

Ticketsystem - Team K.O.M.A

Programmieren II - Programmentwurf

Prof. Dr. Helmut Neemann



INF16A

Matrikelnummern:

9188103

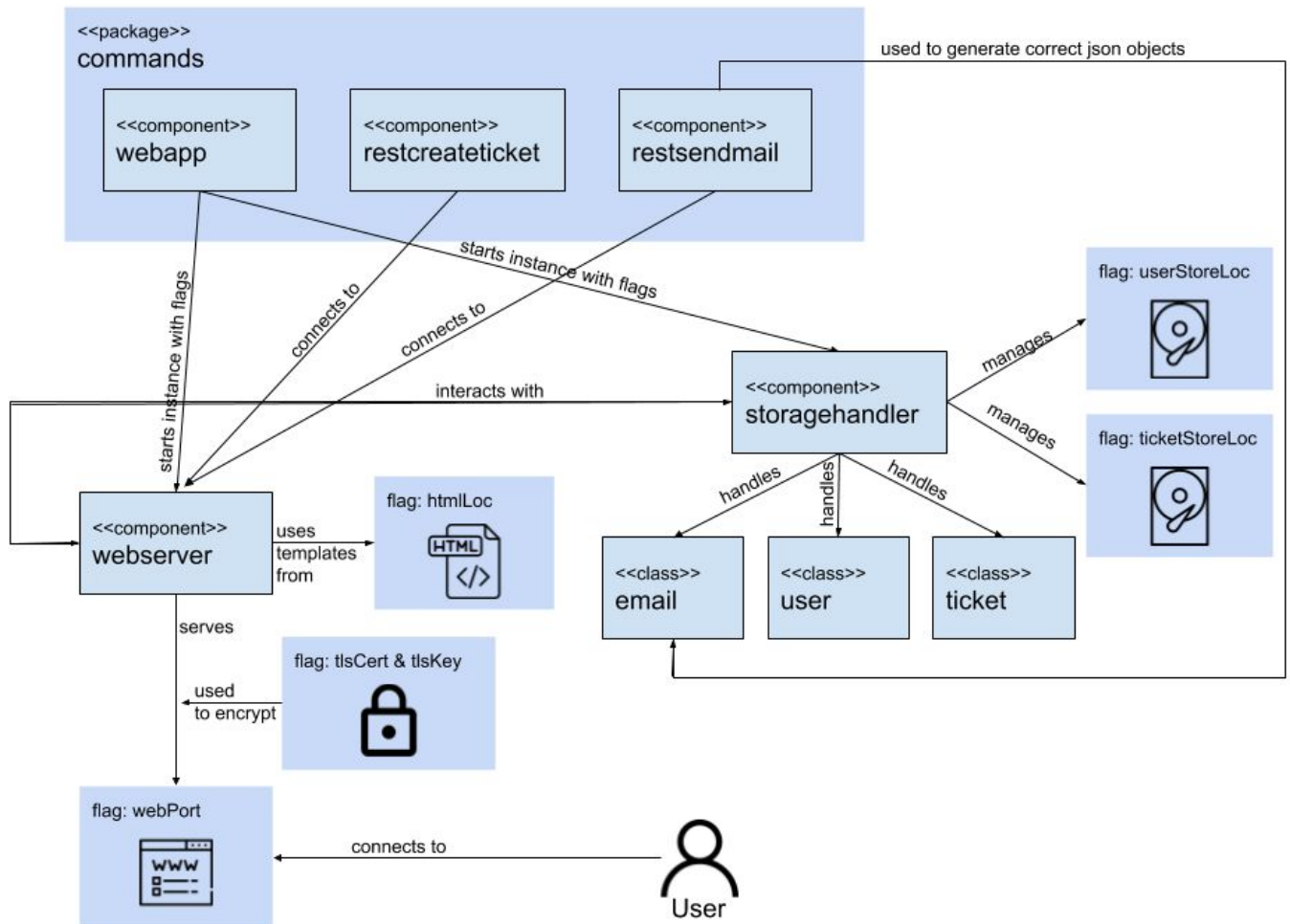
1798794

4717960

Architekturdokumentation	2
Komponentenübersicht	2
Komponentenbeschreibung	2
webserver	3
storagehandler	4
email	4
user	4
ticket	4
commands	5
webapp	5
restcreateticket	5
restsendmail	5
Anwenderdokumentation	7
Ticketerstellung als unauthorisierter Benutzer	7
Administrationsoberfläche	8
Betriebsdokumentation	12
Ticketsystem Webapplikation	12
REST Client Ticketerstellung	12
REST Client E-Mail Verwaltung	12
Aufteilung der Implementierung	13
9188103	13
1798794	13
4717960	13
Konzeption	13
HTML Grunddesign	13
Konfiguration und Startparameter	13
REST Client E-Mail Verwaltung	13
REST Client Ticketerstellung	13
Dokumentation	13

Architekturdokumentation

Komponentenübersicht



Komponentenbeschreibung

Die Applikationslogik ist in drei Punkte gegliedert. An erster Stelle steht der Webserver (das Frontend). Dieser stellt alle visuellen Benutzererfahrungen und Schnittstellen für externe Tools bereit. Als Zweites existiert der storagehandler, die Persistenz der Applikation. Hier wird sichergestellt, dass alle Elemente aus dem Frontend auch auf dem Speicher gesichert werden. Weiterhin sind hier die Models, oder auch Klassen verankert. Im Ticketsystem existieren drei verschiedene Klassen:

- email: Objekte, welche als Mail versendet werden sollen
- user: Benutzerobjekte, die zur Authentifizierung als Bearbeiter dienen
- ticket: Ticketobjekte, welche alle Inhalte der einzelnen Tickets vereinen

