Ruta: GET /register

Opis: Ruta za registraciju, svim korisnicima omogucen pristup

Primer http zahteva: GET http://localhost:8080/register

Opis odgovora u slucaju greske: 500 {err: “server err”}

Ruta: POST /login

Opis: Ruta za login, ako se admin loginuje ispisace ko se logovo

Parametar koji se ocekuje: query parameter, ocekuje ime I password

Primer http zahteva: POST http://localhost:8080/login?name=admin&password=admin

Primer nevalidnog http zahteva: POST <http://localhost:8080/login?name=admin>

Opis odgovora: 200 {name:”admin”, password: “password”}

Opis odgovora u slucaju greske: 404 {err: “User not found”}

Ruta: GET /getUsers

Opis: Samo admin ima pristup ovoj ruti, ispisuje sve korisnike

Primer http zahteva: GET <http://localhost:8080/getUsers>

Opis odgovora: {“admin”:”admin”, “user”,”password”}

Opis odgovora u slucaju greske: 403 {err: Forbidden}

Ruta: POST /changeUser

Opis: Svi useri imaju pristup ruti, menja naziv usera I sifru usera

Parametar koji se ocekuje: query parameter, ocekuje ime I password

Primer http zahteva: POST http://localhost:8080/changeUser?name=user1&password=user1

Primer nevalidnog http zahteva: POST <http://localhost:8080/changeUser?name=user1&>

Opis odgovora: {user: “user1”, password: “password”}

Opis odgovora u slucaju greske: 400 {err: Invalid param}

Ruta: DELETE /deleteUser/:userID

Opis: Admin ima pristup, brise usera iz fajla

Parametar koji se ocekuje: indentifikator korisnika, tj naziv korisnika

Primer http zahteva: DELETE http://localhost:8080/user1

Primer nevalidnog http zahteva: DELETE <http://localhost:8080/>

Opis odgovora: “User deleted”

Opis odgovora u slucaju greske: 401 {err: User not authorised}

Ruta: GET /listaPica

Opis: ova ruta dobavlja podatke o picima iz fajla drinks.json

Primer http zahteva: GET <http://localhost:8080/listaPica>

Opis odgovora: {drinks: [{id: 11,name: “coca-cola”, price: 1.8}, {id:12, name: “coca-cola”, price: 1.8}] }

Opis odgovora u slucaju greske: 500 {err: Internal server error}

Ruta: POST /changePrice/:drinkID

Opis: Admin ima pristup, menja cenu pica u drinks.json

Parametar koji se ocekuje: indentifikator pica, podrzane su cifre

Primer http zahteva: GET <http://localhost:8080/changePrice/808> {“price”:2}

Primer nevalidnog http zahteva: GET <http://localhost:8080/changePrice/808> {“price”: }

Opis odgovora: {name: “coca-cola”, price:2}

Opis odgovora u slucaju greske: 400 {err: Invalid params}, 401 {err: User not authorize}

Ruta: POST /addDrink

Opis: Admin ima pristup, dodaje novo pice u drinks.json

Parametar koji se ocekuje: req.body.name, req.body.price, u telu zahteva se ocekuje

Primer http zahteva: GET <http://localhost:8080/addDrink> {“name”:”coca-cola”,”price”:1.8}

Primer nevalidnog http zahteva: GET <http://localhost:8080/addDrink> {“price”:1.8}

Opis odgovora: {id:342, name:“pepsi”, price:2}

Opis odgovora u slucaju greske: 400 {err: Invalid params}, 401 {err: User not authorized}

Ruta: GET /tablesInUse

Opis: svim korisnicima omogucen pristup, vide koji su stolovi slobodni, a koji zauzeti

Primer http zahteva: GET <http://localhost:8080/tablesInUse>

Opis odgovora: {table1:true, table2:true}

Opis odgovora u slucaju greske: 500 {err: Internal server error}

Ruta: POST /assignTable

Opis: admin I employees imaju pristup, umesto da je sto false dodeljuju mu vrednost true u tables.json fajlu

Primer http zahteva: POST <http://localhost:8080/assignTable>

Opis odgovora: {table3:true }

Opis odgovora u slucaju greske: 401 {err: User not authorized}

Ruta: POST /freeTable

Opis: admin I employees imaju pristup, umesto da je sto true dodeljuju mu vrednost false u tables.json fajlu

Parametar koji se ocekuje: query parameter, naziv stola

Primer http zahteva: Post <http://localhost:8080/freeTable>

Opis odgovora: {table3:false }

Opis odgovora u slucaju greske: 401 {err: User not authorized}

Ruta: POST /getPhoto

Opis: korisnici imaju pristup, preuzimanje slike

Parametar koji se ocekuje: u objektu se nalazi path gde se trenutno nalazi slika

Primer http zahteva: Post <http://localhost:8080/getPhoto>

Opis odgovora: {path: “server/temporaryPath”}

Opis odgovora u slucaju greske: 400 {err: Invalid params}

Ruta: GET /logout

Opis: logout ruta

Primer http zahteva: GET <http://localhost:8080/logout>

Opis odgovora: 200 {status: “Bye!”}

Opis odgovora u slucaju greske: 500 {err: Server error}