TreeView Project

1/ Application structure

Requirements:

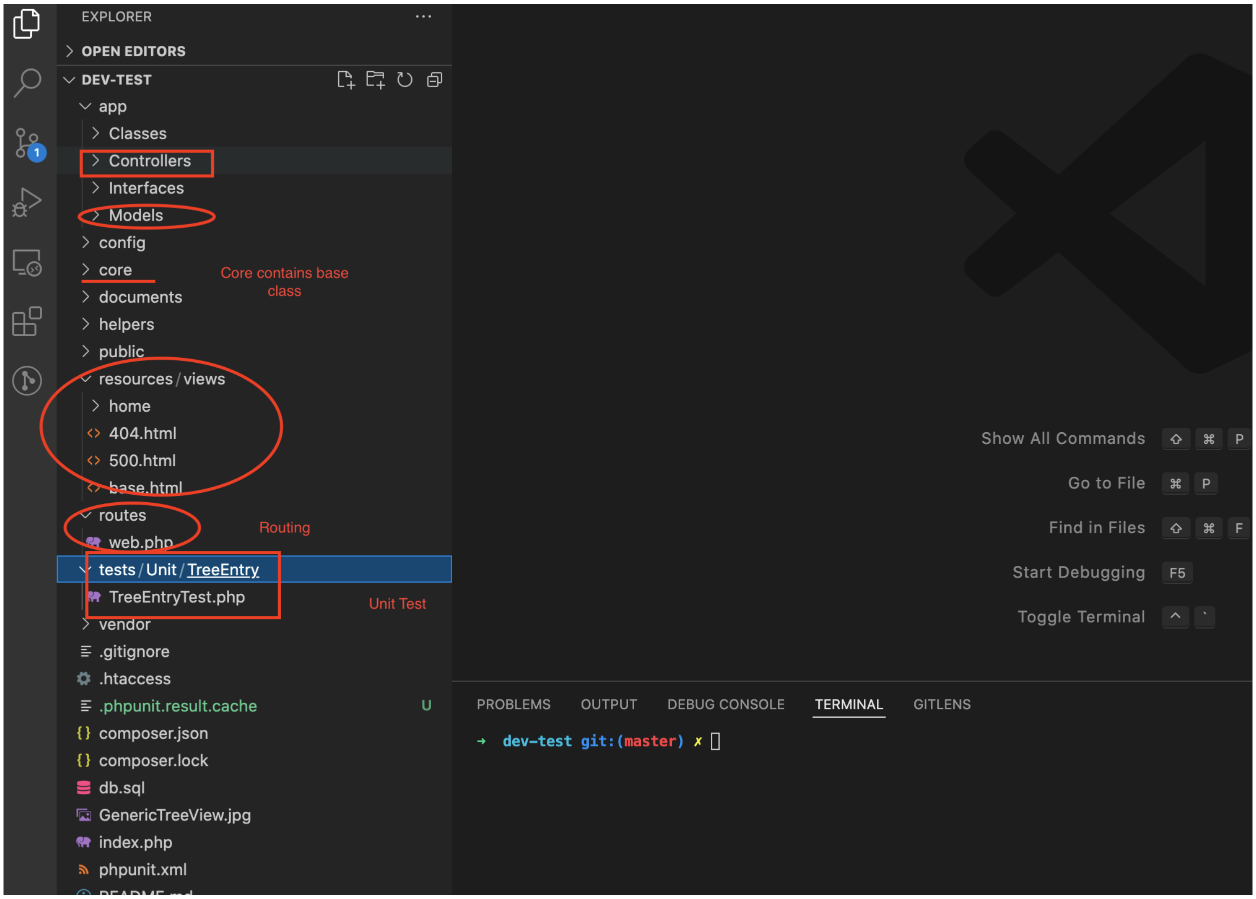
PHP >= 7.2

Package used in project:

twig/twig: template engine
symfony/routing: used to routing
symfony/http-foundation: get incoming request and response
doctrine/dbal: database abstraction layer
phpunit/phpunit: write unit test
guzzlehttp/guzzle: used to call http request to test unit test case.

