

Viele Webseiten bieten die Möglichkeit gewisse Themen - Blogs¹ - zu kommentieren. Blogs bestehen aus einem Titel und dem entsprechenden Text. Blogs sollen nur von registrierten Benutzern erfasst und Blogs und Kommentare nur von registrierten Benutzern bearbeitet respektive gelöscht werden können.

Mit

```
php artisan make:model -mrc Blog
```

wird ein Blog-Modell, ein BlogController und eine Migrationsdatei erzeugt. Mit

```
\database\Migrations\*create_blog_table
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateBlogsTable extends Migration {
    public function up() {
        Schema::create('blog', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('user_id')->default(0);
            $table->string('title');
            $table->text('body');
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('blog');
    }
}
```

```
php artisan migrate
```

wird die Tabelle 'blog' erstellt. **user_id** wird benötigt, um den Urheber des Blogs zu definieren.

Blogs können frei kommentiert werden. Mit

```
php artisan make:model -mrc BlogComment
```

wird ein BlogComment-Modell, ein BlogCommentController und eine Migrations-Datei erzeugt. Mit

```
\database\Migrations\*create_blog_comments_table.php
<?php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
class CreateCommentsTable extends Migration {
    public function up() {
        Schema::create('blogComments', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email');
            $table->integer('blog_id')->default(0);
            $table->string('body');
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('blogComments');
    }
}
```

```
php artisan migrate
```

wird die Tabelle 'blogComments' erstellt.

¹ Ein Blog ist ein auf einer Website geführtes und damit meist öffentlich einsehbares Tagebuch oder Journal, in dem mindestens eine Person, der Blogger, international auch Weblogger genannt, Aufzeichnungen führt, Sachverhalte protokolliert („postet“) oder Gedanken niederschreibt. Häufig ist ein Blog eine chronologisch sortierte Liste von Einträgen. Der Blogger ist Hauptverfasser des Inhalts, häufig sind die Beiträge aus der Ich-Perspektive geschrieben. Ein Blog bildet ein Medium zur Darstellung von Aspekten des eigenen Lebens und von Meinungen zu spezifischen Themen, je nach Professionalität bis in die Nähe einer Internet-Zeitung mit besonderem Gewicht auf Kommentaren. Oft sind auch Kommentare oder Diskussionen der Leser über einen Artikel möglich.

Mit **user_id** wird der Autor des Blogs festgehalten.

blog_id dient der Verknüpfung zwischen einem Blog und seinen Kommentaren.

Die Modelle müssen wie folgt angepasst werden.

```
\app\Blog.php
<?php namespace MyWeb;
use MyWeb\BlogComment;
use MyWeb\User;
use Illuminate\Database\Eloquent\Model;
class Blog extends Model {
    protected $fillable = ['body', 'user_id', 'title'];
    public function comment() {
        return $this->hasMany(BlogComment::class);
    }
    public function user() {
        return $this->belongsTo(User::class);
    }
}

\app\BlogComment.php
<?php namespace MyWeb;
use MyWeb\Blog;
use Illuminate\Database\Eloquent\Model;
class Comment extends Model {
    protected $fillable = ['name', 'email', 'blog_id', 'body'];
    public function blog() {
        return $this->belongsTo(Blog::class);
    }
}

\app\User.php
<?php namespace MyWeb;
use MyWeb\Blog;
class User extends Authenticatable {
    protected $hidden = ['password', 'remember_token'];
    public function blog() {
        return $this->hasMany(Blog::class);
    }
}
```

Aus Sicherheitsgründen ist in Laravel die Übernahme von Daten in einem Array gesperrt. Eine Mehrfachzuweisung kann im Datenmodell mit 'protected \$fillable' erlaubt werden.

hasMany legt fest, dass ein User beliebig viele Blogs oder ein Blog beliebig viele Kommentare haben kann.

belongsTo legt fest, dass ein Blog von einem User erstellt wurde oder ein Kommentar sich auf einen Blog bezieht.

