

Microblog

Developed by Debmalya Jash

Purpose	2
Github URL	2
Responsive Web Design	2
Functionality	2
Listing all blogs	2
Functionality Search blogs	4
Functionality get list of users	6
Create a simple post	7
View individual post	8
Update a simple blog	9
Display of Status Message	11
Display of Error Message	11
Menu	12
List of APIs	12
List of views (twig templates)	14
List of javascripts	14
API Web Service Changes	15
Functionality	15
Listing all posts	15
Create User table	15
Get list of users (Added functionality)	16
Create a simple blog	16
View individual blog	16
Delete a simple blog	16
Update a simple blog	17
Error handling	17
Validation	17
Performance	18
File List	18
Before release checklist	18
Improvement	21
Lazy fetching and pagination.	21
Reduce communication with server. Communication with server should be as minimum as possible.	21
Service /Template Controller	21

Developer : Debmalya Jash

Reduce database communication	21
Add test code	22
Twig macro	22
Prevent SQL injection	22
UTF-8 character handling	22

Purpose

Thanks for providing this opportunity to develop this micro-blog sample application. This document is created to describe the changes made for micro blog development.

Github URL

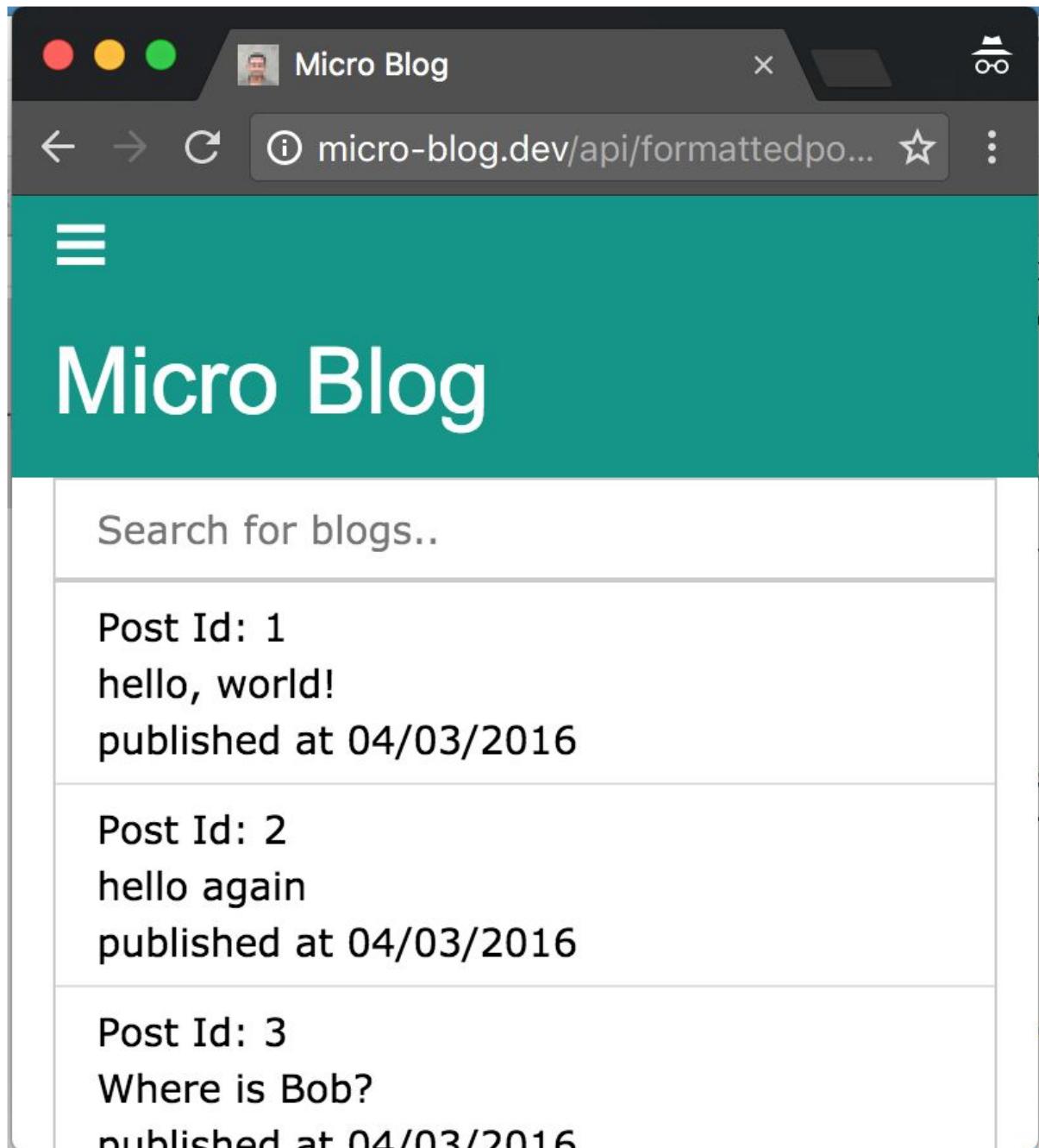
<https://github.com/debmalya/micro-blog-technical-test.git>

Responsive Web Design

Functionality

Listing all blogs

`/api/formattedposts/{format}`



Blogs.twig has following contents to display the blog posts.

```
{% if posts|length > 0 %}

<ul id="id01" class="w3-ul w3-border">

    {% for post in posts %}

        <li>{{ post.content|e }}</li>
```