Using composer to manage the packages.

I have built a MVC (Model - View - Controller) structure.



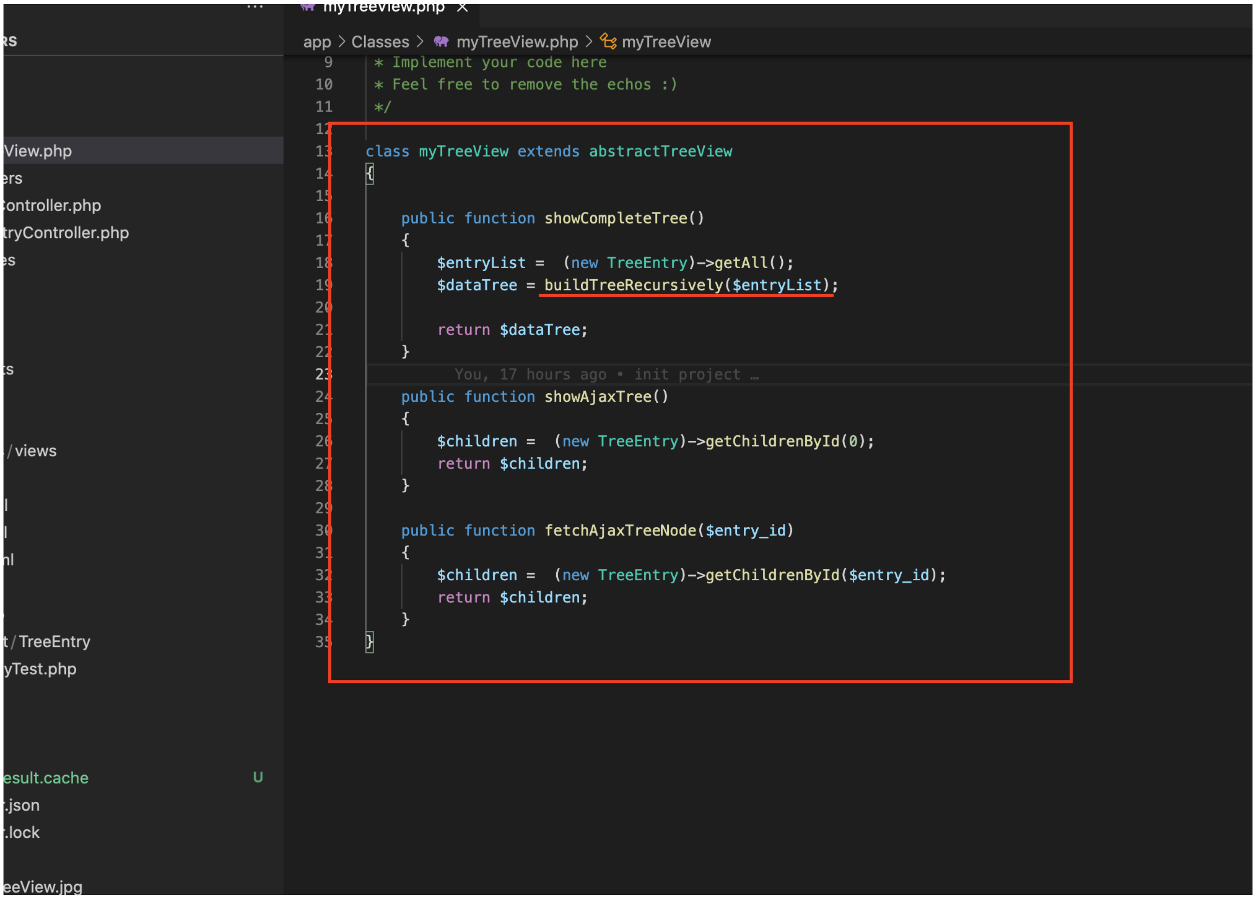
```
➜ dev-test git:(master) ✗
```

2/ Explain the code

Firstly, We define our routes in ***/routes/web.php***
Secondly, take a look to ***/app/Controllers/TreeEntryController.php***, all method of this controller will return a json.