Mit

```
\routes\web.php
Route::resource('blog', 'BlogController');
Route::post('blog/{blog}/comment', 'BlogCommentController@store')->name('blog.comment');
Route::resource('blog_c', 'BlogCommentController');
```

werden folgende Links definiert

```
Route::post('blog', 'BlogController@store')->name('blog.store');
Route::get('blog', 'BlogController@index')->name('blog.index');
Route::get('blog/create', 'BlogController@create')->name('blog.create');
Route::put('blog/{blog}', 'BlogController@update')->name('blog.update');
Route::delete('blog/{blog}', 'BlogController@destroy')->name('blog.destroy');
Route::get('blog/{blog}', 'BlogController@show')->name('blog.show');
Route::post('blog/{blog}/comment', 'BlogCommentController@store')->name('blog.comment');
Route::get('blog/{blog}/edit', 'BlogController@edit')->name('blog.edit');
Route::get('blog', 'BlogCommentController@index')->name('blog_c.index');
Route::post('blog', 'BlogCommentController@store')->name('blog_c.store');
Route::get('blog/create', 'BlogCommentController@create')->name('blog_c.create');
Route::get('blog/{blog}', 'BlogCommentController@show')->name('blog_c.show');
Route::delete('blog/{blog}', 'BlogCommentController@destroy')->name('blog_c.destroy');
Route::put('blog/{blog}', 'BlogCommentController@update')->name('blog_c.update');
Route::get('blog/{blog}/edit', 'BlogCommentController@edit')->name('blog_c.edit');
```

Es macht wenig Sinn, dass jeder Besucher einen Blog erstellen kann. Mit der Funktion `__construct` wird bei der Verwendung des BlogController überprüft, ob es sich um einen registrierten Benutzer handelt. Trifft dies nicht zu, kann der Benutzer nur die Liste der Blogs oder einzelne Blogs ansehen. Ein Blog kann nur von einem registrierten Benutzer erstellt und bearbeitet werden.

```
\app\Http\Controllers\BlogController
<?php
namespace MyWeb\Http\Controllers;
use MyWeb\Blog;
use Illuminate\Http\Request;
class BlogController extends Controller {
    public function __construct() {
        $this->middleware('auth')->except(['index', 'show']);
    }
}
```

Das gilt auch für Kommentare - ausser speichern können andere Funktionen nur von einer registrierten Person ausgeführt werden.

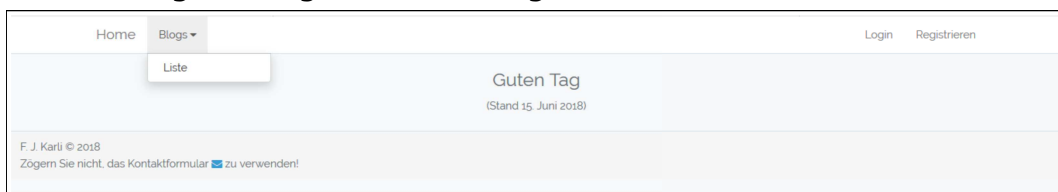
```
\app\Http\BlogCommentController.php
public function __construct() {
    $this->middleware('auth')->except(['store']);
}
```

Mit folgender Anpassung der Menüzeile

```
<!-- Left Side Of Navbar -->
<ul class="nav navbar-nav">
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" data-toggle="dropdown"
      aria-haspopup="true" aria-expanded="false">
      Blogs <span class="fa fa-caret-down"></span>
    </a>
    <ul class="dropdown-menu">
      <li><a href="{{ route('blog.index') }}">Liste</a></li>
      @auth
      <li><a href="{{ route('blog.create') }}">Blog erfassen</a></li>
      @endauth
    </ul>
  </li>
</ul>
```

kann jeder Besucher die Blogs auflisten - jedoch nur registrierte Benutzer Blogs eröffnen.

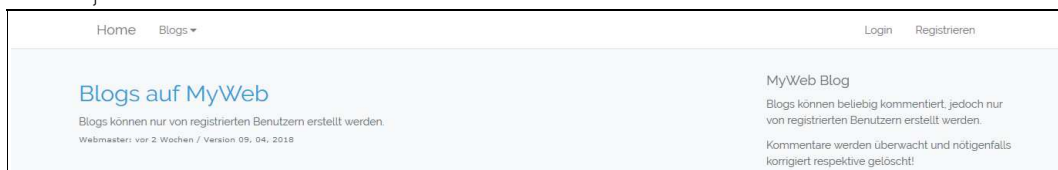
Damit sieht die Begrüssungsseite wie folgt aus:



Ein Klick auf **Liste** bildet die aktuellen Blogs ab.

```
\app\Http\BlogController.php
public function index() {
    $blog = Blog::latest()->get();
    return view('blogs.index', compact('blog'));
}
```

b_index



Ein Klick auf einen Blog-Titel bildet den ausgewählten Blog ab.

