

CodeIgniter Overview and Details



The "Internal Server Error" usually points to a configuration issue within the Apache server or the PHP application. Let's troubleshoot this step by step:

Steps to Resolve:

1. **Check Apache Error Logs:** Inside your Docker container, check the Apache error logs to see the exact cause of the error. You can do this by connecting to the running container:

```
bash
```

```
docker exec -it ci3_app bash
```

Once inside the container, check the Apache logs:

```
bash
```

```
cat /var/log/apache2/error.log
```

The log should give you more details about what is causing the "500 Internal Server Error."

2. **Verify File Permissions:** Ensure that the CodeIgniter application and system files have the correct permissions. Incorrect file permissions can lead to issues. The typical setup requires that:
 - The ``application/cache`` and ``application/logs`` directories should be writable by the web server (Apache).
 - Run the following commands inside your project directory (outside the container) to set proper permissions:

```
bash
```

```
chmod -R 775 application/cache
chmod -R 775 application/logs
```

3. **Ensure ``mod_rewrite`` is enabled:** CodeIgniter typically relies on the ``mod_rewrite`` Apache module for routing. Inside the Docker container, ensure that the module is enabled:

```
bash
```

```
docker exec -it ci3_app bash
a2enmod rewrite
```

```
service apache2 restart
```

Also, verify that the `.htaccess` file in the root of the CodeIgniter project (inside `geolocalis/`) is properly configured to rewrite URLs.

4. Check Database Connection: Verify that the database connection works correctly. If the database connection settings in `application/config/database.php` are incorrect, it could result in an internal server error.

- Double-check the `hostname`, `username`, `password`, and `database` in `database.php` to ensure they match the values in `docker-compose.yml`.
- Test the MySQL connection manually inside the container:

```
bash
```

```
docker exec -it ci3_db mysql -u linklztc_geolocalis -pgeolocalis -Dlinklztc_geolocalis
```

If the connection is successful, the database setup is fine. Otherwise, you may need to review the database credentials.

5. Increase PHP Error Reporting: Modify the `index.php` file in your CodeIgniter project to enable more detailed error reporting for debugging:

```
php
```

```
define('ENVIRONMENT', isset($_SERVER['CI_ENV']) ? $_SERVER['CI_ENV'] : 'development');

if (ENVIRONMENT === 'development') {
    error_reporting(-1);
    ini_set('display_errors', 1);
}
```

This should display PHP errors instead of a generic 500 error.

Next Steps:

1. Check the error logs (`/var/log/apache2/error.log`) to identify the exact cause.
2. Ensure proper permissions on cache and logs directories.
3. Ensure `mod_rewrite` is enabled and `.htaccess` is correctly configured.
4. Verify the database connection is correct.

Once you've checked these, let me know if you still need further assistance!