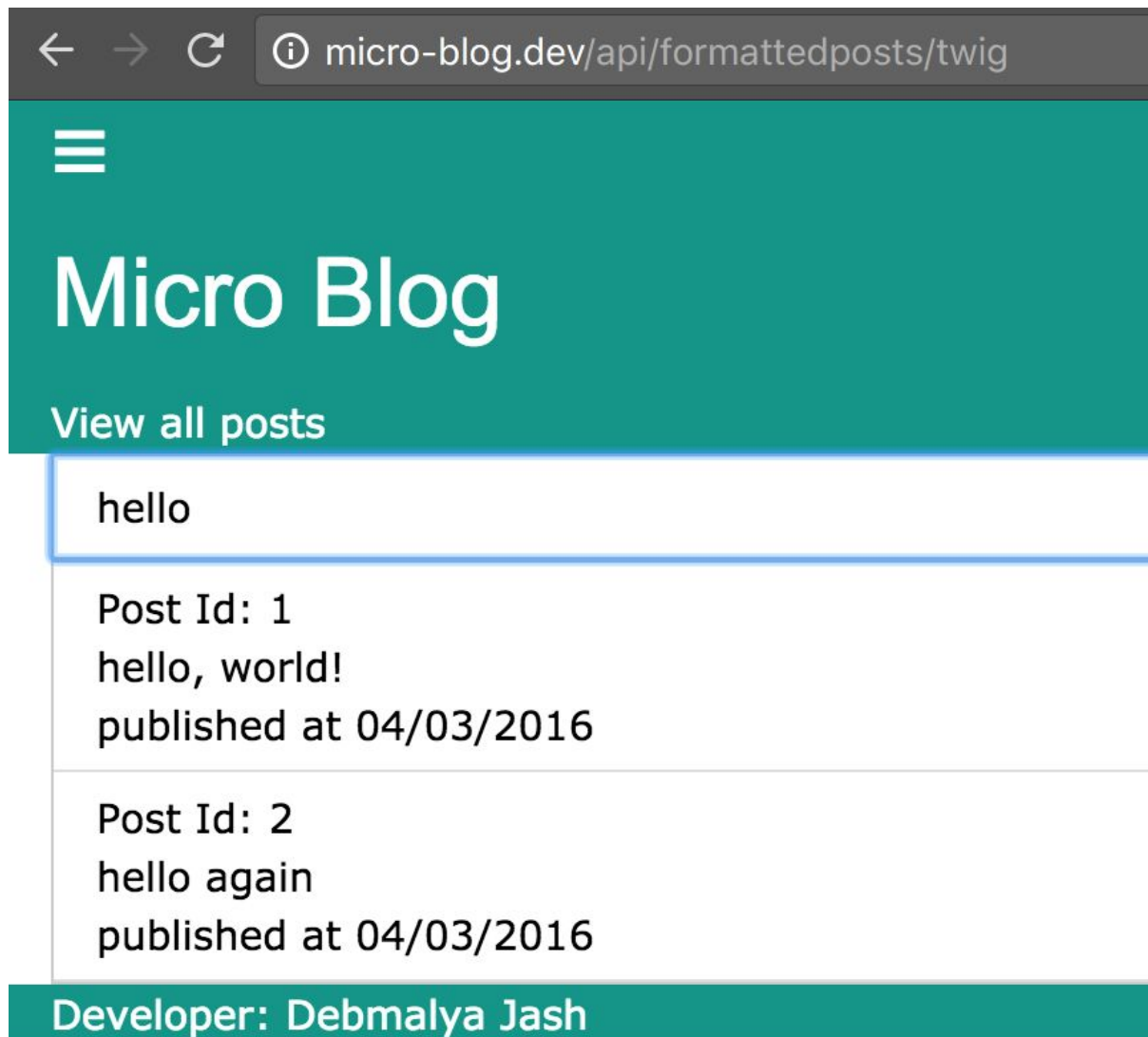
Developer : Debmalya Jash

```
{% endfor %}
```

```
</ul>
```

```
{% endif %}
```

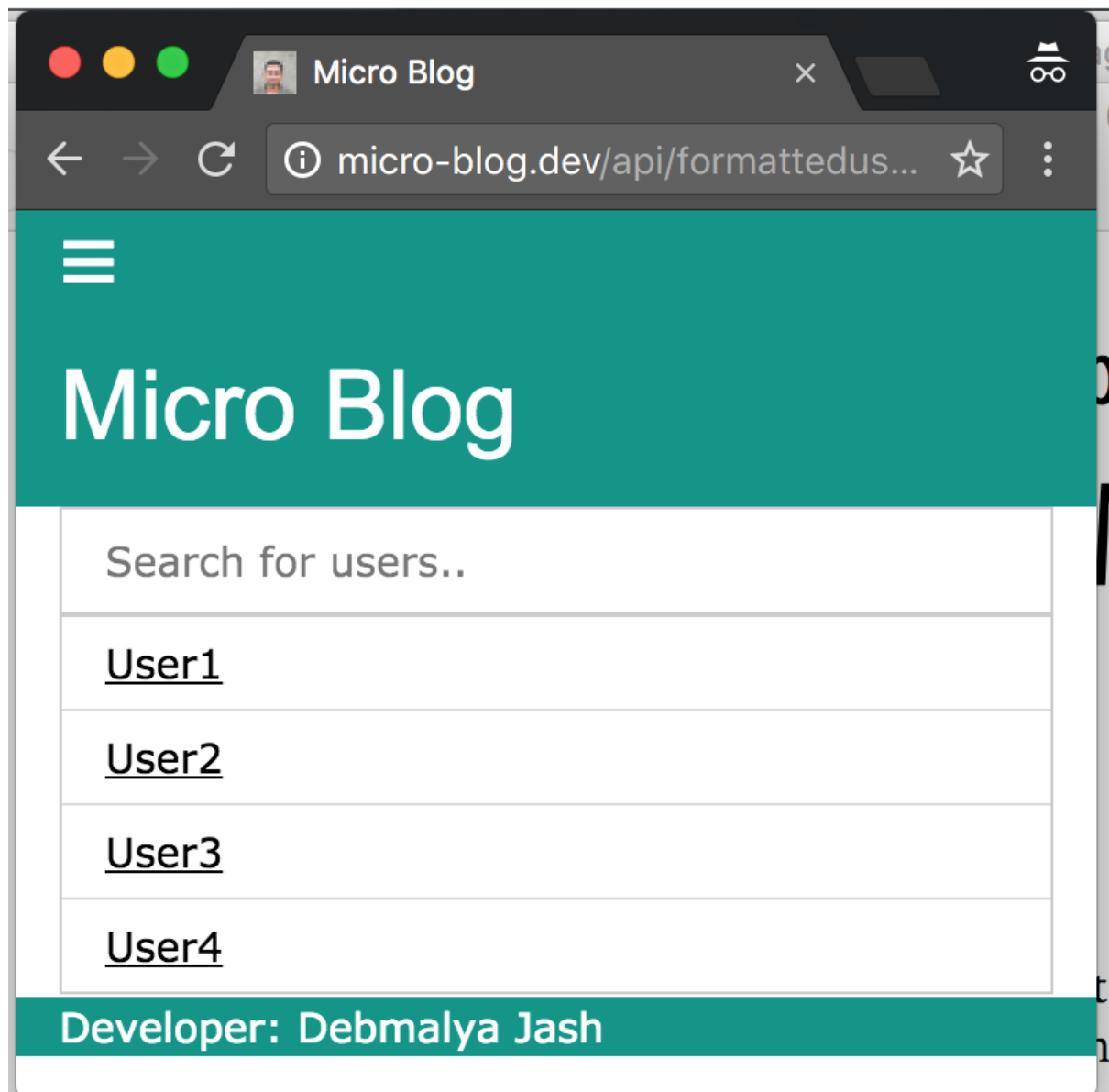
Functionality Search blogs



This search is case insensitive. I am using w3 css / data libraries for this.