Dies ist der geeignete Ort, um Kommentare einzugeben und abzubilden

```
\app\Http\Controllers.php
public function show(Blog $blog) {
    return view('blogs.show', compact('blog'));
}
```

b_show

Mit Klick auf **Kommentar speichern** wird der Kommentar gespeichert.

```
\app\Http\Controllers.php
public function store(Blog $blog) {
    $this->validate(request(), [ 'body' => 'required|min:2',
    'email'=> 'required|email', 'name' => 'required|min:2'
    ]);
    BlogComment::create([
        'body' => request('body'), 'email' => request('email'),
        'name' => request('name'), 'blog_id' => $blog->id
    ]);
    return back();
}
```

Da in den Modellen 'Blog' und 'BlogComment' definiert ist, dass ein Blog mehrfach kommentiert werden kann, ist auch folgendes möglich

```
public function store(Blog $blog) {
    $this->validate(request(), [ 'body' => 'required|min:2',
    'email'=> 'required|email', 'name' => 'required|min:2' ]);
    $blog->addComment(request('body'), request('email'), request('name'));
    return back();
}
```

Dazu muss im Modell 'Blog' die Funktion 'addComment' eingefügt werden.

```
public function addComment($body, $email, $name) {
    BlogComment::create([
        'body' => $body, 'email' => $email,
        'name' => $name, 'blog_id' => $this->id
    ]);
}
```

Da mit der Funktion 'comment' im Modell 'Blog' die Verbindung mit den Kommentaren definiert, ist sind auch folgende Versionen

```
public function addComment($body, $email, $name) {
    $this->comment()->create([ 'body' => $body, 'email' => $email, 'name' => $name ]);
}
```

oder

```
public function addComment($body, $email, $name) {
    $this->comment()->create(compact('body', 'email', 'name'));
}
```

möglich. Nach dem Speichern wird wieder der Blog mit seinen Kommentaren abgebildet.

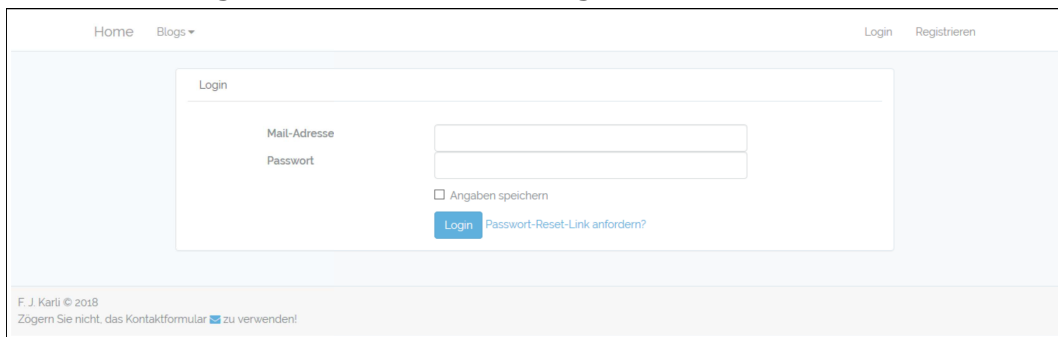
Da Blogs frei kommentiert werden können, gibt es mit grosser Wahrscheinlichkeit Kommentare, welche entfernt oder angepasst werden sollten. Um dies effizient

ausführen zu können, wird der Eintrag eines Kommentars mit einer EMail dem Administrator mitgeteilt.

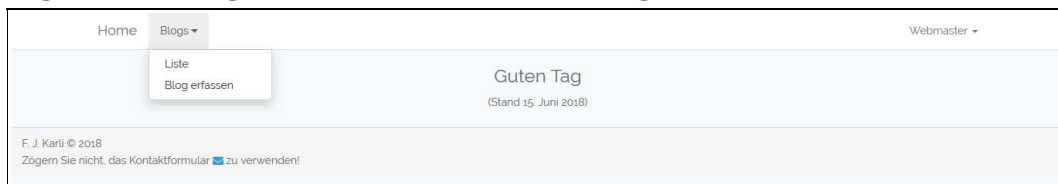
```

app\Http\Controllers\BlogCommentController
public function store(Blog $blog) {
    $this->validate(request(), [
        'body' => 'required|min:2',
        'email'=> 'required|email',
        'name' => 'required|min:2'
    ]);
    $blog->addComment(request('body'), request('email'), request('name'));
    \Mail::send('blogs.c_send', [
        'name' => request('name'),
        'email' => request('email'),
        'comment' => request('body')
    ],
    function($message) {
        $message->from('web@FJKarli.ch');
        $message->to('web@FJKarli.ch', 'Admin')
        ->subject("Blogkommentar für MyWeb");
    });
    return back();
}
    
```

Wie erwähnt bleiben gewisse Funktionen registrierten Benutzern vorbehalten.