In this project, I don't embed html code into php function. Instead, we use process a json data in frontend.

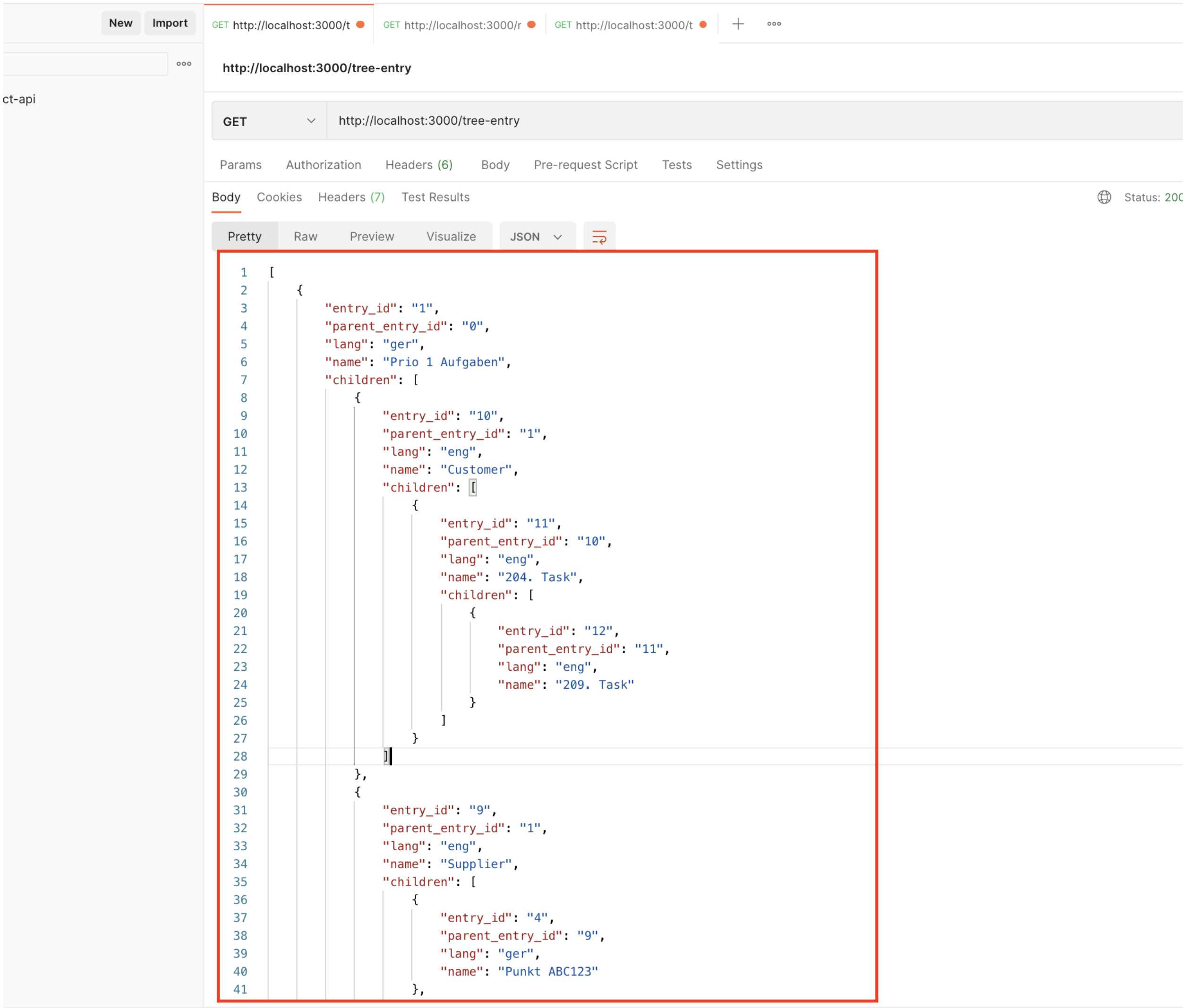
So now we focus on ***/app/Classes/myTreeView.php***



```
 * Implement your code here
 * Feel free to remove the echos :)
 */

class myTreeView extends abstractTreeView
{

    public function showCompleteTree()
    {
        $entryList =  (new TreeEntry)->getAll();
        $dataTree = buildTreeRecursively($entryList);

        return $dataTree;
    }

    public function showAjaxTree()
    {
        $children =  (new TreeEntry)->getChildrenById(0);
        return $children;
    }

    public function fetchAjaxTreeNode($entry_id)
    {
        $children =  (new TreeEntry)->getChildrenById($entry_id);
        return $children;
    }
}
```

