Dokumentation des Go-Projekts

Von 2416160, 5836402

1 Architekturdokumentation

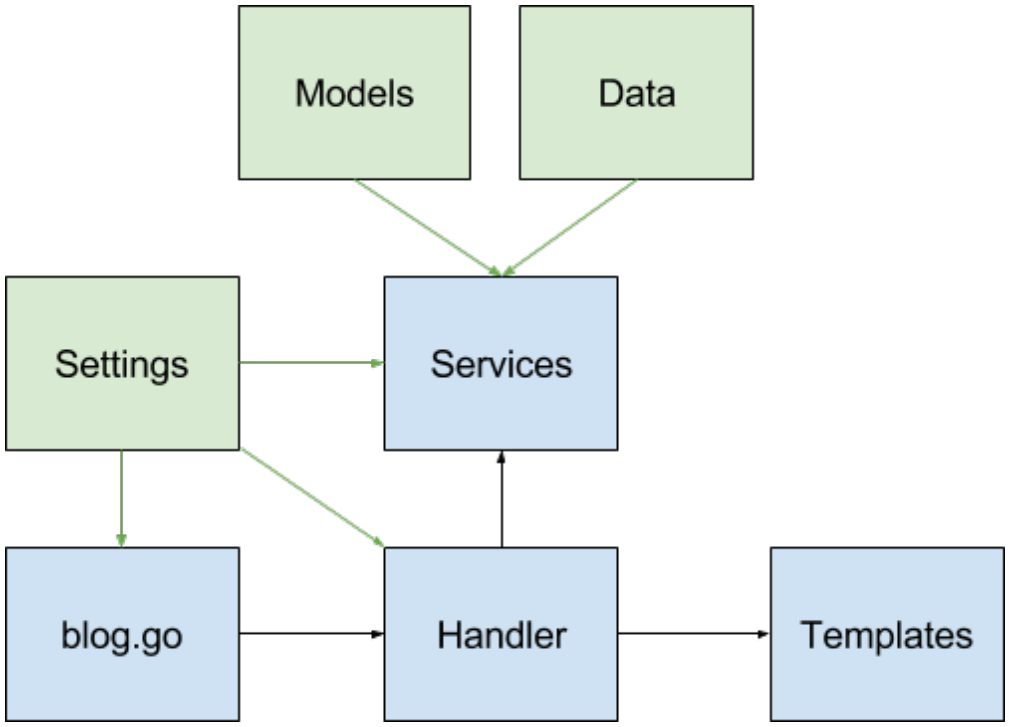

Abbildung 1: Zusammenhang der einzelnen Komponenten

In Abbildung 1 ist eine Übersicht über die Komponenten zu sehen. Die blauen Komponenten sind Teil des Funktionsflusses und die grünen stellen Datenelemente dar. Das Programm beginnt in der Routine *blog.go* und registriert dort alle Handler. Außerdem ist dort das REPL umgesetzt. Bei Aufruf einer gültigen URL wird der entsprechende Handler aufgerufen. Ein Handler ruft je nach Bedarf ein oder mehrere Services auf um Daten abzurufen oder Funktionen auszuführen. Zum Beispiel kann er vom *sessionService* die aktuelle Session abfragen oder eine neue erstellen lassen sofern noch keine vorhanden ist. Die Session wird ihm dann in Form eines *structs*, das in Models definiert ist, übergeben. Alle verwendeten *structs* sind im Package "models" definiert. Im Package "data" liegen alle Daten-Dateien für die Settings, die BlogPosts und die User, also alle Daten die persistent gespeichert werden sollen. Die Settings sind hier nochmals besonders hervorgehoben, da sie zur Laufzeit im Speicher global allen Funktionen zur Verfügung stehen sollen. Aus diesem Grund liegt ihre Instanz im Package "global". Hat ein Handler alle Daten die er braucht und alle nötigen Funktionen ausgeführt, so wird er entweder ein Template rendern und diesem Daten mitgeben oder einen Redirect auf einen anderen Handler starten.