Developer : Debmalya Jash

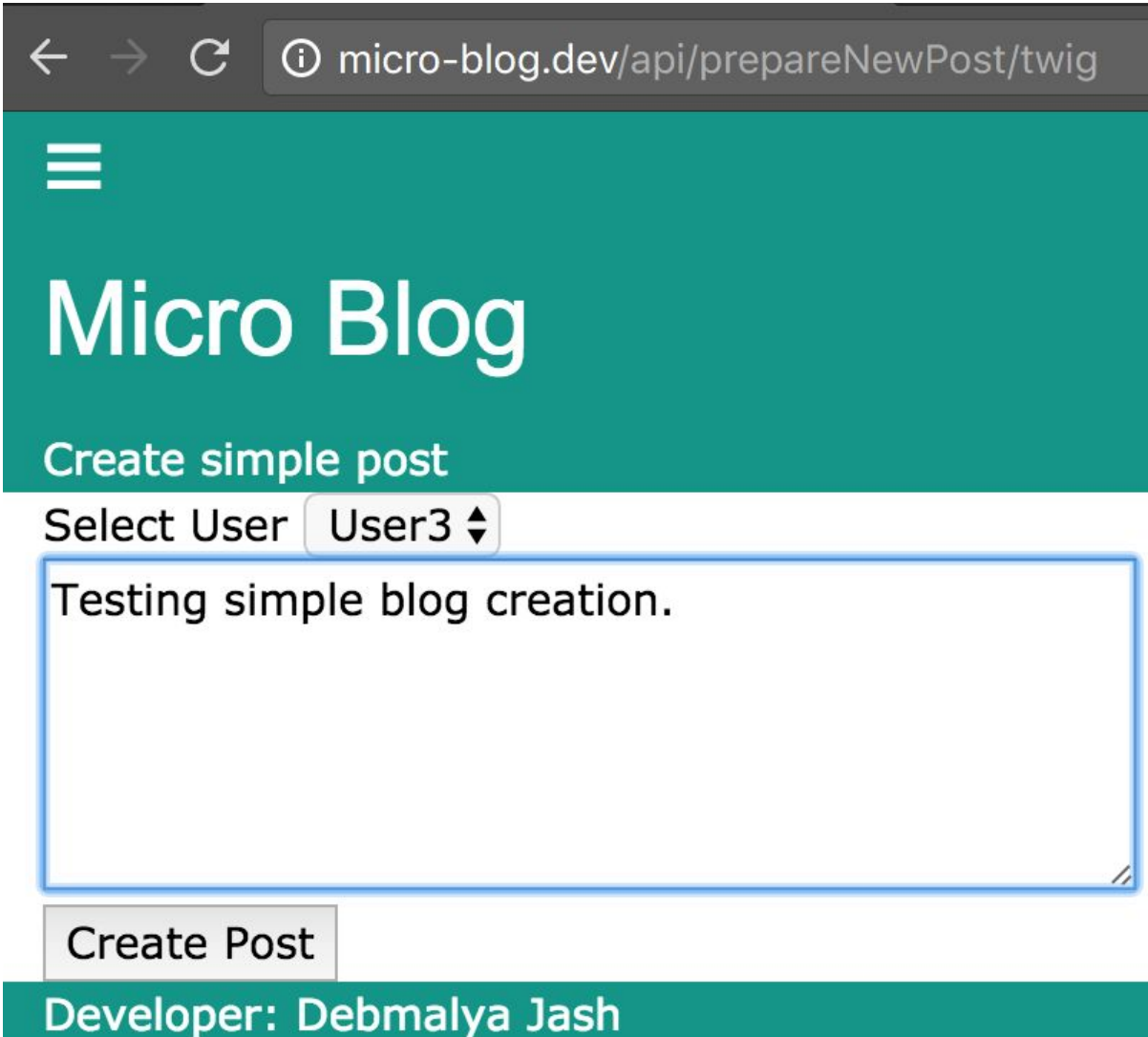
Functionality get list of users



Each user will be a link. On clicking on that user it will display the list of blogs posted by that user.

Developer : Debmalya Jash

Create a simple post



The screenshot shows a web browser window with the address bar displaying `micro-blog.dev/api/prepareNewPost/twig`. The page has a teal header with a hamburger menu icon and the title "Micro Blog". Below the header, the text "Create simple post" is displayed. A "Select User" dropdown menu is set to "User3". A large text input field contains the text "Testing simple blog creation.". Below the input field is a "Create Post" button. The footer of the page, also in teal, displays "Developer: Debmalya Jash".

← → ↻ ⓘ micro-blog.dev/api/prepareNewPost/twig

☰

Micro Blog

Create simple post

Select User User3 ▾

Testing simple blog creation.

Create Post

Developer: Debmalya Jash

Developer : Debmalya Jash



Micro Blog

Blog created successfully.

Welcome !! Click on menu to do different activities.