showCompleteTree() Method:
Using buildTreeRecursively function to convert normal list array to treedata.

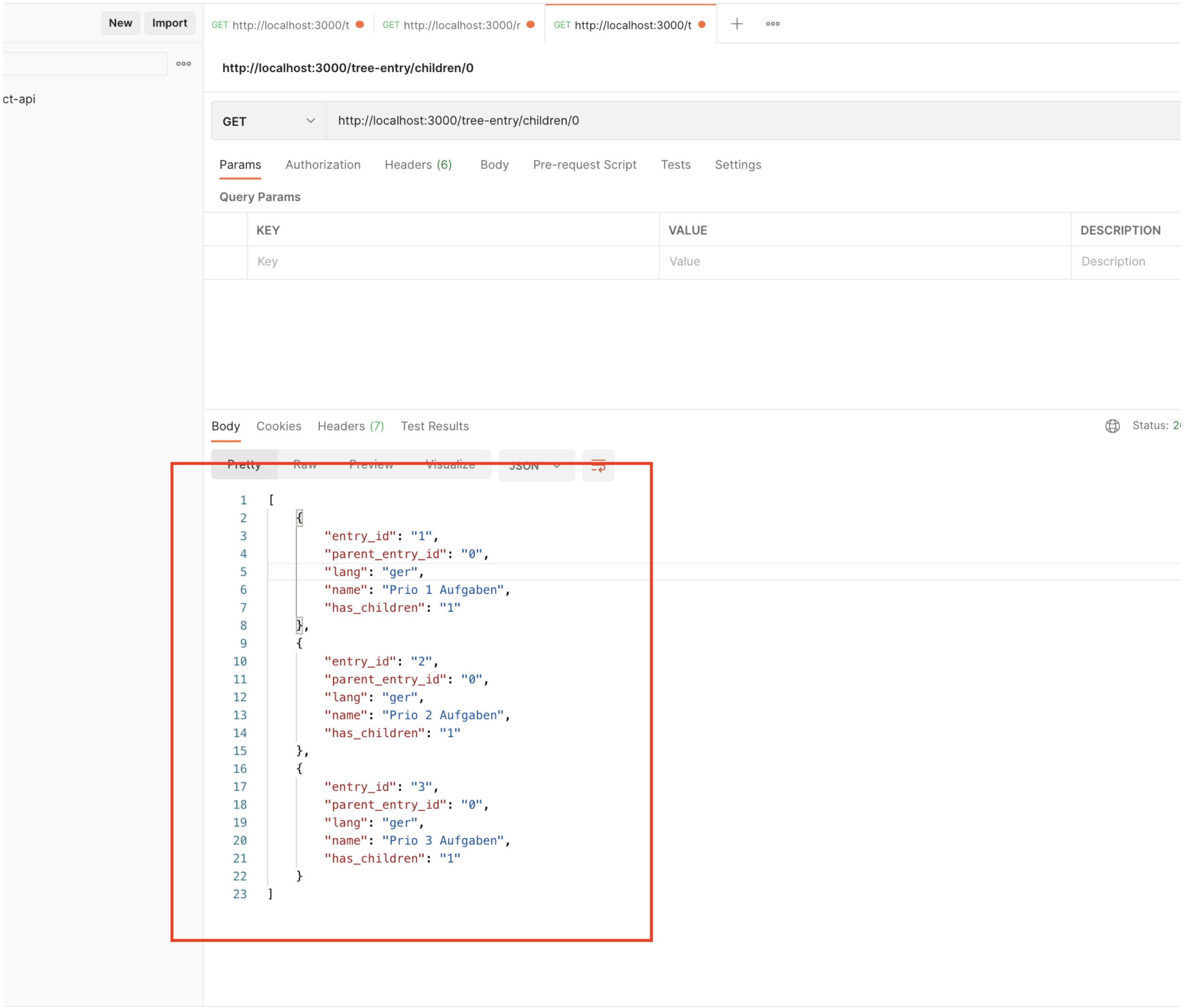
Tree data format:



```
[
    {
        "entry_id": "1",
        "parent_entry_id": "0",
        "lang": "ger",
        "name": "Prio 1 Aufgaben",
        "children": [
            {
                "entry_id": "10",
                "parent_entry_id": "1",
                "lang": "eng",
                "name": "Customer",
                "children": [
                    {
                        "entry_id": "11",
                        "parent_entry_id": "10",
                        "lang": "eng",
                        "name": "204. Task",
                        "children": [
                            {
                                "entry_id": "12",
                                "parent_entry_id": "11",
                                "lang": "eng",
                                "name": "209. Task"
                            }
                        ]
                    }
                ]
            },
            {
                "entry_id": "9",
                "parent_entry_id": "1",
                "lang": "eng",
                "name": "Supplier",
                "children": [
                    {
                        "entry_id": "4",
                        "parent_entry_id": "9",
                        "lang": "ger",
                        "name": "Punkt ABC123"
                    },
```