Abschließend wurde commands als Konfigurations- und ausführbare Ebene eingeführt. Hier werden Initialisierungsparameter und Ausführungsrichtlinien gesetzt. Da insgesamt drei verschiedene Applikationen gefordert waren existieren hier auch drei unterschiedliche Startkonfigurationen:

- webapp: Startet das eigentliche Ticketssystem und Webapplikation
- restcreateticket: Startet das Kommandozeilentool zum Erstellen eines neuen Tickets über eine bereitgestellte REST-Schnittstelle
- restsendmail: Startet das Kommandozeilentool zur Verwaltung der Mail Queue des Ticketssystems basierend auf REST-Technologie

webserver

Die webserver Komponente stellt einen Zugriffspunkt für den Browser zur Verfügung. Weiterhin werden alle Verbindungen und Verbindungsmöglichkeiten von diesem verwaltet. Mit Hilfe der per Flag übergebenen Zertifikatsdateien stellt der Webserver eine TLS - Only Umgebung zur Verbindung zur Verfügung. Weiterhin werden die HTML-Templatefiles zur Darstellung des Frontends eingebunden. Der Webserver kommuniziert direkt mit der Persistenzschicht um Änderungen direkt zu speichern. Einige Funktionen sind durch eine HTTP-BasicAuth geschützt. Folgende URLs sind zugelassene Zugriffspunkte:

Zugriffs URL	Nur autorisierte Benutzer?	Funktion
/	nein	Startseite / neues Ticket erstellen
/open	ja	Zeigt alle offenen Tickets
/assigned	ja	Zeigt alle dem Benutzer zugewiesenen Tickets
/all	ja	Zeigt alle Tickets des Systems
/new	ja	Neues Ticket erstellen
/edit	ja	Ticket bearbeiten
/edit/add	ja	Fügt der Historie einen Eintrag hinzu
/edit/free	ja	Löscht den Bearbeiter aus dem Ticket
/edit/combine	ja	Kombiniert zwei Tickets mit dem selben Bearbeiter zu einem
/assign	ja	Weist ein Ticket einem Benutzer zu