Nach erfolgreichen Login wird das Menü für Blogs erweitert.

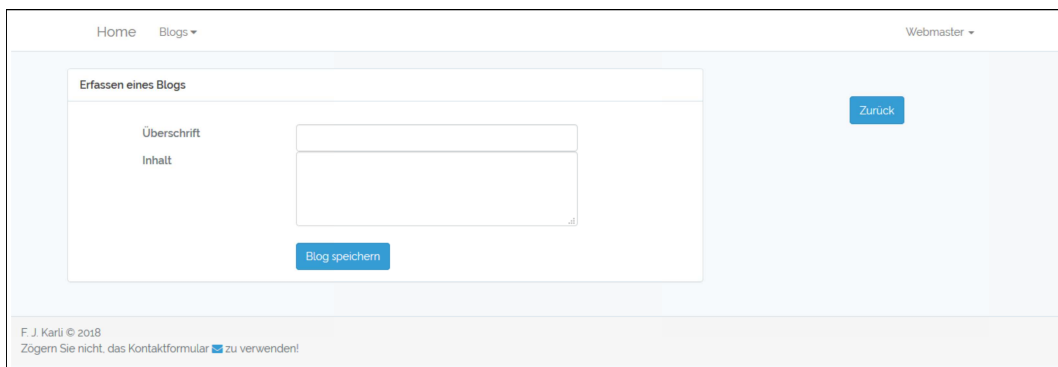


Mit Klick auf **Blog erfassen** können nun Blogs erfasst werden.

```

\app\Http\BlogController.php
public function create() {
    return view('blogs.create');
}
    
```

b_create



Blog speichern führt die Datenbank nach.

```
\app\Http\Controllers.php
public function store(User $user) {
    $this->validate(request(), ['title' => 'required','body' => 'required']);
    Blog::create([
        'title' => request('title'),
        'body' => request('body'),
        'user_id' => auth()->user()->id
    ]);
    return \Redirect::route('blog.index');
}
```

Da in den Modellen 'User' und 'Blog' die Verknüpfung definiert ist, ist auch folgendes möglich

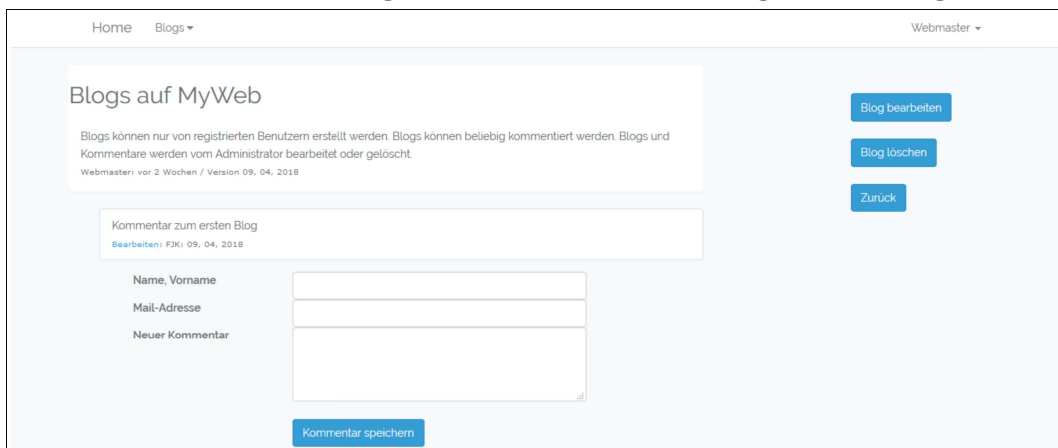
```
public function store(User $user) {
    $this->validate(request(), ['title' => 'required','body' => 'required']);
    Blog::create(request(['title', 'body']));
    return \Redirect::route('blog.index');
}
```

Ein neuer Blog kann auch mit einer Funktion gespeichert werden

```
\app\User.php
public function publish(Blog $blog) { $this->posts()->save($blog); }

\app\Http\Controllers.php
public function store(User $user) {
    $this->validate(request(), ['title' => 'required','body' => 'required']);
    auth()->user()->publish( new Blog(request(['title', 'body'])) );
    return \Redirect::route('blog.index');
}
```

Ein Klick auf den Titel eines Blogs öffnet die Darstellung eines Blogs.