showAjaxTree() and fetchAjaxTreeNode() Method:

Data format:



http://localhost:3000/tree-entry/children/0

GET http://localhost:3000/tree-entry/children/0

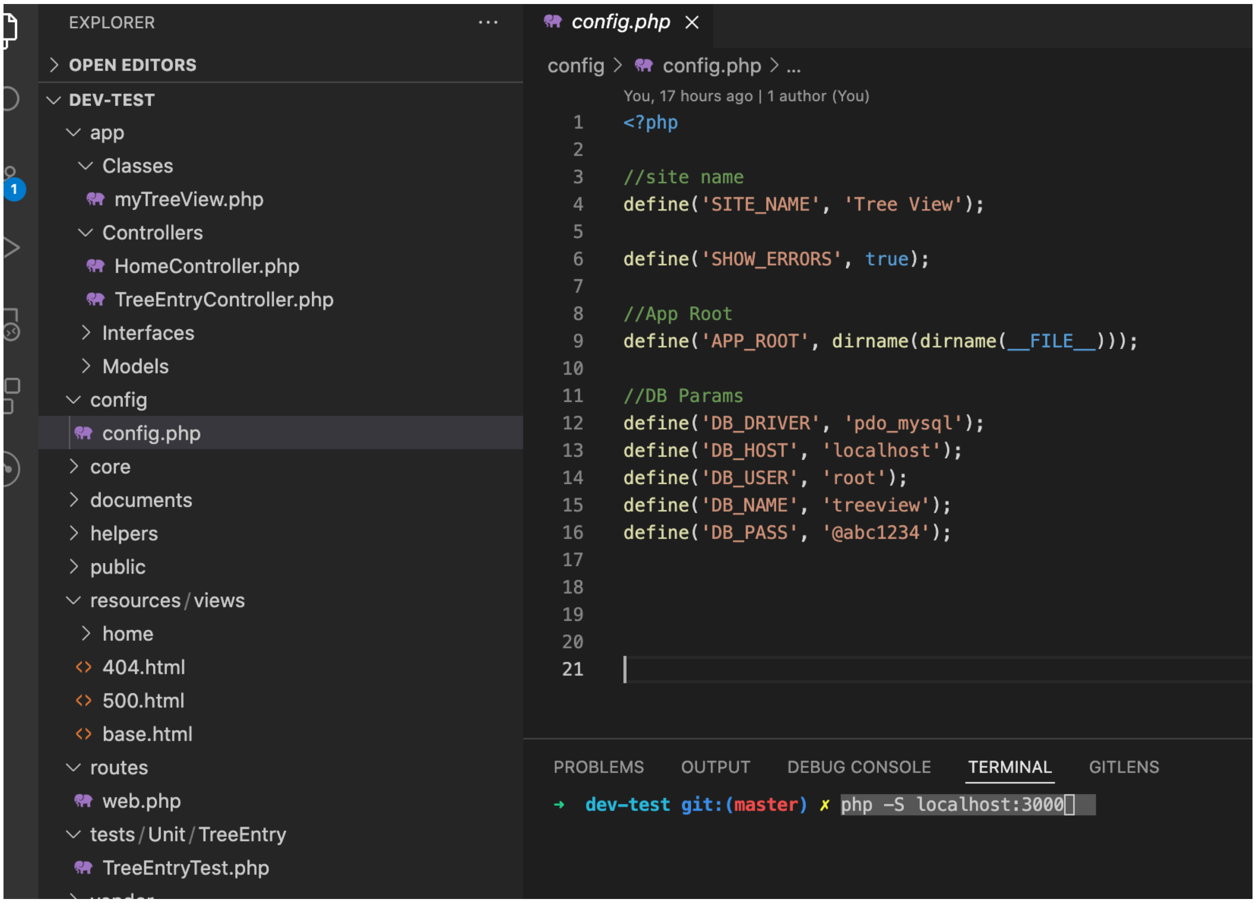
```
[
    {
        "entry_id": "1",
        "parent_entry_id": "0",
        "lang": "ger",
        "name": "Prio 1 Aufgaben",
        "has_children": "1"
    },
    {
        "entry_id": "2",
        "parent_entry_id": "0",
        "lang": "ger",
        "name": "Prio 2 Aufgaben",
        "has_children": "1"
    },
    {
        "entry_id": "3",
        "parent_entry_id": "0",
        "lang": "ger",
        "name": "Prio 3 Aufgaben",
        "has_children": "1"
    }
]
```