/api/new	nein	REST-Schnittstelle zum Erstellen eines neuen Tickets
/api/mail	ja	REST-Schnittstelle zum Verwalten der E-Mail Queue
/user/holiday	ja	Setzt einen Benutzer in den Urlaubsmodus

storagehandler

Der storagehandler stellt die Persistenzschicht dar. Hier werden alle Änderungen und Neuanlagen aus dem Frontend übergeben und in Speicherlogik niedergeschrieben. Tickets werden mit ihrer TicketID als einzigartiger Dateiname im Speicher abgelegt. Die TicketID ist ein SHA1-Hash basierend auf Datum der Erzeugung, E-Mail Adresse und Name des Erstellers. Im zugehörigen JSON-File des Tickets werden ebenfalls neue Einträge gespeichert.

Die vom storagehandler nutzbare Dateisystemstruktur sieht wie folgt aus:

Pfad	Beschreibung
storage	Root Speicherverzeichnis
storage/tickets	Ticket Speicherverzeichnis
storage/users.json	Datei mit autorisierten Benutzern und deren Eigenschaften
storage/tickets/*.json	Datei mit Ticketinformationen

email

Als email Objekt werden werden alle zu versendenden oder bereits gesendeten E-Mail Elemente verstanden.

user

In user sind alle Objekte der user.json gelistet. Es stellt die Bearbeiter des Ticketsystems dar.

ticket

Als ticket werden alle Tickets aus dem Storagepfad gehandhabt.

commands

Hier werden die verschiedenen ausführbaren Tools der Applikation gelistet. Es existiert keinerlei Programmlogik in commands selbst, sondern dient nur als Container für ausführbare Packages.

webapp

In der Komponente webapp liegt die Startkonfiguration für die Webapplikation. Die Komponente startet den Webserver und einen storagehandler mit den gegebenen Parametern.

restcreateticket

Das Tool restcreateticket erstellt ein Ticket über die vom Webserver bereitgestellte api/new Schnittstelle. Basierend auf dem REST Prinzip wird hier JSON Data versendet. Die Konfiguration wird über folgende Startparameter getätigt und erstellt ohne zusätzliche Eingabe durch den Benutzer ein Ticket am Zielsystem:

Parameter	Beschreibung
host	Hostname oder IP auf welche verbunden werden soll
port	Port des Zielsystems
mail	Mail Adresse des Absenders
firstname	Vorname des Absenders
lastname	Nachname des Absenders
subject	Titel des neues Tickets
desc	Beschreibung des neuen Tickets

restsendmail

Mit Hilfe von restsendmail werden die zu sendenden Mails abgerufen und verarbeitet. Das Tool greift auf die bereitgestellte api/mail Schnittstelle zu. Nach REST Prinzip werden hier JSON Daten mit Hilfe von Webrequests bereitgestellt. Das Tool deserialisiert diese und bereitet sie für den Benutzer auf. Der Benutzer wählt dann die zu sendenden Mails aus und versendet diese. Abschließend wird das Kennzeichen für eine versendete E-Mail im Ticket gesetzt. Als zusätzliche Ressource wird auf die Das externe Tool wird mit folgenden Startparametern aufgerufen:

Parameter	Beschreibung
host	Hostname oder IP auf welche verbunden werden soll
port	Port des Zielsystems
user	Benutzername zur Authentifizierung am Webserver
pass	Passwort zur Authentifizierung am Webserver

Anwenderdokumentation

Ticketerstellung als unautorisierter Benutzer

Ticketsystem
INF16A TeamKoma

K INC Management

Create New Ticket: [Admin Panel](#)

First and last name

E-Mail

Subject

Description

[Submit new Ticket](#)

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

Wird die Applikation im Browser als unautorisierter Benutzer aufgerufen wird die obenstehende Startseite angezeigt. Nicht explizit autorisierte Benutzer können Tickets anhand der folgenden Merkmale erstellen:

- Vor und Nachname
- E-Mail Adresse
- Titel des Tickets
- Beschreibung des Tickets

Der Button "Submit new Ticket" bestätigt die Eingaben und erstellt nach einer kurzen Prüfung dieser ein neues Ticket im System.

Administrationsoberfläche

Die Administrationsoberfläche hat drei verschiedene Ansichten. Diese sind in der Navigation zu wählen und zu wechseln.

Alle offenen Tickets:

Ticketsystem

INF16A TeamKoma



Welcome Werner

Overview: Open Tickets Tickets Assigned to me All Tickets						Ab in den Urlaub!	New Ticket
#	Subject	Items	Creation Date	Assign to me	Edit		
bba3b4d97c3b0daf7a0d56aec75d674854f164d8	test	1	30 Dec 2018 20:01	Assign To Me	Edit		
9447d3947e3c2a768e55d83177ca0551961a49e4	Bildschirm kaputt	1	30 Dec 2018 20:06	Assign To Me	Edit		
ed12da16844ce2014e9bfff8c64db54ad9e0646e	Drucker geht nicht	1	30 Dec 2018 20:06	Assign To Me	Edit		
ddc5422bed17b2034e29a6510285f556eba113f0	Maus geht nicht	1	30 Dec 2018 20:09	Assign To Me	Edit		
fff705cd0400937a83bd20988dae18eaa738ed19	PC funktioniert nicht	1	30 Dec 2018 20:10	Assign To Me	Edit		

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

Alle offenen Tickets werden gelistet. Zu jedem Ticket existiert ein Button um das Ticket sich selbst zuzuweisen oder es zu editieren. In der Navigation kann der Bearbeiter sich selbst in den Urlaubsmodus versetzen, dann können ihm keine weiteren Tickets mehr zugewiesen werden. Weiterhin kann ein neues Ticket über die Navbar erzeugt werden.

Alle Tickets, die dem angemeldeten Benutzer zugewiesen sind:

Ticketsystem

INF16A TeamKoma



Welcome Werner

Overview: Open Tickets Tickets Assigned to me All Tickets							Ab in den Urlaub!	New Ticket
#	Subject	Creation Date	Items	Last Modified	Reopen	Edit		
9447d3947e3c2a768e55d83177ca0551961a49e4	Bildschirm kaputt	30 Dec 2018 20:06	1	30 Dec 2018 20:06	Reopen	Edit		

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

Diese Ansicht zeigt alle dem Benutzer zugewiesenen Tickets. Mit Hilfe von Reopen kann das Ticket wieder als unzugewiesen markiert werden.

Alle Tickets:

Ticketsystem

INF16A TeamKoma



Welcome Werner

Overview: Open Tickets Tickets Assigned to me All Tickets								Ab in den Urlaub!	New Ticket
#	Subject	Processor	State	Creation Date	Items	Last Modified	Edit		
1ca6e2bb2887c369e6dbc72720a822b82977df8f	asfsff		Open	27 Nov 2018 16:37	3	30 Dec 2018 17:08	Edit		
6fd4989f67fba775f66b2a23b7c5af96b898214f	gdfgfd		Open	27 Nov 2018 16:37	2	27 Nov 2018 16:37	Edit		
bba3b4d97c3b0daf7a0d56aec75d674854f164d8	test		Open	30 Dec 2018 20:01	1	30 Dec 2018 20:01	Edit		
9447d3947e3c2a768e55d83177ca0551961a49e4	Bildschirm kaputt	Werner	In Progress	30 Dec 2018 20:06	1	30 Dec 2018 20:06	Edit		
ed12da16844ce2014e9bfff8c64db54ad9e0646e	Drucker geht nicht		Open	30 Dec 2018 20:06	1	30 Dec 2018 20:06	Edit		
ddc5422bed17b2034e29a6510285f556eba113f0	Maus geht nicht		Open	30 Dec 2018 20:09	1	30 Dec 2018 20:09	Edit		
fff705cd0400937a83bd20988dae18eaa738ed19	PC funktioniert nicht		Open	30 Dec 2018 20:10	1	30 Dec 2018 20:10	Edit		

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

In dieser Ansicht werden alle Tickets inkl. ihrer Bearbeitungsstatus angezeigt. Soll ein Ticket bearbeitet werden muss der zugehörige edit Button gedrückt werden.

Ticketsystem

INF16A TeamKoma



Welcome Werner

ed12da16844ce2014e9bfff8c64db54ad9e0646e [Admin Panel](#)

Creator:	Max Mustermann
Subject:	Drucker geht nicht
Description:	mach wieder ganz!

State: In Progress ▾

Processor: Werner ▾

Save

Combine ed12da16844ce2014e9bfff8c64db54ad9e0646e and Bildschirm kaputt - 9447d3947e3c2a768e55d83177ca0551961a49e4 ▾

Combine

New Entry:

Save

max@mustermann.de

Inform

Entries:

30 Dec 2018 20:06
mach wieder ganz!
Max Mustermann

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

Im Editierungsmodus kann der Status angepasst und der Bearbeiter eines Tickets verändert werden. Dies muss mit Save bestätigt werden.

Darunter findet man die Funktion Tickets zu kombinieren um Tickets mit selbem Inhalt zusammenzufassen, zu beachten ist hierbei, dass beide Tickets dem angemeldeten Benutzer zugewiesen sein müssen.

Die Funktion New Entry fügt einem Ticket einen neuen Historieneintrag hinzu. Dieser kann entweder nur gespeichert oder per Mail gesendet werden.

Entries:

27 Nov 2018 16:37 dfgxs hsdhio hisodhio ToSend: hsfiohs@sodfhsc
27 Nov 2018 16:37 gdfdfg Werner ToSend: hsfiohs@sodfhsc ✓
30 Dec 2018 17:08 neuer entry Werner ToSend: hsfiohs@sodfhsc

INF16A TeamKoma || Programmieren in Go (Doz. H. Nemann) || 2018

Einträge, welche per Mail zu versenden sind werden als solche markiert. Wurde ein Eintrag versendet markiert ein grüner Haken den Versandstatus als abgeschlossen.

Betriebsdokumentation

Ticketsystem Webapplikation

Das Ticketsystem wird über das Tool webapp gestartet. Konfigurationsparameter sind der Komponentenbeschreibung zu entnehmen. Die Applikation ist über den Browser erreichbar sobald eine Meldung in der Konsole ausgegeben wird. Alle Verzeichnisse für die Benutzung des Ticketsystems werden automatisch angelegt, falls sie nicht vorhanden sind. Ein Standardbenutzer wird angelegt, falls keine users.json existiert. Der Standardbenutzer lautet:

Benutzername	Kennwort
admin	admin

REST Client Ticketerstellung

Mit Hilfe von restcreateticket wird ein neues Ticket nach Angabe von Startparametern über die vom Webserver bereitgestellte API-Schnittstelle erstellt. Diese sind der Komponentenbeschreibung zu entnehmen. Die Applikation startet vollautomatisch und ohne Benutzerinteraktion. Zu beachten ist, dass die Webapplikation korrekt ausgeführt und erreichbar sein muss.

REST Client E-Mail Verwaltung

Die geforderte E-Mail Queue Verwaltung wird über das Package restsendmail gestartet. Startparameter sind der Komponentenbeschreibung zu entnehmen. Zur Authentifizierung muss ein valider Benutzer an der Webapplikation angegeben werden. Das Tool benötigt eine lauffähige Instanz der Ticketsystem Webapplikation, um Anfragen verarbeiten zu können. Nach erfolgreicher Verbindung wird eine Benutzerinteraktion gefordert. Diese wird direkt durch das Programm beschrieben.

Aufteilung der Implementierung

9188103

Konzeption

In der Konzeptionsphase hab ich mich hauptsächlich mit den zu speichernden Daten auseinander gesetzt. Dabei wurden Speicherkonzepte zu den Tickets und den Usern entwickelt. Des Weiteren habe ich Tutorials durchgearbeitet zur JSON-Files und deren Speicherung auf dem Filesystem.