Developer: Debmalya Jash

View individual post



I have a drop down of rowid, on selecting a rowid that post will appear in the text box. Code for this is

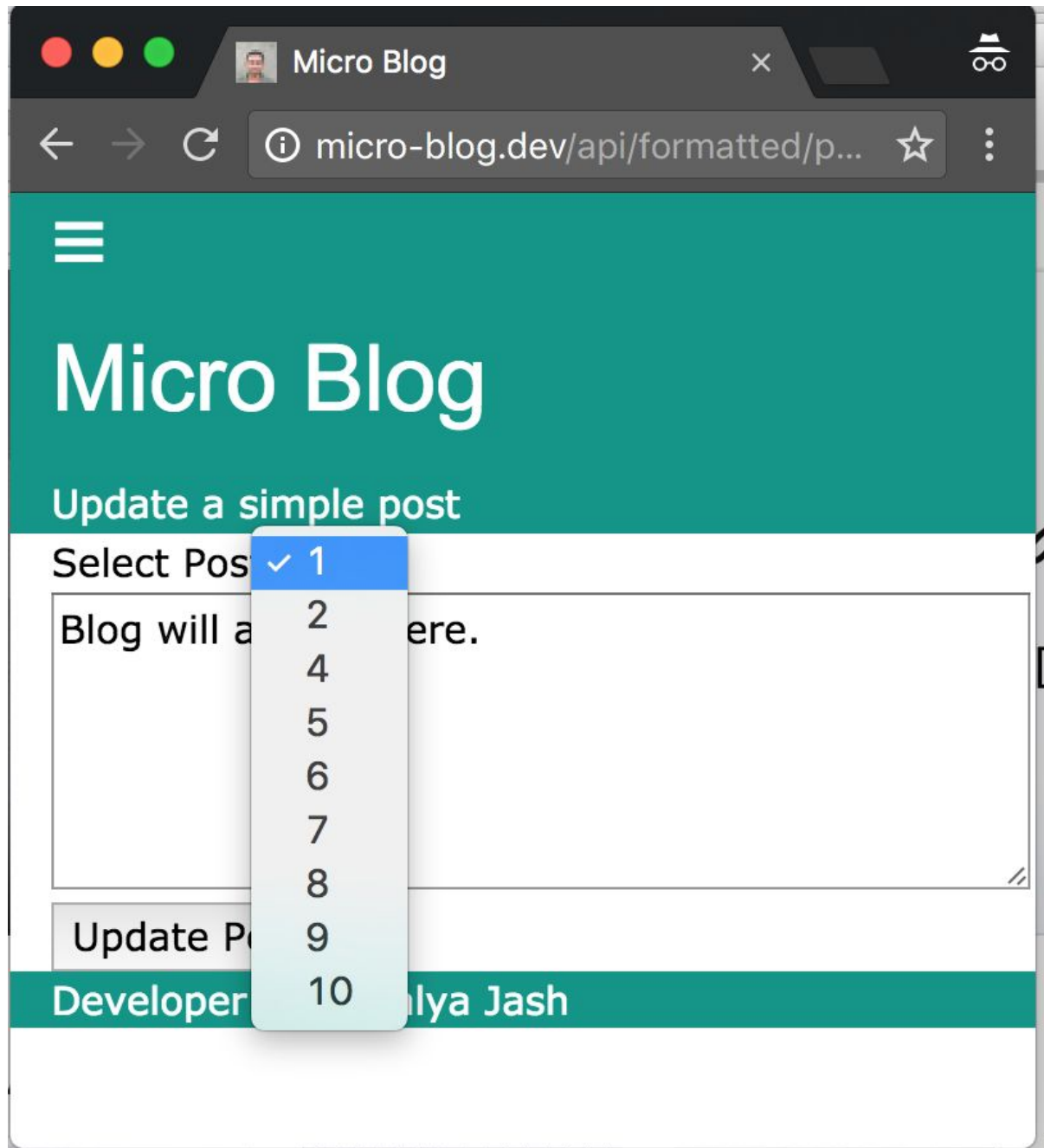
```
$("#post_id").change(function () {
    $.ajax({
        url: "/api/posts/id/" + $(this).val()
    }).then(function (data) {
        $('#content').val(data.content);
    });
});
```

Developer : Debmalya Jash

Update a simple blog

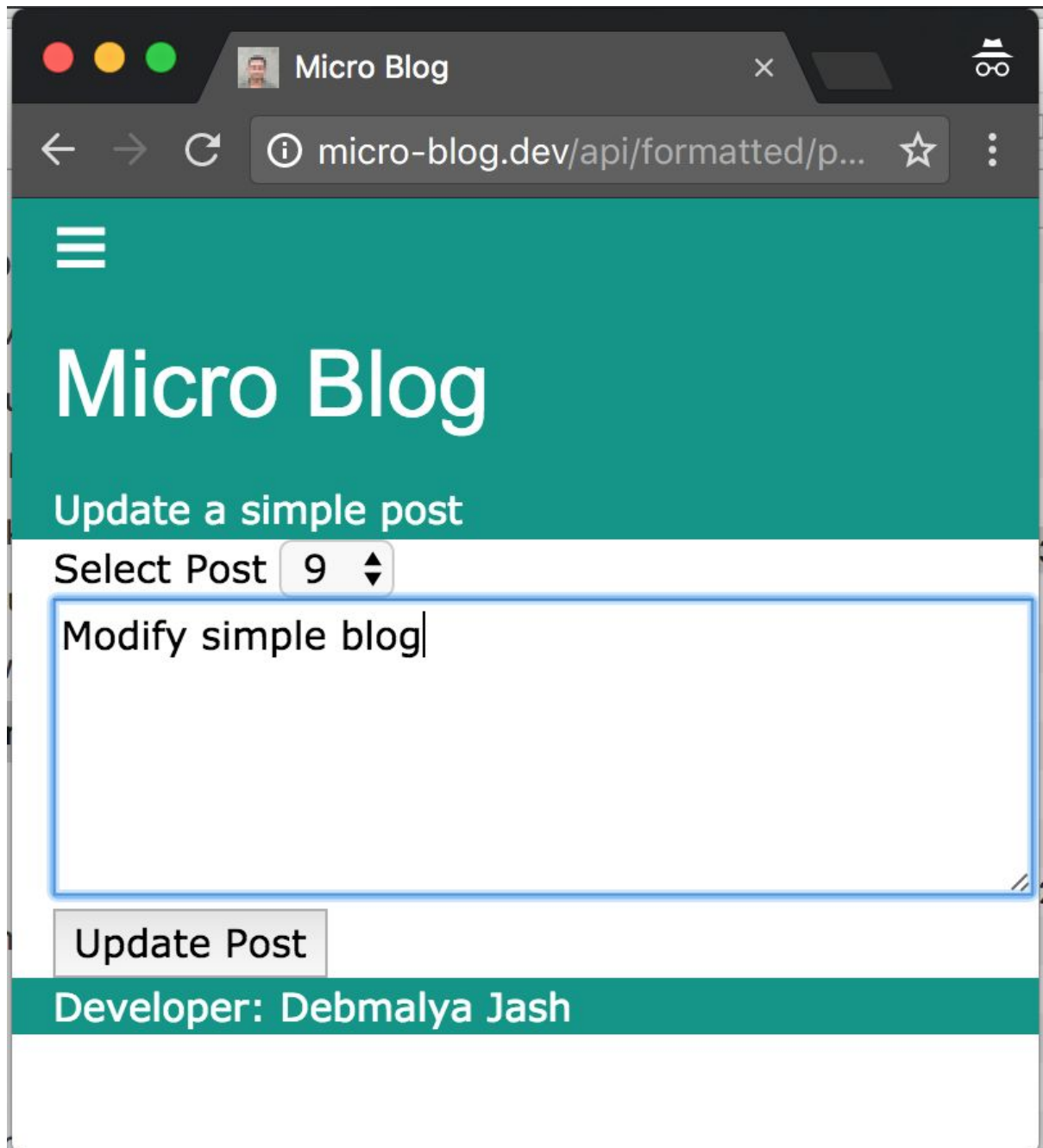
From the menu select 'update a simple blog'.

From the drop down select postid.

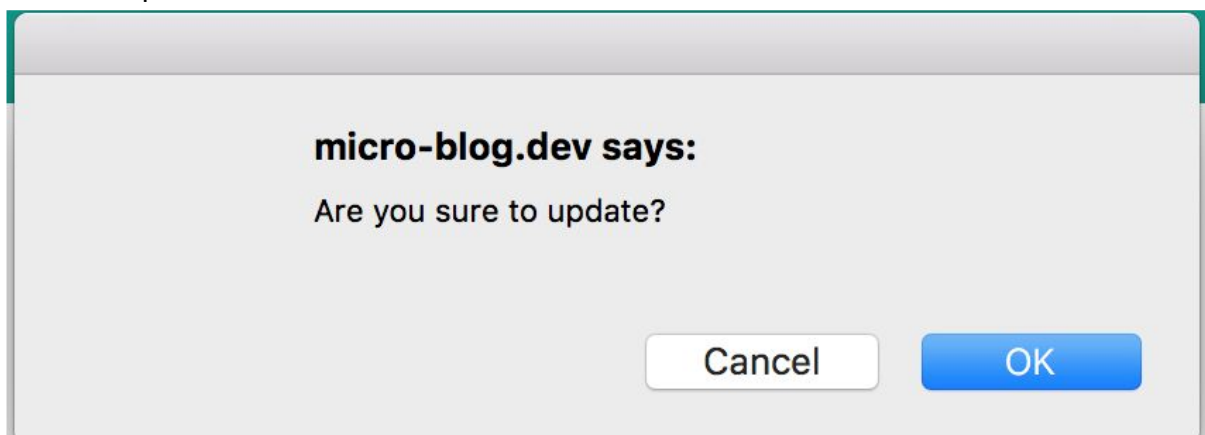


Modify the post.

Developer : Debmalya Jash



Click on 'Update Post'.



Developer : Debmalya Jash

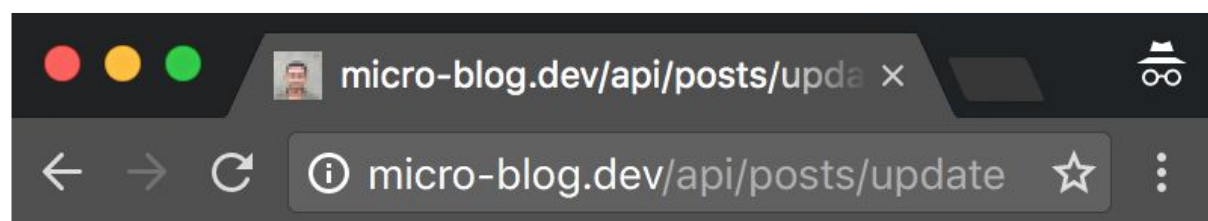
Display of Status Message



Status message "Blog updated successfully".

Display of Error Message

Sample error message screen.

