

Web04 Ue01

Valentin Rothensteiner

20. Mai 2018

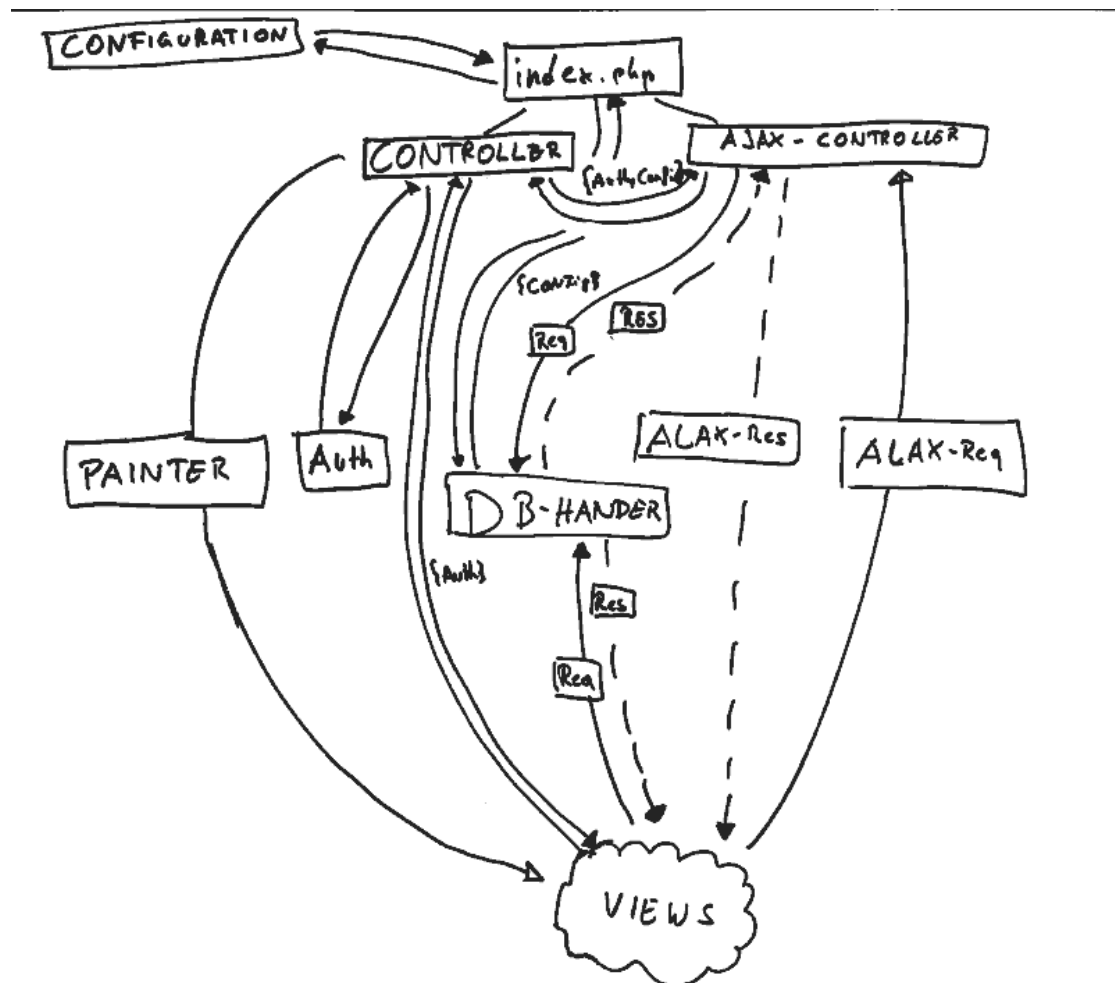
Aufwand: 15-20h

1 Web-Blog

1.1 Lösungsidee

1.1.1 Code

Um den Code besser beschreiben zu können soll vorher eine Übersicht gegeben werden:



Das Zentrale Objekt - bessergesagt das einzige File das direkt im Browser geöffnet werden soll - *index.php* ist daher auch das wichtigste File in dem Konstrukt. Es bindet 2 Files direkt ein: *configuration.php* und *bootstrap.php*. *configuration.php* beinhaltet die Konfigura-

tionsvariablen, *bootstrap.php* beinhaltet php Einstellungen und lädt die Library-Klassen. Nachdem diese zwei Dateien geladen sind, wird geschaut ob einer der zwei Controller eine Aktion übergeben werden soll. Wenn nicht, wird versucht über den normalen Controller eine Seite zu laden.

