```
#import the necessary modules(libraries)
import json, unittest, datetime
#use the open function to open the json files
with open("./data-1.json","r") as f:
    jsonData1 = json.load(f)
with open("./data-2.json","r") as f:
    jsonData2 = json.load(f)
with open("./data-result.json","r") as f:
    jsonExpectedResult = json.load(f)
#convert json data from format 1 to a unified format
def convertFromFormat1(jsonObject):

    # IMPLEMENT: Conversion From Type 1
    locationParts = jsonObject['location'].split('/')

    result = {
        'deviceID': jsonObject['deviceID'],
        'deviceType': jsonObject['deviceType'],
        'timestamp': jsonObject['timestamp'],
        'location': {
            'country': locationParts[0],
            'city': locationParts[1],
            'area': locationParts[2],
            'factory': locationParts[3],
            'section': locationParts[4]
        },
        'data': {
            'status': jsonObject['operationStatus'],
            'temperature': jsonObject['temp']
        }
    }

    return result


def convertFromFormat2(jsonObject):

    # IMPLEMENT: Conversion From Type 2
    date = datetime.datetime.strptime(jsonObject['timestamp'],
                                      '%Y-%m-%dT%H:%M:%S.%fZ')
    timestamp = int(
        (date - datetime.datetime(1970, 1, 1)).total_seconds() * 1000)

    result = {
        'deviceID': jsonObject['device']['id'],
        'deviceType': jsonObject['device']['type'],
        'timestamp': timestamp,
        'location': {
            'country': jsonObject['country'],
            'city': jsonObject['city'],
            'area': jsonObject['area'],
            'factory': jsonObject['factory'],
```

```
            'section': jsonObject['section']
        },
        'data': jsonObject['data']
    }

    return result

def main (jsonObject):

    result = {}

    if (jsonObject.get('device') == None):
        result = convertFromFormat1(jsonObject)
    else:
        result = convertFromFormat2(jsonObject)
    return result

#Test cases using unittest
class TestSolution(unittest.TestCase):

    def test_sanity(self):
#convert json data to python objects using json.loads and checks if they match
        result = json.loads(json.dumps(jsonExpectedResult))
        self.assertEqual(
            result,
            jsonExpectedResult
        )

    def test_dataType1(self):
#convert the json data from format 1 to unfied format and compare it with expected result
        result = main (jsonData1)
        self.assertEqual(
            result,
            jsonExpectedResult,
            'Converting from Type 1 failed'
        )

    def test_dataType2(self):
#convert the json data from format 1 to unified format and compare it with expected result
        result = main (jsonData2)
        self.assertEqual(
            result,
            jsonExpectedResult,
            'Converting from Type 2 failed'
        )
if __name__ == '__main__':
    #run the unittest
    unittest.main()
```