# BasicController Documentation

## Overview

The BasicController is responsible for managing fundamental website settings and configurations in a Laravel application. It handles essential features like favicon, logo, basic information, maintenance mode, SEO settings, and various other core website functionalities.

## Class Dependencies

```php

use Artisan;

use Session;

use Purifier;

use Validator;

use App\Models\Seo;

use App\Models\Language;

use App\Models\Timezone;

use App\Models\BasicSetting;

use App\Models\BasicExtended;

```

## Constructor

```php

public function \_\_construct()

{

$abs = BasicSetting::first();

Config::set('app.timezone', $abs->timezone);

}

```

- Sets the application timezone based on the basic settings when controller is instantiated

## Key Features and Methods

### 1. Media Management

#### Favicon Management

- `favicon()`: Displays favicon management view

- `updatefav(Request $request)`:

- Handles favicon image upload

- Validates file type (jpg, png, jpeg, ico)

- Generates unique filename

- Updates all basic settings records with new favicon

#### Logo Management

- `logo()`: Displays logo management view

- `updatelogo(Request $request)`:

- Handles logo image upload

- Validates file type (jpg, png, jpeg)

- Updates all basic settings with new logo

#### Preloader Management

- `preloader()`: Displays preloader settings view

- `updatepreloader(Request $request)`:

- Handles preloader image upload

- Updates preloader status and image

- Supports gif, png, jpg, jpeg formats

### 2. Basic Information Management

- `basicinfo()`:

- Displays basic website information settings

- Loads timezone data

- Retrieves basic and extended settings

- `updatebasicinfo(Request $request)`:

- Updates core website settings:

- Website title

- Timezone

- Email verification status

- Base color

- Currency settings (symbol, position, text, rate)

- Updates environment variables

- Clears configuration cache

### 3. Plugin and Script Management

- `script()`: Displays script management view

- `updatescript(Request $request)`:

- Updates various third-party integrations:

- Tawk.to chat

- Disqus comments

- Google reCAPTCHA

- WhatsApp integration

### 4. Maintenance Mode

- `maintainance()`: Displays maintenance mode settings

- `updatemaintainance(Request $request)`:

- Toggles maintenance mode

- Updates maintenance page content

- Handles secret path for maintenance bypass

- Manages maintenance image

### 5. Custom CSS/JS Management

- `css()` & `updateCss(Request $request)`: Manages custom CSS

- `js()` & `updateJs(Request $request)`: Manages custom JavaScript

### 6. SEO Management

- `seo(Request $request)`:

- Manages SEO settings per language

- Creates/updates SEO information

- `updateSEO(Request $request)`:

- Updates language-specific SEO settings

### 7. Cookie Alert Management

- `cookiealert(Request $request)`:

- Manages cookie consent alert settings

- `updatecookie(Request $request, $langid)`:

- Updates cookie alert text and status

- Supports multiple languages

## Security Features

1. File Upload Validation

- Strict file type checking

- Unique filename generation

- Secure file handling

2. Input Sanitization

- Uses Purifier for HTML content

- Input validation using Laravel's Validator

- Clean method for specific inputs

3. Error Handling

- Session-based flash messages

- Proper error responses

- Validation error handling

## Important Notes

1. Multi-language Support

- Most settings support multiple languages

- Language-specific SEO settings

- Localized cookie alerts

2. Performance Considerations

- Configuration cache clearing when needed

- Proper file cleanup for replaced images

- Efficient database queries

3. Maintenance Mode Features

- Secret path support for bypass

- Custom maintenance page

- Configurable maintenance message

## Usage Examples

### Updating Basic Information

```php

$request->validate([

'website\_title' => 'required',

'timezone' => 'required',

'base\_color' => 'required',

// ... other validations

]);

$bss = BasicSetting::all();

foreach ($bss as $key => $bs) {

$bs->website\_title = $request->website\_title;

$bs->timezone = $request->timezone;

// ... other updates

$bs->save();

}

```

### Managing Maintenance Mode

```php

if ($request->maintenance\_status == 1) {

@unlink('core/storage/framework/down');

Artisan::call($down);

} else {

Artisan::call('up');

}

```

## Best Practices

1. Always validate file uploads

2. Use proper error handling

3. Implement proper security measures

4. Clean up old files when replacing

5. Use transaction when updating multiple records

6. Implement proper caching strategies

7. Follow Laravel conventions

## Related Models

- BasicSetting: Core website settings

- BasicExtended: Extended website settings

- Language: Language management

- Timezone: Timezone configurations

- Seo: SEO settings