```python
"""
request_handlers/server.py is the file that handles requests from the
bottle server. It determines the requests and calls the respective
function from another class.
"""
from UniversalRequestHandler import UniversalRequestHandler
import config.ConfigConnector as ConfigConnector
from AdminRequestHandler import AdminRequestHandler
from StudentRequestHandler import StudentRequestHandler
from sql.PostgresConnector import PostgresConnector


config_file = 'db.config'


class ServerRequestHandler:
    def __init__(self):
        self.postgres_connector = PostgresConnector(config_file)
        self.admin_request_handler = AdminRequestHandler(self.postgres_connector)
        self.universal_request_handler = UniversalRequestHandler(self.postgres_connector)
        self.student_request_handler = StudentRequestHandler(self.postgres_connector)

        self.REQUEST_OPTIONS = {
            'universal.login': self.universal_request_handler.login,
            'universal.create_account': self.universal_request_handler.create_account,
            'admin.get_students': self.admin_request_handler.get_students,
            'admin.get_student_location': self.admin_request_handler.get_student_location,
            'admin.get_beacons': self.admin_request_handler.get_beacons,
            'admin.add_admin': self.admin_request_handler.add_admin,
            'macos.admin.get_beacons': self.admin_request_handler.get_beacons_macos,
            'admin.get_zones': self.admin_request_handler.get_zones,
            'admin.edit_beacon': self.admin_request_handler.edit_beacon,
            'admin.create_beacon': self.admin_request_handler.create_beacon,
            'admin.delete_beacon': self.admin_request_handler.delete_beacon,
            'student.update_location': self.student_request_handler.update_location,
            'student.get_info': self.student_request_handler.get_info
        }

    def handle_request(self, request):
        """
        Calls function based on the request type. Firstly,
        checks if type exists, then calls functions as defined
        in REQUEST_OPTIONS.

        :returns a JSON object (python dictionary)
            depending on function called.
        """
        if request['type'] in self.REQUEST_OPTIONS.keys():
            # ** denotes kwargs: keyword arguments.
            return self.REQUEST_OPTIONS[request['type']](**request['args'])
        else:
            return {
                'successful': False,
                'reason': 'Request type \'' + request['type'] + '\' does not exist.'
            }
```