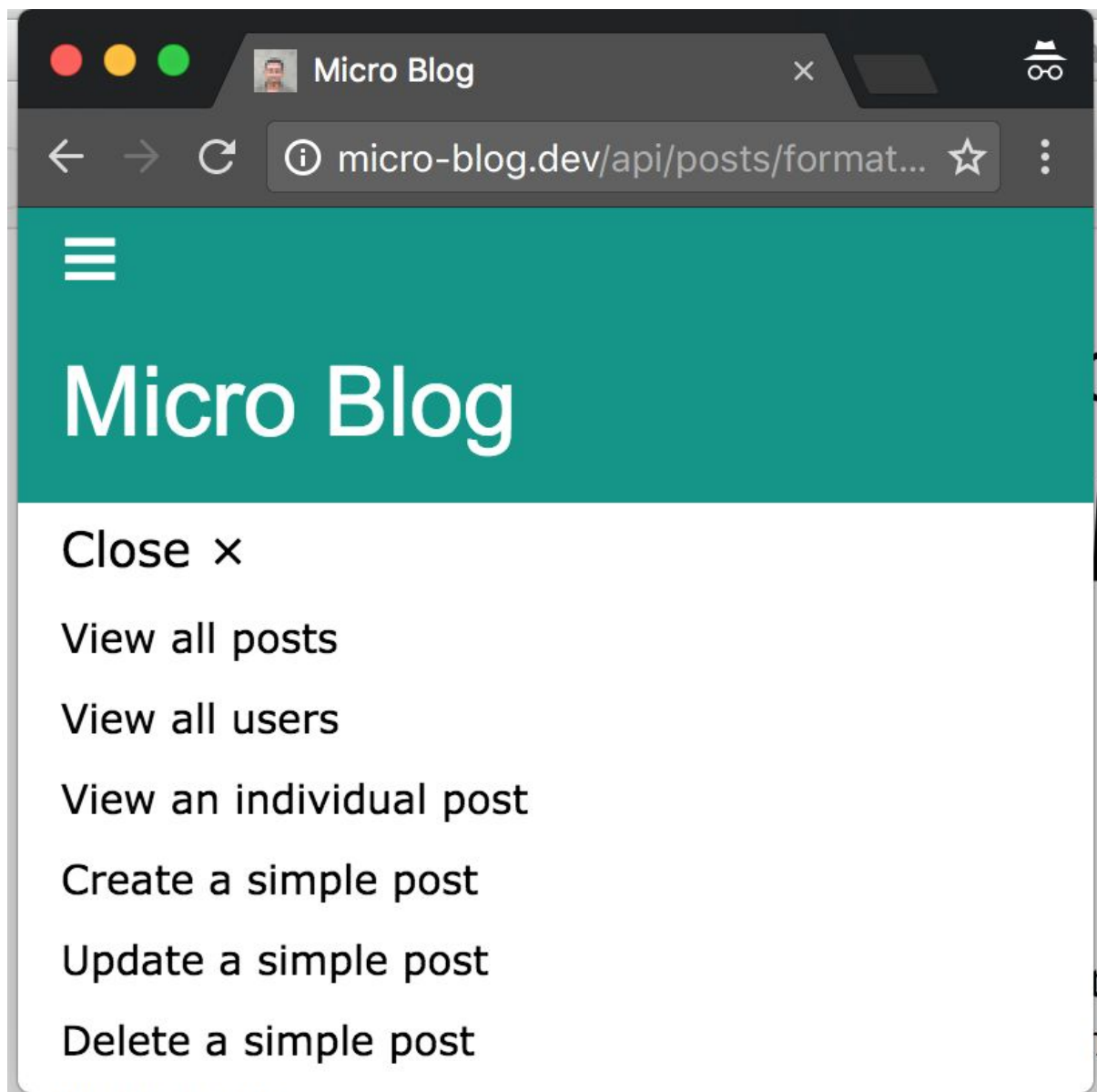


I am sorry, but something went terribly wrong. Error Code :500 Error message :An exception occurred while executing 'UPDATE posts SET content = :content WHERE rowid = :rowid' with params [":content", "hello again", ":rowid", 2]: SQLSTATE[HY000]: General error: 25 bind or column index out of range

I will try to represent the errors in prettier way.

Developer : Debmalya Jash

Menu



List of APIs

API	Status	Description	Remarks
/api/posts	Modified	Existing	Provided checking if there is no post at all.
/api/formattedposts/{format}	New	To get posts in different format.	

Developer : Debmalya Jash

		Currently It supports twig template.	
/api/formattedusers/{format}	New	To get user details in different formats. Currently it supports twig template.	
/api/posts/user/{user_id}	Modified	Existing	Provided checking if there is any post for the user.
/api/posts/formatteduser/{user_id}/{format}	New	To get all the blogs of an user in different formats. Currently it supports twig template.	Each user has hyperlink. On clicking on the hyperlink all the blogs for that user will appear.
/api/posts/id/{post_id}	Modified	Existing	Provided checking if there is any post for the post id.
/api/formattedposts/{format}/{post_id}	New	To get post for the passed id in different format . Currently it supports twig template.	
/api/posts/new	New	To create new post	
/api/formattedposts/new	New	To create new post get specific template, currently supported template is 'twig'.	
/api/prepareNewPost/{format}	New	To prepare template for create simple blog. It will populate the list of users.	
/api/posts/delete	New	To delete a post	
/api/posts/update	New	To update a post	
/api/postid	New	To get all rowids from posts	
/api/postid/{id}	New	To get content of a blog. Id is passed as parameter.	

List of views (twig templates)

Twig template name	Purpose	Remarks
blogs.twig	To display blogs	Try to merge these three into one. Based on some parameter either delete, update link button will appear.
create.post.twig	To create blog	
single_blog.view.twig	To view individual blog	
single_blog.update.twig	To update individual blog	
single_blog.delete.twig	To delete individual blog	
footer.twig	To display footer	
form.html	Trying with twig macro	
header.twig	To display header	
index.twig	To display the landing page	
user.link.twig	To display user. Each user is a link to display all of their blogs.	
users.twig	To display users without any link.	

List of javascripts

Javascript file name	Purpose	Remarks
jquery-3.1.1.min.js	jQuery library	To call web service, handling different event like form confirmation, drop down selection.
functionality1.js	To handle menu opening and closing.	
w3data14.js	W3data javascript library	W3 CSS and W3 data library used for responsive web designing. They are reported to be faster than

Developer : Debmalya Jash

		bootstap.
blog_controller15.js	With the help of jQuery and w3data14.js here I handled web service call, form confirmation, on keyup event user search and blog search.	In future like to make it a full fledged controller. More separation of code from model.

API Web Service Changes

Functionality

Listing all posts

Changes in index.php

```
$app->get('/api/posts', function() use($app) {
    $sql = "SELECT rowid, * FROM posts";
    $posts = $app['db']->fetchAll($sql);

    if (count($posts) == 0) {
        return new Response("There is no post.", 404);
    }

    return $app['twig']->render('blogs.twig', array('posts' => $posts, ));
});
```

Create User table

use Doctrine\DBAL\Schema\Table;

// Create user table

```
$schema = $app['db']->getSchemaManager();
if (!$schema->tablesExist('users')) {
    $users = new Table('users');
    $users->addColumn('user_id', 'integer', array('unsigned' => true,
'autoincrement' => true));
    $users->setPrimaryKey(array('user_id'));
    $users->addColumn('user_name', 'string', array('length' => 32));
    $users->addUniqueIndex(array('user_name'));
    $schema->createTable($users);

    // insert sample rows
    $app['db']->insert('users', array( 'user_name' => 'User1',));
    $app['db']->insert('users', array( 'user_name' => 'User2',));
    $app['db']->insert('users', array( 'user_name' => 'User3',));
```

Developer : Debmalya Jash

```

    $app['db']->insert('users', array( 'user_name' => 'User4',));
}

```

Get list of users (Added functionality)

```

$app->get('/api/users', function() use($app) {
    $sql = "SELECT rowid, * FROM users";
    $posts = $app['db']->fetchAll($sql);

    if (count($posts) == 0) {
        return new Response("There is no users.", 404);
    }

    return $app['twig']->render('users.twig', array('posts' => $posts, ));
});

```

Create a simple blog

```

$app->post('/api/posts/new', function (Request $request) use ($app) {
    $user_id = $request->request->get('user_id');
    $content = $request->request->get('content');

    $app['db']->insert('posts', array( 'content' => $content, 'user_id' =>
(int)$user_id, 'date' => time()));
    $posts = array('message' => 'Blog created successfully. ');
    return $app->json($posts, 200);
});

```

Where is error handling ? - I will try to put this in try catch block.

Where is retry mechanism? If there is database down, or network issue will it retry ?

