Programmieren II GO

Dokumentation

5807262, 9899545 & 8622410

1. How to start the server?

1. Run the main.go with:

go run main.go

If you want to see all possible flags: go run main.go -h

If you want to change flags: go run main.go -flagName flagValue

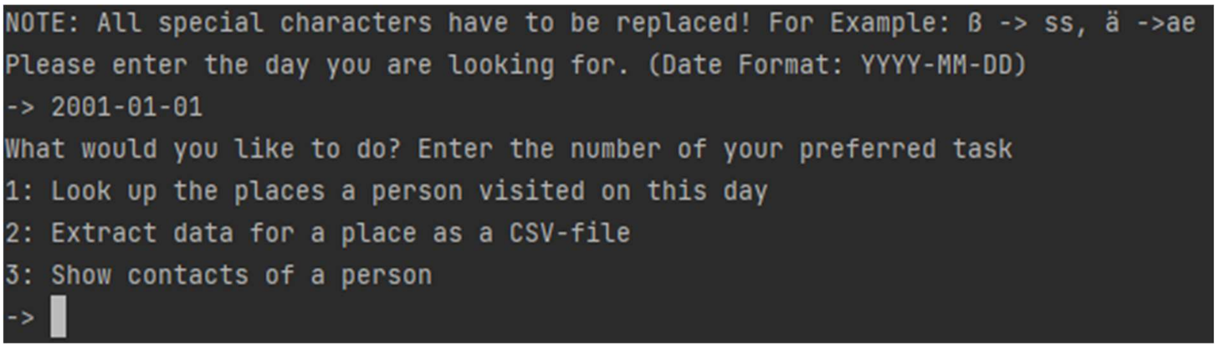
2. Now you’ve been asked to make a choice:

„1“ Starts the Analyzer

„2“ Starts the Server

If you choose the Analyzer:

Follow the Instructions given by the Command line



```
NOTE: All special characters have to be replaced! For Example: ß -> ss, ä ->ae
Please enter the day you are looking for. (Date Format: YYYY-MM-DD)
-> 2001-01-01
What would you like to do? Enter the number of your preferred task
1: Look up the places a person visited on this day
2: Extract data for a place as a CSV-file
3: Show contacts of a person
->
```

FYI: The CSV-file of „2“ can be found under „model/log/files/date_location.csv“

If you choose the Server:

Under the link you find a page where you can choose the Location as an organizer of an registered event.

https://127.0.0.1:8443/location

Important: This Link is not for the customers

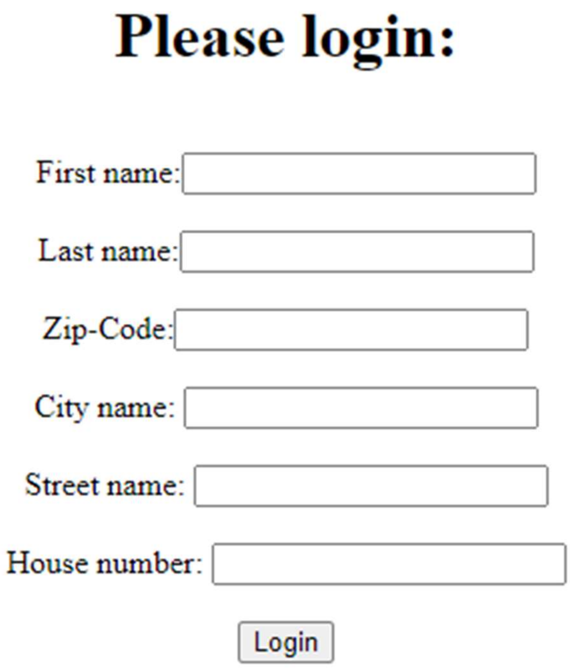
There you choose your Location, for example Mosbach

This takes you to the page where the QR Code is displayed:

https://127.0.0.1:8444/Mosbach?

Scanning the QR Code takes you to the Login Page

Here you can log in with the necessary Information.