2 Anwenderdokumentation

Beim Aufruf der Seite (lokal via "localhost") wird der Nutzer direkt auf die gesicherte https-Ansicht umgeleitet. Auf der Startseite wird dem Nutzer der aktuellste Post dargestellt, falls verfügbar. Unter der Überschrift befindet sich ein Link zur Archivdarstellung, auf der sich alle Posts befinden, der sich auf der Archivdarstellung zu einem Link auf die Startseite ändert. Alle Posts sind mit den zugehörigen Kommentaren unterlegt, die von jedem Nutzer erstellt werden können. Vorausgesetzt der Nutzer besitzt ein Authorenkonto, kann er sich am unteren Ende der Seiten über den Link "Login" einloggen. Auf der Seite befinden sich lediglich die Felder für den Benutzernamen und das Passwort, bei Fehlschlägen wird ein Fehler angezeigt (z.B. "Falsches Passwort").

Ist der Nutzer erfolgreich eingeloggt, kann er sich jederzeit über den Link, wieder am unteren Ende der Seiten, ausloggen. Ebenfalls ist es eingeloggten Nutzern möglich Posts zu verfassen, und zwar über das Feld und den Button "New Post", jeweils am unteren Ende der Auflistungen der Posts auf der Index- und Archivseiten. Neben dem Logout Button am unteren Ende der Seite kann der eingeloggte Nutzer sein Passwort ändern. Zuletzt ist es eingeloggten Nutzern möglich ihre eigenen Posts aufzulisten, über den Link "My Posts" am unteren Ende der Seite. Auf dieser Seite kann jeder Post einzeln editiert oder gelöscht werden.

Nicht eingeloggte Nutzer die auf Authorenfunktionen zugreifen werden auf die Startseite zurücknavigiert.

3 Inbetriebnahme

Der Server kann gänzlich ohne Anpassungen gestartet werden. Die Standardeinstellungen sehen wie folgt aus:

PortNumber	4443
SessionTimeout	15
PostSuffix	.json
KeyFile	key.pem
CertFile	cert.pem

Die Standardeinstellungen werden von den Einstallungen in der settings-Datei überschrieben, die wiederum von den Einstellungen der CommandLine Parameter überschreibene werden, vorrausgesetzt diese existieren jeweils.

Standardmäßig ist auch ein root Benutzer eingerichtet, der folgende Logindaten hat: Name: "root", Passwort:"toor".

3.1 Serverinformationen via REPL

Über einen sog. "Read Eval Print Loop" ist es möglich im laufenden Betrieb Informationen abzurufen. Das Kommando "?" zeigt die Befehle an, die akzeptiert werden. Die Befehle sehen folgendermaßen aus:

Kommando	Effekt	Beschreibung
q	quit	Beendet den Server
s	settings	Gibt die aktuell aktiven Einstellungen aus
v	version	Gibt die Versionsnummer aus
u	users	Gibt alle aktiven User aus (alle Users mit Session)
(r config)	-	Undocumented: Lädt die Konfiguration neu

3.2 Einstellungen via CommandLine Parameter

Folgende CommandLine Parameter stehen zur Verfügung:

Parameter	Bedeutung
-port	Ändert den Port der Anwendung
-timeout	Stellt den Session Timeout ein
-postSufx	Dateiendung der Posts
-keyFile	Name des Server-Schlüssels
-certFile	Name des Zertifikats

Beispiel: "blog -port=1234SStartet den Blog auf Port 1234.

4 Beitrag zum Projekt

4.1 2416160

- Entwurf und Umsetzung der Templates (index, archive, login, changePassword, myPosts)
- Entwurf und Implementierung der BlogPost Funktionen (LoadPosts, GetPosts, Save-, New-, Change-, DeletePost)
- Entwurf und Implementierung der Settings Funktionen (Defaults, File und CommandLine Parameter)
- Entwurf und Umsetzung der Benutzerverwaltung
- Entwurf und Einbau der Krypto-Funktionen
- Entwurf und Bau des REPL's
- Einbau der Cookie Verwaltung
- Handler und Redirect (redirect to index, tls redirect) Logik
- Zertifikate
- Unit Tests für PostsService und UserService
- Dokumentation

4.2 5836402

- Entwurf und Umsetzung der Anwendungsarchitektur
- Entwurf und Implementierung der Dateisystem-Funktionen
- Entwurf und Implementierung des SessionHandlings
- Entwurf und Implementierung der Template Funktionen
- Schutz der entsprechenden Pfade vor unauthorisiertem Zugriff (CheckSession)
- Zertifikate
- Tests für FileService
- Dokumentation