Für registrierte Benutzer werden nun zwei Schaltflächen **Blog bearbeiten** und **Blog löschen** abgebildet. Kommentare erhalten den Präfix **Bearbeiten**.

Blog löschen entfernt den Eintrag.

```
\app\Http\Controllers.php
public function destroy(Blog $blog) {
    Blog::destroy($blog->id);
    return \Redirect::route('blog.index');
}
```

Blog bearbeiten erlaubt den Blog anzupassen

```
\app\Http\Controllers.php
public function edit(Blog $blog) {
    return view('blogs.b_edit', compact('blog'));
}
```

b_edit

Mit **Änderung speichern** wird der überarbeitete Blog gespeichert

```
\app\Http\Controllers\BlogController.php
public function update(Request $request, Blog $blog) {
    $blog::find($blog->id)->update(request(['title', 'body']));
    return \Redirect::route('blog.index');
}
```

Mit **Bearbeiten** kann ein Kommentar angepasst oder gelöscht werden.

```
\app\Http\Controllers\BlogCommentController.php
public function show($id) {
    $blogComment = BlogComment::find($id);
    return view('blogs.c_show', compact('blogComment'));
}
```

c_show

Damit kann ein Kommentar korrigiert oder gelöscht werden.

```
\app\Http\Controllers\BlogCommentController.php
public function edit($id) {
    $blogComment = BlogComment::find($id);
    return view('blogs.c_edit', compact('blogComment'));
}
```

c_edit

Mit **Änderung speichern** wird der Kommentar korrigiert.

```
public function update(Request $request, $id) {
    BlogComment::find($id)->update(request(['name', 'email', 'body']));
    return \Redirect::route('blog.index');
}
```

Mit **Kommentar löschen** wird der Kommentar in der Datenbank gelöscht.

```
public function destroy($id) {
    BlogComment::destroy($id);
    return \Redirect::route('blog.index');
}
```

b_index

\resources\views\blogs\b_index.blade.php

```
@extends('layouts.master')
@section('left')
    <div class="col-md-offset-1 col-md-10">
        @foreach($blogs as $blog)
            <div class="blog-post">
                <h2 class="blog-post-title">
                    <a href="{{ route('blog.show', $blog->id) }}">{{ $blog->title }}</a>
                </h2>
                {{ $blog->body }}
            <note10>
                <br>
                @if ($blog->user_id > 0)
                    {{ $blog->user->name }}:
                @endif
                {{ $blog->created_at->diffForHumans() }} /
                Version {{ $blog->updated_at->format('d, m, Y') }}
            </note10>
        </div>
    @endforeach
</div>
@endsection
@section('side')
    @include('blogs.side')
@endsection
```

\resources\views\blogs\side.blade.php

```
<div class="col-md-offset-1 col-md-10">
    <div class="sidebar-module sidebar-module-inset">
        <h4>{{ config('app.name', 'FJKarli') }} Blog</h4>
        Blogs können beliebig kommentiert, jedoch nur von registrierten
        Benutzern erstellt werden.<br>
        Kommentare werden überwacht und nötigenfalls korrigiert respektive gelöscht!
    </div>
</div>
```