Da in der Übung der Controller Aufgrund von Komplexität kaum - eigentlich schon fast zu wenig - verwendet wurde versucht, zumindest einen Schritt weiterzugehen und auf der obersten Ebene einen View-Painter zu implementieren. Dieser lädt die gerade benötigten Views. Der HTTP-Get-Parameter *view* wurde somit als Aktion für den Controller festgelegt.

Ein weiterer Grund warum dies verändert wurde ist, da der Code aus der Übung eigentlich eine PHP-Include-Injection enthielt, sodass jedes PHP-File auf das der www-user Leserecht hat betrachtet werden konnte. Dies wurde so gelöst, dass Sonderzeichen nicht mehr erlaubt wurden.

Zu dem Hauptkontroller der nur Views und login verwaltet wurde ein zweiter Controller *AJAXController.php* eingeführt. Dieser verwaltet Aktionen die über AJAX abgesetzt wurden. Theoretisch könnte dies über den Hauptkontroller abgefertigt werden, jedoch gibt dies eine bessere Struktur.

Der DB-Handler fertigt alle Datenbank-Abfragen ab und bezieht Voreinstellungen aus *Configuration.php*. Dabei wird PDO verwendet um SQL-Injection zu verhindern und die php-funktion `htmlspecialchars` um xss zu vermeiden.

Für die Ajax-Calls wurde eine loader-Funktion in `assets/lib.js` erstellt. diese verpackt die zu verschickenden daten als encodiertes JSON und schickt diese an `index.php` mit der Aktion `ajax`. So 'sieht' der *ajax-Controller* die Anfrage und kann sie Verarbeiten.

Die jeweiligen javascript-Files, die eigentlich js/php Mischformen sind, wurden in `views/js/` unter dem Viewnamen gespeichert.

Für das erfassen eines Neuen Artikels wurde nur für das UI ein fertiger Editor aufgrund von Zeitersparnis LineControl developed by [Suyati Technologies] verwendet. Dieser wurde für Das verwendete Bootstrap 4.x angepasst und Funktionen leicht abgeändert. Dies erlaubt es Formatierung in auf die Seite zu bringen. Da das Script jedoch html-tags verwendet wurde beim DataManager eine eigene Filter-methode für diesen String-input entwickelt. Das Java-Script Editor-'Programm' an sich Escaped schon html-Tags. Wird dies jedoch umgangen werden Keywords, welche auf eine Blacklist stehen aus dem String gelöscht. Die Blacklist beinhaltet folgende Wörter: `'script'`, `'iframe'`, `'button'`, `'input'`, `'textbox'`, `'link'`, `'src'`, `'href'`, `'javascript'`, `'name'`, `'alert'`, `'onmouseover'`, `'onload'`, `'expression'`, `'blockquote'`. Damit sollte im Großen und Ganzen XSS verhindert werden. Dies ist natürlich keine optimale Lösung. eine bessere wäre es, nur Elemente aus eine white-List zuzulassen und alle anderen nicht zu löschen sondern zu escapen. Da jedoch nur wenige Minuten für diese Aufgabe zur Verfügung standen wurde es auf diese Weise gelöst. Im Echtfall sollte man sowieso auf fertige Lösungen(Framework/Lib) zurückgreifen, dies war bei der Übung jedoch leider nicht erlaubt.

