

U5CRANMFV : ``` d["Jan. 5, 2017"] Out[43]:

```
{'issue': u'important',  
'url': '<http://www.roga.html>'}  
...
```

if in the dictionary I want the date to add another `issue` and the `url` how to do it properly?

U5CRANMFV : ``` d["Jan. 5, 2017"] Out[43]:

```
{'issue': [u'important', u"very important"],  
'url': ['<http://www.roga.html>', "<http://www.kopita.com|www.kopita.com>"]}```
```

I think that this kind of suit me, but I don't know how to do it...

U5NMSURAQ : ```>>> from collections import defaultdict

```
>>> d = defaultdict(list)
```

```
>>> d['issue'].append('important')
```

```
>>> d['issue'].append('very important')
```

```
>>> d['url'].append('<http://google.com>')
```

```
>>> d['url'].append('<http://yahoo.com>')
```

```
>>> d
```

```
defaultdict(<class 'list'>, {'issue': ['important', 'very important'], 'url': ['<http://google.com>', '<http://yahoo.com>']})
```

```
>>> dict(d)
```

```
{'issue': ['important', 'very important'], 'url': ['<http://google.com>', '<http://yahoo.com>']}
```

```
...
```

U5CRANMFV : Thank you :clap::blush:

U5CRANMFV : Did not understand until the end. I hope to help.```z = {u'Jan. 5, 2017': {'url\_\_result':

```
['<http://www.something.htm>'], 'issue': [u'something_issue']}]}
```

```
...
```

For example I have the following date is the same.

I want to add to `z` -

```
```url__result = "<http://www.google.com|www.google.com>"
```

```
issue = "import"
```

```
...
```

What should I use?

U5VGKQ2SY : and the date stay the same? So Jan 5, 2017 would have two {url\_\_result, issue} sets?

U5CRANMFV : <@U5VGKQ2SY> I want this ``` d["Jan. 5, 2017"] Out[43]:

```
{'issue': [u'important', u"very important"],  
'url': ['<http://www.roga.html>', "<http://www.kopita.com|www.kopita.com>"]}```
```

U5CRANMFV : I have a few lists with these dates, I need a dictionary, the Date can be the same as each other, so I need to get my dictionary are not overwritten as I have now, and added new values.

U5VGKQ2SY : actually, take another look at leovp's solution.

U5CRANMFV : The decision is good, but I can't figure out how to make the key "10.01.2007" was {'issue': ['important'], 'url': ['<http://google.com>']}. And what if the next date goes "10.01.2007" the data was added to the list to "important" to "<http://google.com|google.com>". I have overwritten now.

U5CRANMFV : ```from collections import defaultdict d = defaultdict(list)

```
d['issue'].append('important')
```

```
d['url'].append('<http://google.com>')
```

```
z = {}
```

```
z["10.01.2005"] = dict(d)
```

```
z
```

```
Out[13]: {'10.01.2005': {'issue': ['important'], 'url': ['<http://google.com>']}}
```

```
...
```

U5CRANMFV : how to continue to add data, I do not understand.

U5CRANMFV : `append` to the list...for sure...now try

U5VGKQ2SY : So there can be multiple `{'10.01.2005': {'issue': ['important1', 'important2'], 'url': ['url1', '<http://google.com>']}}` ?

U5CRANMFV : all. tired better not to sit at the computer. go to the beach.

U5VGKQ2SY : so if important3 and url3 came in, there are different "10.01.2005" that it can go into?

U5CRANMFV : append, it list.

U5CRANMFV : `z[date]["issue"].append(df["issue"][i])` everything is very easy.))

U5VGKQ2SY : I don't think you can use a dictionary for this. Hopefully someone else can confirm, but I think you're looking at using a list instead.

U5NMSURQA : z[date]['issue'] is probably a list

U5CRANMFV : Exactly :)

U1BP42MRS : In PyCharm, can you have the project run kick off a bunch of things at once? For example, I want to run Django, Celery Workers, and Celery Beat

U28MDQRL2 : If you find out that share it with me. I have to open like 5 command windows to get started

U29163YQH : `subprocess.open` at the start of the script ? :smile:

U1BP42MRS : haha, no kidding. I guess if you don't need a bunch of workers you can run beat and the worker:

...

celery -A &lt;your proj> worker --beat -l debug --scheduler django

...

U5NMSURQA : Is that it?

U1BP42MRS : I think those block before starting

U5NMSURQA : `&`

U0NRYQNAZ : <@U1BP42MRS> i use honcho to do that now.

U28MDQRL2 : I tried to run the worker and beat with django-beat and it didn't seem to work

U0NRYQNAZ : <<https://github.com/nickstenning/honcho>>

U28MDQRL2 : And running django in a docker container involves too much hacking. I prefer to just run postgres and redis in it

U0NRYQNAZ : but you can use it as a library.

U1BP42MRS : Looks like celery got rid of the `--autoreload` option :cry: