We wrote API on golang which has user model. Now registration and login is available on our app.

```go
type User struct {
        gorm.Model
        Nickname   string `json:"nickname"`
        Status     string `json:"status"`
        FirstName  string `json:"firstname"`
        SecondName string `json:"lastname"`
        Password   string `json:"password"`
        Age        uint16 `json:"age"`
}

func InitialMigration() {
        db, err = gorm.Open(sqlite.Open(DSN), &gorm.Config{})
        if err != nil {
                fmt.Println(err.Error())
                panic("cannot connect to db")
        }
        db.AutoMigrate(&User{})
}

func UpdateUser(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "application/json")
        data := mux.Vars(r)
        var user User
        db.First(&user, data["id"])
        json.NewDecoder(r.Body).Decode(&user)
        db.Save(&user)
        json.NewEncoder(w).Encode(user)
}
```

```go
func DeleteUser(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "application/json")
        data := mux.Vars(r)
        var user User
        db.Delete(&user, data["id"])
        json.NewEncoder(w).Encode("User successfully deleted")
}

func GetUsers(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "application/json")
        var users []User
        db.Find(&users)
        json.NewEncoder(w).Encode(users)
}

func GetUser(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "application/json")
        data := mux.Vars(r)
        var user User
        db.First(&user, data["id"])
        json.NewEncoder(w).Encode(user)
}

func CreateUser(w http.ResponseWriter, r *http.Request) {
        w.Header().Set("Content-Type", "application/json")
        var user User
        json.NewDecoder(r.Body).Decode(&user)
        db.Create(&user)
        json.NewEncoder(w).Encode(user)
}
```