1.1.2 Datenbank

In der Datenbank Sollen User abgespeichert werden. Diese haben eine von 2 Rollen. Rollen können *Editor* oder *User* sein. User sollen einen eindeutigen Usernamen haben und ein Passwort.

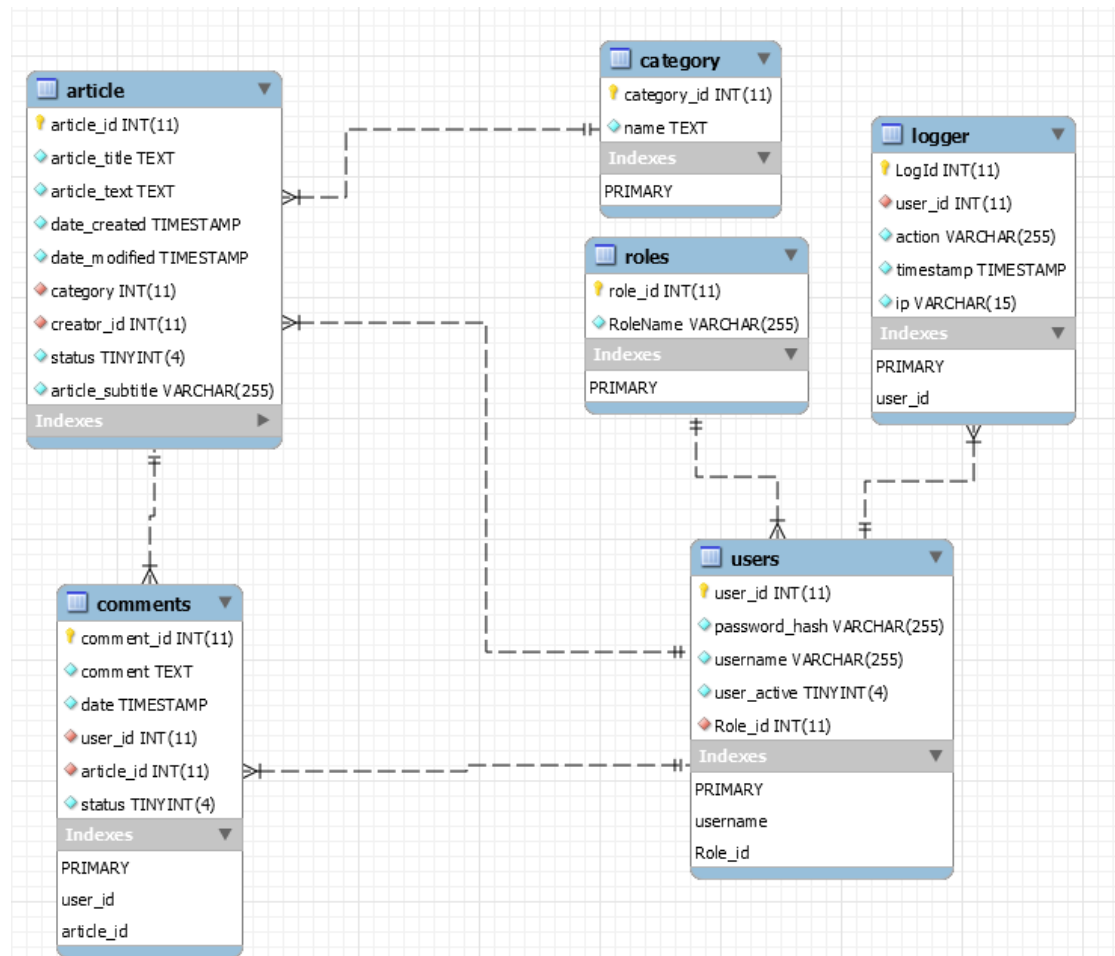
Es soll Artikel geben, welche einen Titel, Untertitel, Text, eine Erstellungs- und ein Modifikationsdatum, eine Kategorie, einen Lösch-Status speichern sollen. Artikel können von *Editor-Usern* erstellt werden.

