What Django can learn from web2py

By Samuel Marks

au.linkedin.com/in/samuelmarks

Why move from web2py to Django?

- Number of web2py jobs available on careersone, seek and freelancer: 0
- Number of Django jobs available on aforementioned job sites: 36

Maybe I should learn PHP instead? >.<

Web2py feature highlights (pt1)

- In-built ticketing system for errors
- In-built RBAC
- In-built services support for JSONRPC, XMLRPC, REST (CSV, JSON, XML) and SOAP
- Database Abstraction Layer (DAL) is used rather than an ORM
- There‘s a default for everything, e.g.: create a controller function and you have a default view

Web2py feature highlights (pt2)

Decorators for doing useful tasks, e.g. prepending controller functions:

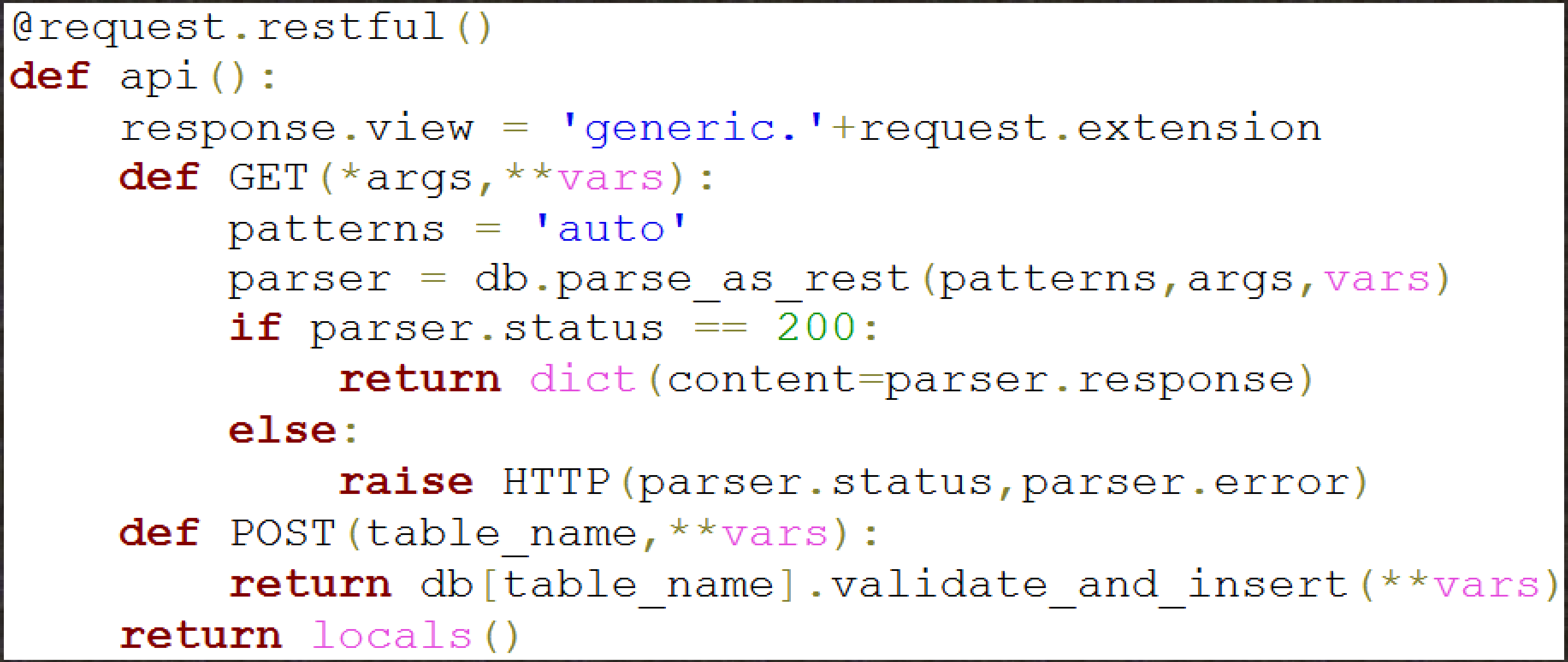
- @auth.requires_login()
- @auth.requires_permission('read', secrets)
- @auth.requires(auth.user_id==1 or request.client=='127.0.0.1', requires_login=True)
- @auth.requires_membership('agents')

Web2py feature highlights (pt3)

- @service.run
- @service.xml
- @service.json
- @service.rss
- @service.csv
- @service.xmlrpc
- @service.jsonrpc
- @service.amfrpc3
- @service.soap

au.linkedin.com/in/samuelmarks

Example usage (REST)



```python
@request.restful()
def api():
    response.view = 'generic.'+request.extension
    def GET(*args,**vars):
        patterns = 'auto'
        parser = db.parse_as_rest(patterns,args,vars)
        if parser.status == 200:
            return dict(content=parser.response)
        else:
            raise HTTP(parser.status,parser.error)
    def POST(table_name,**vars):
        return db[table_name].validate_and_insert(**vars)
    return locals()
```

Are there Django equivalents with similar power + simplicity?

□ [edit this page] □

au.linkedin.com/in/samuelmarks