b_show

\resources\views\blogs\b_show.blade.php

```
@extends('layouts.master')
@section('left')
    <div class="panel col-md-offset-1">
        <h2>{{ $blog->title }}</h2>
        <div class="panel-body">
            {{ $blog->body }}
        <note10>
            <br>
            @if ($blog->user_id > 0)
                {{ $blog->user->name }}:
            @endif
            {{ $blog->created_at->diffForHumans() }} /
            Version {{ $blog->updated_at->format('d, m, Y') }}
        </note10>
    </div>
</div>
<div class="col-md-offset-1">
    <!-- Liste der Kommentare -->
    <ul>
        @foreach ($blog->comment as $comment)
            <li class="list-group-item">
                {{ $comment->body }}
                <br>
                <note10>
                    @auth
                        <a href="{{ route('blog_c.show', $comment->id) }}">Bearbeiten: </a>
                    @endauth
                    {{ $comment->name }}:
                    {{ $comment->updated_at->format('d, m, Y') }}
                </note10>
            </li>
        @endforeach
    </ul>
    <!-- Erfassen eines Kommentars -->
    @include('layouts.errors')
    {!! Form::open(['class' => 'form',
        'route' => ['blog.comment', $blog->id]]) !!}
    @include('partials.input_string',
        ['label' => 'Name, Vorname', 'value' => 'name'])
    @include('partials.input_email')
```



```

        @include('partials.input_memo',
            ['label' => 'Neuer Kommentar', 'value' => 'body'])
        @include('layouts.submit',
            ['submitButtonText' => 'Kommentar speichern'])
        {!! Form::close() !!}
    </div>
@endsection
@section('side')
    @auth
        {!! Form::open(['class'=>'form', 'method' => 'GET',
            'route' => ['blog.edit', $blog->id]]) !!}
        @include('layouts.submit', ['submitButtonText' => 'Blog bearbeiten'])
        {!! Form::close() !!}
        {!! Form::open(['class'=>'form', 'method' => 'DELETE',
            'route' => ['blog.destroy', $blog->id]]) !!}
        @include('layouts.submit', ['submitButtonText' => 'Blog löschen'])
        {!! Form::close() !!}
    @endauth
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog.index']]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```

b_create

\resources\views\blogs\b_create.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <strong>Erfassen eines Blogs</strong>
        </div>
        <div class="panel-body">
            @include('layouts.session_message')
            {!! Form::open(['class' => 'form', 'route' => 'blog.store']) !!}
            @include('partials.input_string',
                ['label' => 'Überschrift', 'value' => 'title'])
            @include('partials.input_memo',
                ['label' => 'Inhalt', 'value' => 'body'])
            @include('layouts.submit', ['submitButtonText' => 'Blog speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog.index']]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```

\resources\views\blogs\back.blade.php

```

@section('side')
    <div class="col-md-offset-2">
        <a href="{ route('blog.index') }">Zurück</a>
    </div>
@endsection

```

b_edit

\resources\views\blogs\b_edit.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <strong>Bearbeiten eines Blogs</strong>
        </div>
        <div class="panel-body">
            @include('layouts.session_message')
            {!! Form::model($blog, ['class'=>'form', 'method' => 'PUT',
                'route'=> ['blog.update', $blog->id] ]) !!}
            @include('partials.input_string',
                ['label' => 'Überschrift', 'value' => 'title'])
            @include('partials.input_memo', ['label' => 'Inhalt', 'value' => 'body'])
            @include('layouts.submit', ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')

```

```

    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog.index']] !!)
        @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
S@endsection

```

c_show

\resources\views\blogs\c_show.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="col-md-offset-1 col-md-11">
        <div class="blog-post">
            <h2 class="blog-post-title">{{ $blogComment->body }}</h2>
            <note10>
                <br>
                {{ $blogComment->name }}:
                {{ $blogComment->created_at->diffForHumans() }} /
                Verion {{ $blogComment->updated_at->format('d, m, Y') }}
            </note10>
        </div>
    </div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog_c.edit', $blogComment->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Kommentar bearbeiten'])
    {!! Form::close() !!}
    {!! Form::open(['class'=>'form', 'method' => 'DELETE',
        'route' => ['blog_c.destroy', $blogComment->id]]) !!}
    @include('layouts.submit', ['submitButtonText' => 'Kommentar löschen'])
    {!! Form::close() !!}
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog.index']] !!)
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```

c_edit

\resources\views\blogs\c_edit.blade.php

```

@extends('layouts.master')
@section('left')
    <div class="panel panel-default col-md-offset-1">
        <div class="panel-heading">
            <strong>Bearbeiten eines Blogkommentars</strong>
        </div>
        <div class="panel-body">
            @include('layouts.session_message')
            {!! Form::model($blogComment, ['class'=>'form', 'method' => 'PUT',
                'route'=> ['blog_c.update', $blogComment->id] ]) !!}
            @include('partials.input_string',
                ['label' => 'Name, Vorname', 'value' => 'name'])
            @include('partials.input_email')
            @include('partials.input_memo',
                ['label' => 'Kommentar', 'value' => 'body'])
            @include('layouts.submit', ['submitButtonText' => 'Änderungen speichern'])
            {!! Form::close() !!}
        </div>
    </div>
@endsection
@section('side')
    {!! Form::open(['class'=>'form', 'method' => 'GET',
        'route' => ['blog.index']] !!)
    @include('layouts.submit', ['submitButtonText' => 'Zurück'])
    {!! Form::close() !!}
@endsection

```