Eine Kategorie kann von mehreren Artikeln verwendet werden, ein Artikel hat eine Kategorie.

Alle angemeldeten User können einen Kommentar einem gewissen Artikel erstellen, diesen editieren und dessen Status auf gelöscht stellen. Dieser Kommentar beinhaltet den Kommentar, ein Datum, ob er gelöscht ist, wer ihn verfasst hat und unter welchem Artikel er steht.

Um Aktionen Loggen zu können, wird eine Tabelle Logger benötigt. Diese speichert eine User-Id. Den Aktions-Typ, einen Zeit-Stempel und eine IP-Adresse.

Wird dies nun weiter abstrahiert kommt man auf folgendes Modell:



1.2 Testfälle

1.2.1 Benutzerregistrierung

Nur Editoren Können neue Benutzer registrieren.

Vorher:

Register a new User

Username

Enter your preferred Username

Password

Enter your preferred Password

User Role

Select your User Role

user_id	password_hash	username	user_active	Role_id
29	68be59da0cf353ae74ee	admin	1	1

Nachher:

Register a new User

Username

Enter your preferred Username

Password

Enter your preferred Password

User Role

Select your User Role

user_id	password_hash	username	user_active	Role_id
29	68be59da0cf353	admin	1	1
32	28e123b2d8041c	test	1	2

erfasst Beitrag

[illegible]

editiert Beitrag

Save

Standard

Title: testtitleEDIT

Subtitle: testsubtitleEDIT

Font

Formatting


Font size


B


I


U


A
















































































testtitleEDIT

testsubtiteLEDITED

Standard

0 Comments 2018-05-19 23:37:42

121212121221211221MY SUPER-ARRTiCle

aadsasd

[illegible]

1. Test
2. 123123123



asdasdasdasdasdasdasdasdasdasdasdasdasd12121212d



New Comment:

Comment

No Comments were found!

löscht Beitrag

ID	Article Header	Category	Created	Last Modified	Status	Edit
41	testtitle EDIT	Standard	2018-05-19 23:34:07	2018-05-19 23:37:42		

ID	Article Header	Category	Created	Last Modified	Status	Edit
41	testtitle EDIT	Standard	2018-05-19 23:34:07	2018-05-19 23:37:42		

erfasst Kommentar

New Comment:

Comment

admin (2018-05-19 23:39:31)

MYComment



editiert Kommentar

admin (2018-05-19 23:39:31)

MYCommentEDITED



admin (2018-05-19 23:39:31)

MYCommentEDITED



löscht Kommentar von sich

114

MYCommentEDITED

2018-05-19 23:39:31



löscht Kommentar von User

New Comment:

Comment

test (2018-05-19 23:42:35)

testComment TESTUSER



1.2.3 Benutzer

erfasst Kommentar

New Comment:

Comment

test (2018-05-19 23:44:49)

testComment



editiert Kommentar

admin (2018-05-19 23:45:39)

ADMINCOMMENT

test (2018-05-19 23:44:49)

testCommentEDIT<script>alert("hello WORLD :D XSS");<script>



löscht Kommentar

116 `testCommentEDIT<script>alert("hello WORLD :D XSS");</script>`

2018-05-19 23:44:49



1.2.4 mehrere Kommentar im Beitrag

test (2018-05-19 23:50:46)

weqweqwe



test (2018-05-19 23:50:43)

123123



test (2018-05-19 23:50:42)

asdfasdfsadf



test (2018-05-19 23:50:41)

asdfasdfsadf



test (2018-05-19 23:50:40)


asdfasdf



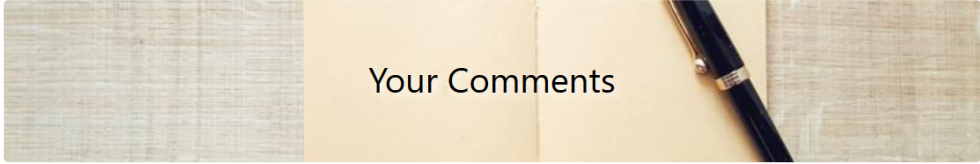
1.2.5 Benutzerübersicht

Editor

[MyCMS](#) [User Overview](#) [Register New User](#) [New Article](#) Logged in as admin