3/ How to run this project

Run composer:

```
composer install
```

Configuring your database in ***/config/config.php***



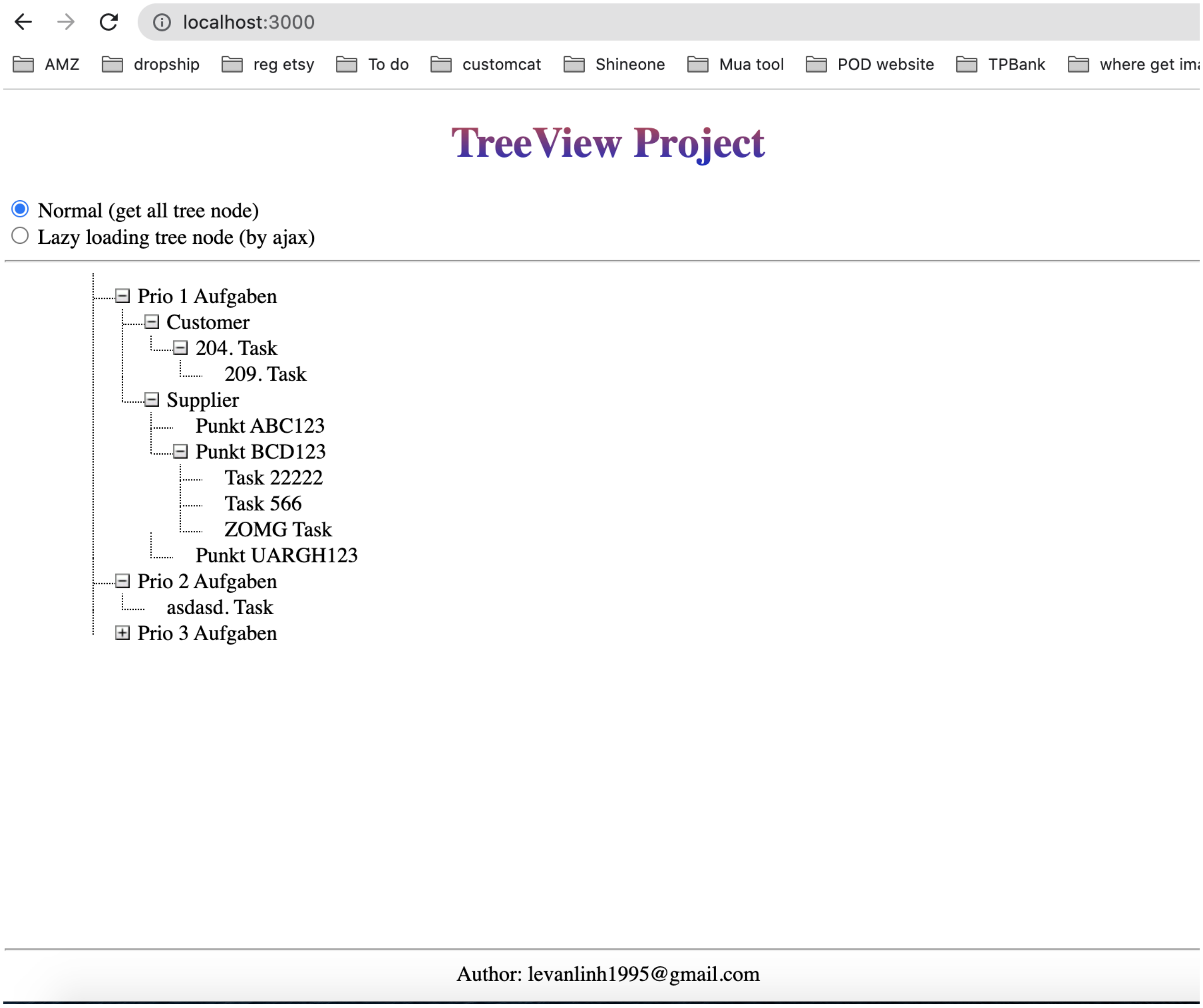
```php
<?php

//site name
define('SITE_NAME', 'Tree View');

define('SHOW_ERRORS', true);

//App Root
define('APP_ROOT', dirname(dirname(__FILE__)));

//DB Params
define('DB_DRIVER', 'pdo_mysql');
define('DB_HOST', 'localhost');
define('DB_USER', 'root');
define('DB_NAME', 'treeview');
define('DB_PASS', '@abc1234');
```

```
➜ dev-test git:(master) ✗ php -S localhost:3000
```

Run dev environment:

php -S localhost:3000

Final result:



TreeView Project

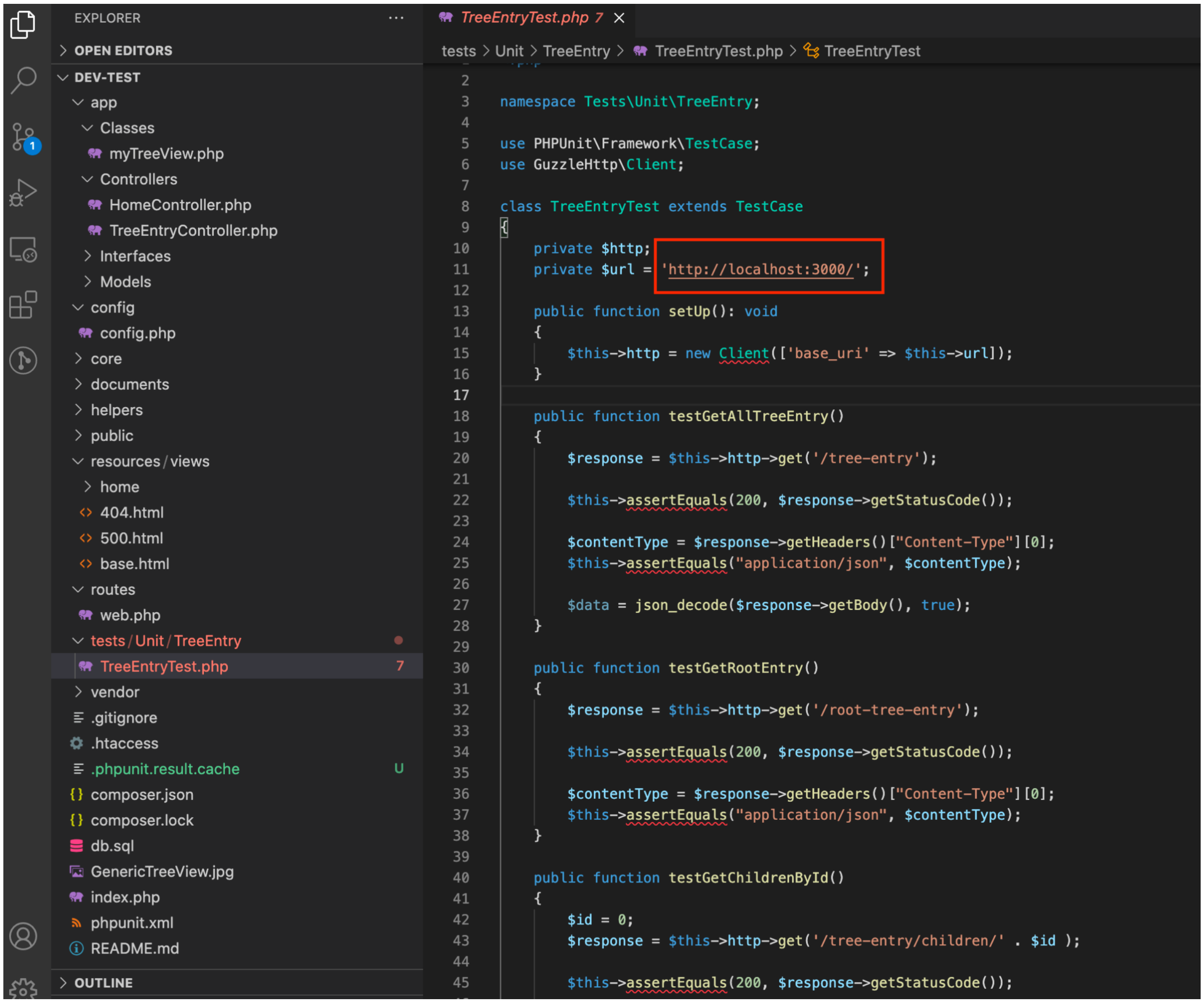
● Normal (get all tree node)
○ Lazy loading tree node (by ajax)

- ⊟ Prio 1 Aufgaben
  - ⊟ Customer
    - ⊟ 204. Task
      - 209. Task
  - ⊟ Supplier
    - Punkt ABC123
    - ⊟ Punkt BCD123
      - Task 22222
      - Task 566
      - ZOMG Task
    - Punkt UARGH123
- ⊟ Prio 2 Aufgaben
  - asdasd. Task
- ⊞ Prio 3 Aufgaben

Author: levanlinh1995@gmail.com

4/ Unit test

To run unit code, you need to edit URL in ***/tests/Unit/TreeEntry/TreeEntryTest.php***
Finally, run ***./vendor/bin/phpunit***



```php

namespace Tests\Unit\TreeEntry;

use PHPUnit\Framework\TestCase;
use GuzzleHttp\Client;

class TreeEntryTest extends TestCase
{
    private $http;
    private $url = 'http://localhost:3000/';

    public function setUp(): void
    {
        $this->http = new Client(['base_uri' => $this->url]);
    }

    public function testGetAllTreeEntry()
    {
        $response = $this->http->get('/tree-entry');

        $this->assertEquals(200, $response->getStatusCode());

        $contentType = $response->getHeaders()["Content-Type"][0];
        $this->assertEquals("application/json", $contentType);

        $data = json_decode($response->getBody(), true);
    }

    public function testGetRootEntry()
    {
        $response = $this->http->get('/root-tree-entry');

        $this->assertEquals(200, $response->getStatusCode());

        $contentType = $response->getHeaders()["Content-Type"][0];
        $this->assertEquals("application/json", $contentType);
    }

    public function testGetChildrenById()
    {
        $id = 0;
        $response = $this->http->get('/tree-entry/children/' . $id );

        $this->assertEquals(200, $response->getStatusCode());
```