Please login:

First name:

Last name:

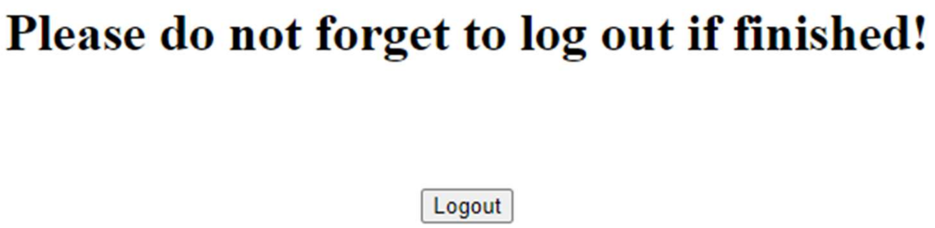
Zip-Code:

City name:

Street name:

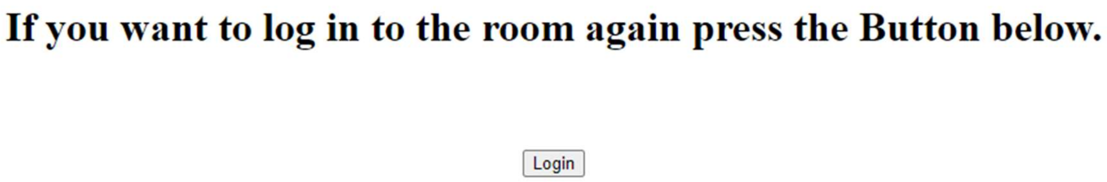
House number:

If Login is done, you’re redirected to the Logout page, where you’ve to log out if you're leaving.



Please do not forget to log out if finished!

After the Logout your redirected to the Re Login Page in Case your going back to the Location



If you want to log in to the room again press the Button below.

Because of storing cookies you’ll automaticly login until they were deleted.

2. Architecture MVVM

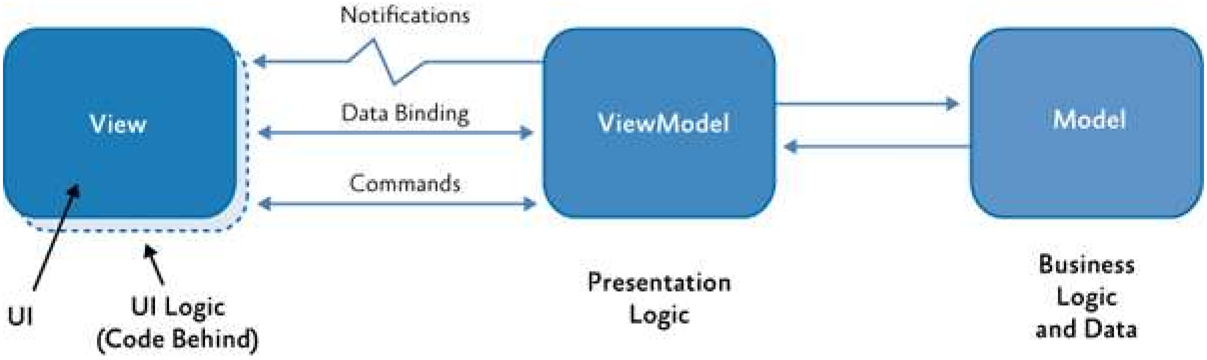
Used in the project is a three-layer architecture, which has three layers on the software side.

Our three-layer architecture consists of the following three layers:

Presentation layer: this represents the front-end and is responsible for representing the data, user input and the user interface responsible on the website. This includes everything in the "view" folder.

Logic layer: this contains all our application logic. So almost everything that is stored in the "viewmodel" folder. The analyzer, the handler, the config and the token.

Data storage layer: Is the "model" folder. It contains the xml and log files and is responsible for saving and loading the data.



Source: http://agredo.agredoapplication.com/2018/05/02/einfuehrung-in-die-uwp-und-xamarin-programmierung-model-view-viewmodel-mit-xamarin-forms/

3. Distribution of the tasks

5807262:

handler.go

handler_test.go

tokenProvider.go

tokenProvider_test.go

9899545:

analyzer.go

analyzer_test.go

logfileParser.go

logfileParser_test.go

8622410:

xmlParser.go

xmlParser_test.go

configuration.go

configuration_test.go

5807262, 9899545 & 8622410:

main.go

Please note that the partition of the tasks is merely done because it is a requirement. In reality we have supported each other and each one has contributed to each task.