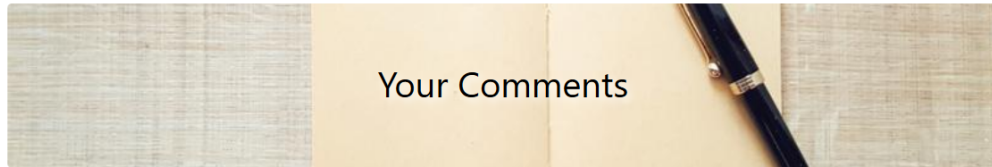


ID	Article Header	Category	Created	Last Modified	Status	Edit
41	testtitelEDIT	Standard	2018-05-19 23:34:07	2018-05-19 23:39:23		
40	asdasd	Standard	2018-05-19 21:46:46	2018-05-19 21:46:46		
39	dasd	Standard	2018-05-19 21:46:08	2018-05-19 21:46:08		
37	asd12312312312	Standard	2018-05-19 21:33:45	2018-05-19 21:34:03		
38	eqwe	Standard	2018-05-19 21:33:55	2018-05-19 21:33:55		
36	asd	Standard	2018-05-19 21:33:33	2018-05-19 21:33:33		
35	asdas	Standard	2018-05-19 20:33:33	2018-05-19 20:33:33		
34	asdasdasd	Standard	2018-05-19 20:33:04	2018-05-19 20:33:04		



ID	Comment	Created	Status
107	asdasdasd	2018-05-19 20:58:48	
108	asdasd	2018-05-19 20:58:50	
109	asdasd12123123	2018-05-19 20:58:51	
113	asdasdasd	2018-05-19 21:33:49	
114	MYCommentEDITED	2018-05-19 23:39:31	
117	ADMINCOMMENT	2018-05-19 23:45:39	

User



ID	Comment	Created	Status
115	testComment TESTUSER	2018-05-19 23:42:35	
116	testCommentEDIT<script>alert("hello WORLD :D XSS");<script>	2018-05-19 23:44:49	
118	asdfasdf	2018-05-19 23:50:40	
119	asdfasdfasdf	2018-05-19 23:50:41	
120	asdfasdfsadf	2018-05-19 23:50:42	
121	123123	2018-05-19 23:50:43	
122	weqweqwe	2018-05-19 23:50:46	

1.2.6 Log

LogId	user_id	action	timestamp	ip
1	29	newArticle	2018-05-19 21:46:08	192.168.10.243
2	29	newArticle	2018-05-19 21:46:46	192.168.10.243
3	29	registerUser	2018-05-19 23:04:44	192.168.10.243
4	29	registerUser	2018-05-19 23:06:05	192.168.10.243
5	29	newArticle	2018-05-19 23:34:07	192.168.10.243
6	29	updateArticle	2018-05-19 23:35:19	192.168.10.243
7	29	updateArticle	2018-05-19 23:37:41	192.168.10.243
8	29	updateArticleStatus	2018-05-19 23:38:43	192.168.10.243
9	29	updateArticleStatus	2018-05-19 23:39:23	192.168.10.243
10	29	createComment	2018-05-19 23:39:31	192.168.10.243
11	29	updateComment	2018-05-19 23:40:50	192.168.10.243
12	29	loadAllComments	2018-05-19 23:40:50	192.168.10.243
13	29	delComment	2018-05-19 23:41:11	192.168.10.243
14	32	createComment	2018-05-19 23:42:35	192.168.10.243
15	29	delComment	2018-05-19 23:43:13	192.168.10.243
16	32	createComment	2018-05-19 23:44:49	192.168.10.243
17	29	createComment	2018-05-19 23:45:39	192.168.10.243
18	32	updateComment	2018-05-19 23:46:49	192.168.10.243