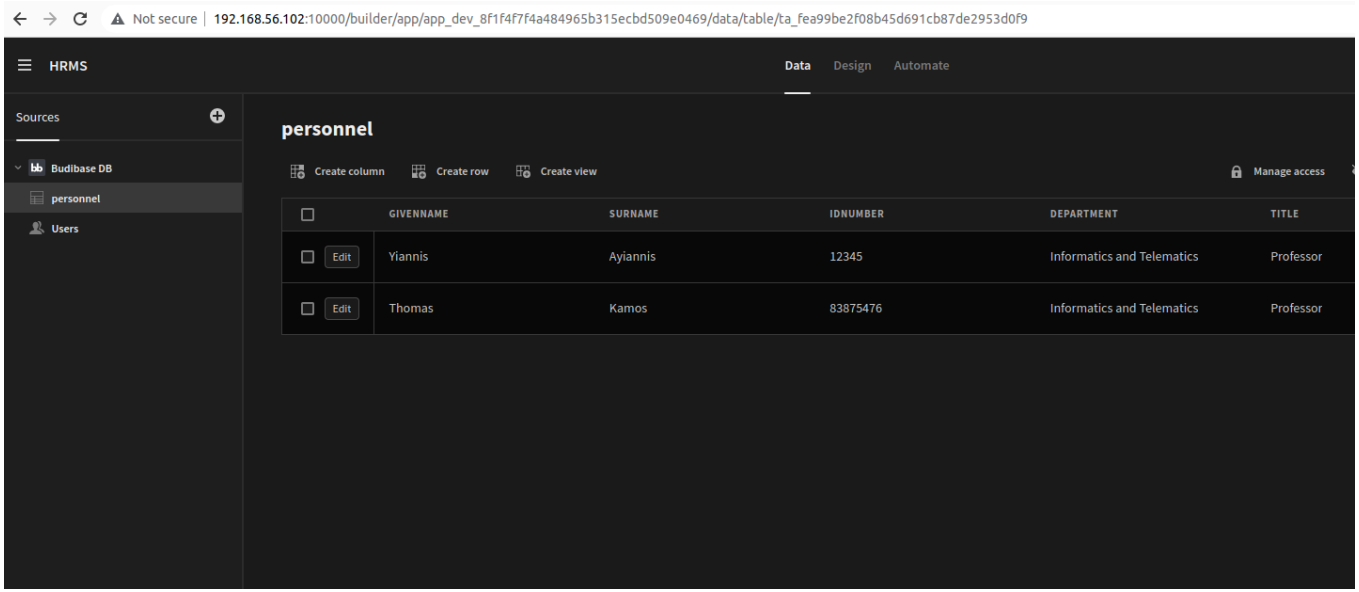
```
69 def update_row(self, table_name, app_name, query_dict, value_dict):
70
71     rows = self.search_rows(table_name, app_name, query_dict)
72     if len(rows) == 1:
73         table_id = self.table_ids[ table_name ]
74         app_id = self.app_map[ app_name ]
75         row_id = rows[0]['_id']
76         url = self.master_url + '/tables/' + table_id + '/rows/' + row_id
77         headers = self.post_headers( app_id = app_id )
78         response = requests.put(url, headers = headers, json = value_dict)
79         return json.loads(response.text)[ 'data' ]
80     else:
81         raise ValueError('Number of matching rows must be exactly 1 not ' + str(len(rows)) + '.')
```

Listing: budi.py

```
1 from budi import budiapi
2 b = budiapi()
3
4 r = b.update_row('personnel', 'HRMS',
5                   {'surname': 'Kamalakis' },
6                   {'surname' : 'Kamos'})
7 print(r)
```

Listing: updaterow.py

Ενημέρωση




```
thomas@kirklaptop:~/Documents/mscpython/code/lecture8$ python3 -i updaterow.py
{'_id': 'ro_ta_fea99be2f08b45d691cb87de2953d0f9_a0826f72896c4ba9b79ab0d902092512', 'notes': '', 'title': 'Profess
or', 'department': 'Informatics and Telematics', 'idNumber': '83875476', 'surname': 'Kamos', 'givenName': 'Thomas
', 'tableId': 'ta_fea99be2f08b45d691cb87de2953d0f9', 'Auto ID': 1, 'Created At': '2022-11-06T10:52:56.424Z', 'Upd
ated At': '2022-11-13T13:34:32.030Z', 'type': 'row', 'Created By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2a
e94026d75a1', '_rev': '2-be9845daefd0459fe36206e3dfd40ac4', 'createdAt': 1667730273312, 'email': 'thomas@hua.gr',
 'builder': {'global': True}, 'admin': {'global': True}, 'tenantId': 'default', 'status': 'active', 'updatedAt':
'2022-11-13T12:18:19.760Z', 'dayPassRecordedAt': '2022-11-13T12:18:19.760Z', 'roleId': 'ADMIN', 'tableId': 'ta_us
ers'}], 'Updated By': [{'_id': 'ro_ta_users_us_7271ae1007bd4522a5c2ae94026d75a1', '_rev': '2-be9845daefd0459fe362
06e3dfd40ac4', 'createdAt': 1667730273312, 'email': 'thomas@hua.gr', 'builder': {'global': True}, 'admin': {'glob
al': True}, 'tenantId': 'default', 'status': 'active', 'updatedAt': '2022-11-13T12:18:19.760Z', 'dayPassRecordedA
t': '2022-11-13T12:18:19.760Z', 'roleId': 'ADMIN', 'tableId': 'ta_users'}]}
>>>
```