View individual blog

To view individual blog i rely on following apis

/api/postid

/api/formatted/postid/{format}

Delete a simple blog

```

$app->delete('/api/posts/delete', function (Request $request) use ($app){
    $post_id = $request->request->get('post_id');
    $app['db']->delete('posts', array( 'rowid' => (int)$post_id,));
    $posts = array('message' => 'Blog deleted successfully.', 'rowid'=>$post_id);
    return $app->json($posts, 200);
});

```

Needs more error handling like database down, network issue etc. Have to implement retry mechanism (delayed retry, count retry (e.g. 3 times).

Developer : Debmalya Jash

Update a simple blog

```
$app->put('/api/posts/update', function (Request $request) use ($app){
    $post_id = $request->request->get('post_id');
    $content = $request->request->get('content');
    $sql = "UPDATE posts SET content = :content WHERE rowid = :rowid";
    $app['db']->executeUpdate($sql, array($content, (int)$post_id));
    $posts = array('message' => 'Blog id $post_id updated successfully.');
```

```
    return $app->json($posts, 200);
});
```

Error handling

```
$app->error(function (\Exception $e, $code) {
    return new Response('I am sorry, but something went terribly wrong.' . " Error
Code : " . $code . " Error message : " . $e->getMessage());
});
```

Validation

1. Post id must be a positive integer, ->assert('id', '\d+');
2. Error message for non existing post id.



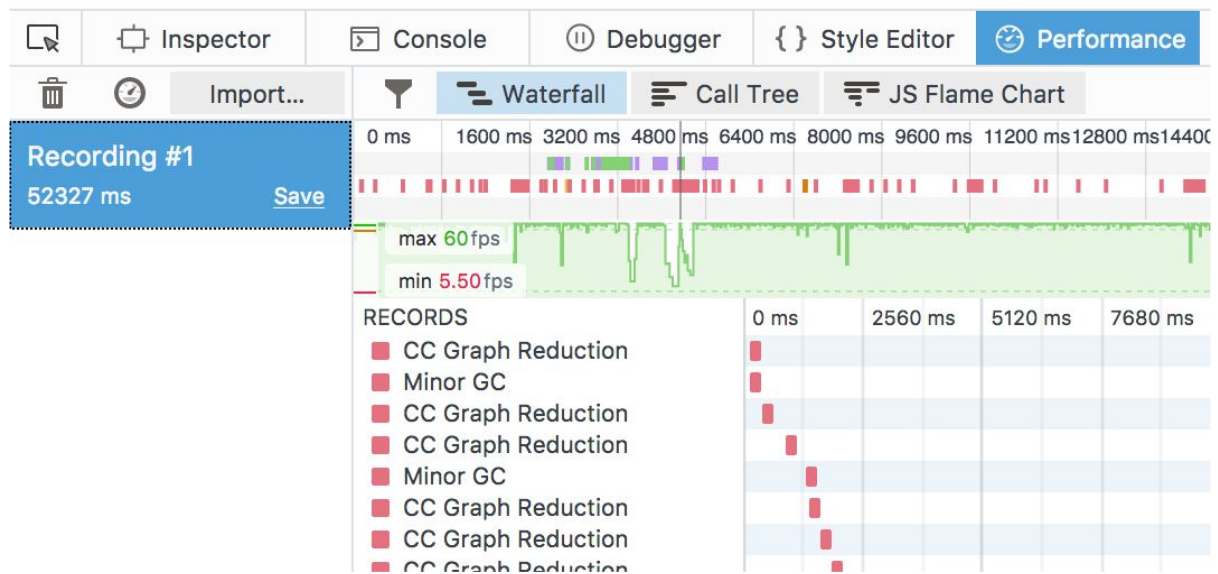
Post id 10 does not exist.

3. Error message for non existing user id.



User id 9 does not exist.

Performance



Will test more.

File List

File Name	Location	Purpose
index.php	src	Our model.
All javascripts files	src/assets/javascripts	Managing the show.
All twig files	src/views	The view of three musketeers MVC.

Before release checklist

Functionality	OK ?
1. View all posts?	
2. View all Users?	
3. After getting all the list of users, click on the user link to check whether all his/her post appeared?	