Datenspeicherung

Alle Implementierungen rund um das Thema Datensicherung und Datenhandling wurden von mir durchgeführt. Der Code dazu ist im Package storagehandler zu finden. Bei der Implementierung wurde Wert darauf gelegt, dass die persistenten Daten nur beim Serverstart geladen werden. Alle Änderungen, welche zur Laufzeit auftreten, werden sowohl in dem physischen, als auch in den vom Server geladenen Datensätzen geändert. Damit ist ein erneutes Laden der aktualisierten Daten nicht notwendig.

Als Daten werden zwei verschiedene Elemente verwendet. Zum einen werden Tickets benötigt und zum anderen werden User benötigt. Beide Daten werden als JSON-Files abgespeichert. Das Passwort des Users wird mittels SHA256 und Salting gehasht.

Damit die Daten vom Webserver erreichbar und manipulierbar sind, wurden entsprechende Methoden implementiert.

1798794

Konzeption

In der Konzeptionsphase hab ich mich mit der Erstellung eines Webserver in Go auseinandergesetzt mit Hilfe der Folien aus der Vorlesung. Weiter wurden Ideen zur Umsetzung aus dem empfohlenen Video gesammelt. Ebenfalls hab ich einige Online-Tutorials gemacht, welche mich mit der Go-Template Engine vertraut machten.

Webserver

Die Programmierung aller Funktionen rund um den Webserver wurde von mir übernommen. Hierzu zählen neben dem Starten des Webserver und der Auslieferung der Webseite die Bereitstellung der REST-API für die externen Programme. Alle Aktivitäten rund um den Webserver sind in dem Go-Paket webserver gebündelt. Dieser wurde ohne externe Pakete umgesetzt mit der Idee, Funktionen für Teilaufgaben einzelner HTTPS-Requests zu implementieren, welche man für mehrere Requests wiederverwenden kann und auch aneinander hängen kann. Über den von Go bereitgestellten Context für ein HTTPS-Request werden nötige Daten zwischen den einzelnen Händler-Funktionen ausgetauscht. Nötige

Parameter für den Webserver wie z.B. die Zertifikate für den HTTPS-Server werden als Argumente beim Starten des HTTPS-Servers mitgegeben.

HTML Go Templates

Die Webseite wurde mit Hilfe der Go-Template Engine umgesetzt. Von mir wurde die HTML-Mockups in diese Engine umgesetzt und für den leichteren Aufbau der Webseite noch statische Hilfs-Datei wie z.B. Bootstrap eingebunden. Alle Template-Dateien befinden sich in dem Ordner html und werden beim Starten des Webserver eingelesen. Die Templates werden zum Teil wiederverwendet und auch ineinander verschachtelt. So wurde z.B. für den Header ein eigenes Template entwickelt, welches auf mehreren Seiten genutzt wird und

4717960

Konzeption

In der Konzeptionsphase habe ich mich hauptsächlich mit den Grundfunktionen eines Ticketsystem beschäftigt und wie diese am besten darzustellen sind.

HTML Grunddesign

Ein HTML Mockup dient als grundlegendes Design für die Webapplikation. Die Erstellung des Mockups und die Konzeptionierung einer grundlegenden Designidee ist in meine Hände gefallen.

Konfiguration und Startparameter

REST Client E-Mail Verwaltung

Die Anforderung der kleineren Tools zum Testen der REST E-Mail Schnittstelle war ebenso vollständig meine Aufgabe.

REST Client Ticketerstellung

Weiterhin war die vollständige Implementierung des REST Clients für die Ticketerstellung meine Aufgabe.

Dokumentation

Das Aufbereiten und Visualisieren und Beschreiben der einzelnen Komponenten für die Architekturdokumentation wurde vollständig von mir übernommen. Neben der Anwenderdokumentation habe ich auch die Betriebsdokumentation vollständig erstellt.