Senior Developer Pre-Interview Test Assignment

Part 1

- Create a REST API service as per requirements on the following page in Java (or PHP)
- You are free to utilise any open source frameworks or libraries you like
- Add Basic Auth to REST API with user/password
    - *Can be just hard coded as "demo:pwd1234"*
- Write unit tests
    - *Full code coverage is not necessarily required*
- **Submit**
    - Source code, compiled file, build scripts, project files and all accompanying libraries
    - Should be able to unzip files to a folder and run a start script/bat file to bring up api

Part 2

- Create a new PHP class PurchaseOrderService so that it wraps the above REST API call
- Ensure the code below for existing TotalsCalculator class continues to work without change
- Use good object oriented design
- **Submit**
    - PurchaseOrderService.php along with all other relevant classes
        - *Keep classes in same namespace (or nest below)*
        - *Supply copies of any libraries you may use (if any)*
    - Should be able to copy files on server running TotalsCalculator and just work

```php
<?php
namespace BearClaw\Warehousing;

class TotalsCalculator
{
    /**
     * @param array $ids
     */
    public function generateReport(array $ids) {
        $service = new PurchaseOrderService();
        $result = $service->calculateTotals($ids);
        foreach($result as $record) {
            echo "Product Type " . $record['product_type_id'] . " has total of " .
$record['total']."\n";
        }
    }

}
```

Service Requirements

POST http://localhost/test

Request Body

{ "purchase_order_ids": [2344, 2345, 2346] }

Requirements for this Service

- Call the below API asynchronously to get the “PurchaseOrder” for each id in the array
  - **GET** https://api.cartoncloud.com.au/CartonCloud_Demo/PurchaseOrders/{id}?version=5&associated=true
  - Where {id} is an integer
  - Basic Auth User: interview-test@cartoncloud.com.au
  - Basic Auth Password: test123456
- For all “PurchaseOrderProduct” records across the all the above “PurchaseOrders” calculate the "total" grouped by product_type_id
- The formula to calculate "total" will vary by product_type_id as below. Ensure design allows for easy adding of additional mappings

product_type_id	calculation method
1	By Weight
2	By Volume
3	By Weight

- Calculation method are as follows, use object oriented design to allow for easy adding for additional calculation methods potentially with much more complex logic
  - By Weight sum(unit_quantity_initial x Product.weight)
  - By Volume sum(unit_quantity_initial x Product.volume)

Response Body

{"result": [
{"product_type_id": 1, "total": 41.5},
{"product_type_id": 2, "total": 13.8},
{"product_type_id": 3, "total": 25.0}] }