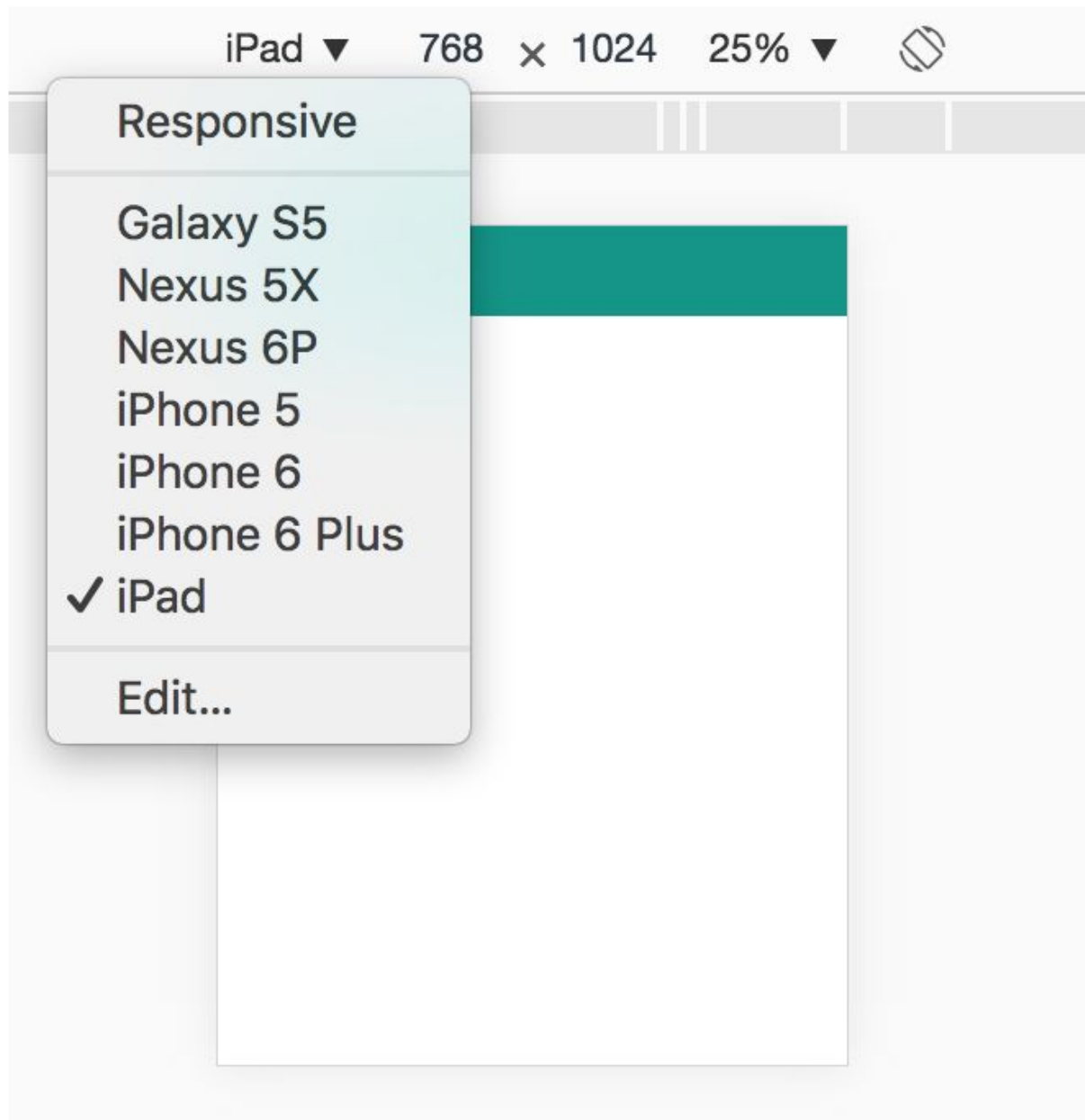
Developer : Debmalya Jash

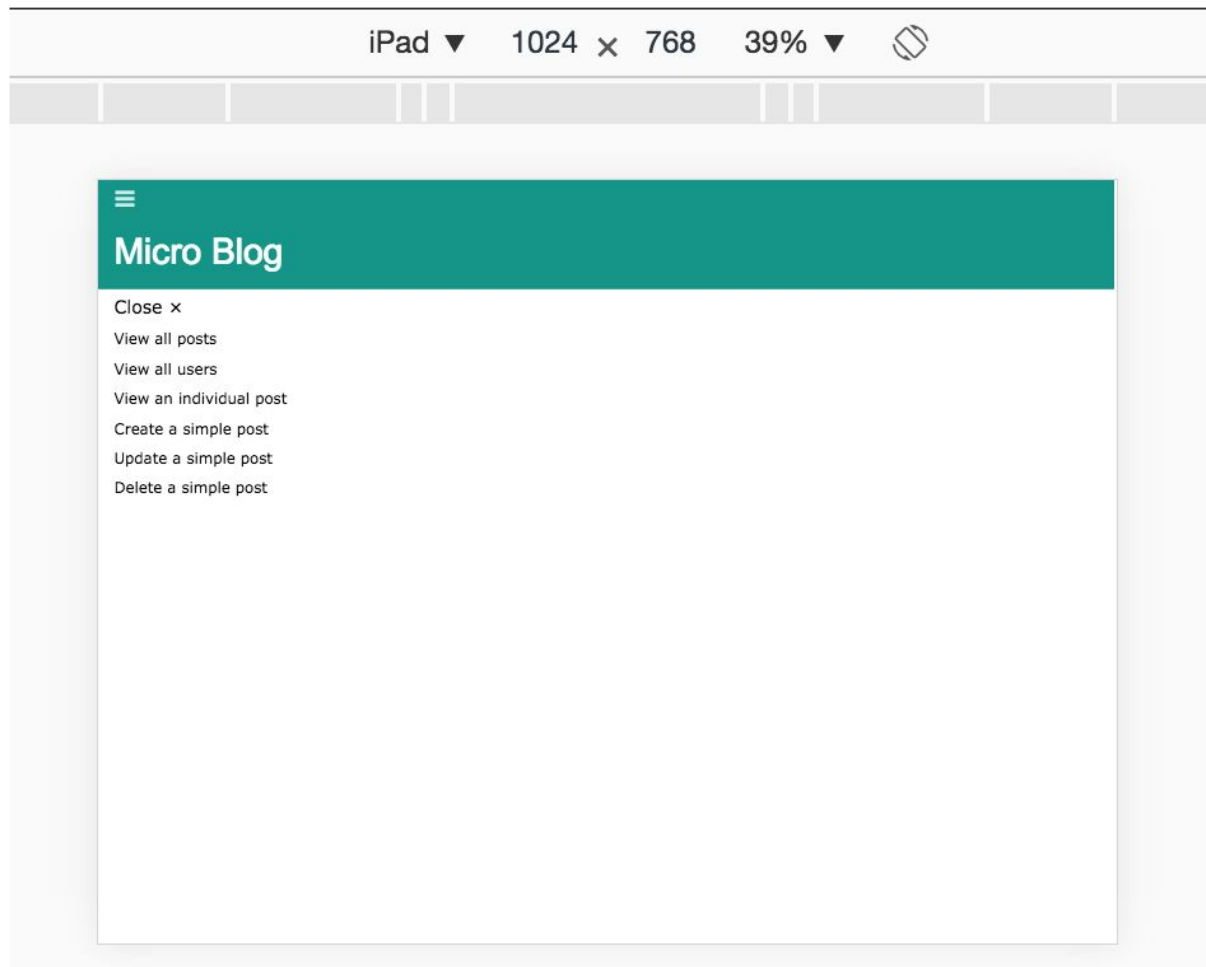
4. Whether view individual post is working or not?	
5. Check whether create a simple post is working or not?	
6. Check whether update a simple post is working or not?	
7. Check whether delete a simple post is working or not?	

Check these functionalities in the following browser

Browser Name	OK?
IE	
Google Chrome	
Safari	
Firefox	

Check these functionalities in different screen resolution





Improvement

Lazy fetching and pagination.

Reduce communication with server. Communication with server should be as minimum as possible.

Service /Template Controller

Create a template controller or template engine, so that template provider (e.g. twig) can be changed any time. Template provider change will be just a matter of configuration and addition /modification of template files.

Reduce database communication

Use [HttpCacheServiceProvider](#) to reduce database hit.

Developer : Debmalya Jash

Add test code

For this I can use [phpunit](#) or Web test case.

Twig macro

I can improve twig files by create a macro for the basic one, then extending on that.

Prevent SQL injection

Have to use `htmlspecialchars`

UTF-8 character handling

This one is not tested.