

# Heroku-Anleitung

## Inhalt

DB-Zugangsdaten .....	2
Gruppe 1 (Mike, Robin, Denis): .....	2
Gruppe 2 (Timo, Nadine, Steven): .....	2
Gruppe 3 (Dylan, Claudia): .....	2
Gruppe 4 (Severin): .....	2
URL für API-Zugriff (Postman, App) .....	2
Deploy von Anwendung .....	3
Vorbereitungen: .....	3
Schritt 1: DB-Zugangsdaten in REST-API ändern .....	3
Schritt 2: Datenbank importieren .....	3
Schritt 3: Ordner für GIT-Repository erstellen .....	3
Schritt 4: Deploy der REST-API auf Server .....	3
Testen der REST-API: .....	4
Registrieren .....	4
Login .....	4
User-Liste abfragen .....	5

## DB-Zugangsdaten

### Gruppe 1 (Mike, Robin, Denis):

**Hostname:** eu-cdbr-west-02.cleardb.net

**Datenbank:** heroku\_6f051d1ae5bfa3e

**Username:** b5282ccf39ae83

**Passwort:** 26e79bce

### Gruppe 2 (Timo, Nadine, Steven):

**Hostname:** eu-cdbr-west-02.cleardb.net

**Datenbank:** heroku\_b1a017ede6874d9

**Username:** b4745bf70c00bb

**Passwort:** 7676818e

### Gruppe 3 (Dylan, Claudia):

**Hostname:** eu-cdbr-west-02.cleardb.net

**Datenbank:** heroku\_bb8b8f8d2e018d6

**Username:** b15e7d229e207d

**Passwort:** ec62f8c4

### Gruppe 4 (Severin):

**Hostname:** eu-cdbr-west-02.cleardb.net

**Datenbank:** heroku\_d173ecace30a069

**Username:** bece104bdb064b

**Passwort:** 6c0f26e7

## URL für API-Zugriff (Postman, App)

Gruppe 1: <https://uek223-gruppe1.herokuapp.com/>

Gruppe 2: <https://uek223-gruppe2.herokuapp.com/>

Gruppe 3: <https://uek223-gruppe3.herokuapp.com/>

Gruppe 4: <https://uek223-gruppe4.herokuapp.com/>

# Deploy von Anwendung

## Vorbereitungen:

- Heroku Account erstellen und via. Kursleiter zum Heroku-Projekt hinzufügen.
- Heroku CLI herunterladen und installieren:  
<https://devcenter.heroku.com/articles/heroku-cli>

## Schritt 1: DB-Zugangsdaten in REST-API ändern

Bevor Sie die REST-API hochladen können, müssen Sie im **application.properties** die DB-Zugangsdaten ändern auf die DB von Heroku (siehe oben).

Beispiel für die REST-APP-Demo auf GitHub:

```
spring.jpa.hibernate.ddl-auto=none
spring.datasource.url=jdbc:mysql://eu-cdbr-west-02.cleardb.net:3306/heroku_d173ecace30a069
spring.datasource.username=b5d741d263cb7d
spring.datasource.password=148b22ba
```

## Schritt 2: Datenbank importieren

Verbinden Sie sich mit MySQL-Workbench und den oben genannten Zugangsdaten auf Ihre Heroku-Datenbank. Erstellen Sie ihre Datenbank mit einem SQL-Script (z.B. Export von lokaler DB)

## Schritt 3: Ordner für GIT-Repository erstellen

Sie laden die App via GIT-Repository auf den Heroku-Server hoch. Dies muss in einem Verzeichnis gemacht werden, in welchem noch kein anderes GIT-Repository vorhanden ist. Dies ist bei Ihnen wahrscheinlich aber der Fall. Gehen Sie wie folgt vor:

1. Erstellen Sie am gewünschten Ort auf Ihrem PC einen neuen Ordner für das Heroku-Git-Repository.
2. Kopieren Sie den Code Ihrer REST-API (d.h. Projektordner) in das neu erstellte Verzeichnis.
3. Löschen Sie innerhalb des Projektordners das Verzeichnis **.git**

## Schritt 4: Deploy der REST-API auf Server

1. Konsole öffnen
2. In den bei Schritt 3 erstellten Ordner (In Projektordner) navigieren mit cd.
3. Eingabe von „heroku login“, Enter klicken
4. E-Mail Adresse und Passwort eingeben, Enter klicken
5. Nun folgende Befehle eingeben um das GIT-Repository zu erstellen:

```
$ cd my-project/
$ git init
$ heroku git:remote -a uek223-gruppe1
```

6. GIT-Repository pushen nach Heroku-Master

```
$ git add .  
$ git commit -am "first push"  
$ git push heroku master
```

7. GIT-Repo wird nun gepusht und REST-API automatisch gestartet (kann ein paar Minuten dauern)

## Testen der REST-API:

### Registrieren


The screenshot shows a REST client interface with a POST request to `https://rest-app-login.herokuapp.com/loginrestapi/sign-up`. The 'Body' tab is selected, and the request body is a JSON object: `{ "email": "holzer@nyp.ch", "password": "12345", "firstname": "Joel", "lastname": "Holzer" }`. The 'form-data' and 'x-www-form-urlencoded' options are unselected, while 'raw' is selected. The 'JSON (application/json)' option is also visible.

### Login


The screenshot shows a REST client interface with a POST request to `https://rest-app-login.herokuapp.com/login`. The 'Body' tab is selected, and the request body is a JSON object: `{ "email": "holzer@nyp.ch", "password": "12345" }`. The 'form-data' and 'x-www-form-urlencoded' options are unselected, while 'raw' is selected. The 'JSON (application/json)' option is also visible.

Below the request body, the 'Headers' tab is selected, showing 13 headers. The 'Authorization' header is highlighted, showing a Bearer token: `Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJMSlMlV4cCI6MTUzMDQ2Nzk1MH0.IGRlCPbIWYMUFXKhIA5WIRY9QITr70vcEGs3LkeJLAn9GtubtAQ1j6YZkOoQoQIZWWUpvALHfK9-ndUtWQoEQ`.

## User-Liste abfragen

GET 

https://rest-app-login.herokuapp.com/loginrestapi/users/all

Authorization 


Headers (1)

Body

Pre-request Script


Tests

TYPE

Bearer Token 

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Preview Request

 Heads up! The

Token

Body

Cookies


Headers (13)


Test Results


Pretty


Raw

Preview

JSON 



1 

2 

3


4

5

6

7

8

9 

10

11

12

13

14

15

16

[

{

"id": 11,

"email": "joel.holzer@nyp.ch",

"password": "\$2a\$10\$i298ibKawKeRBgQmAq00DezNui/3vEqJQtCZ2it3D2VJzJva9rCMu",

"firstname": "Joel",

"lastname": "Holzer"

},

{

"id": 21,

"email": "holzer@nyp.ch",

"password": "\$2a\$10\$v/5MmxsS9XytbRub/w5yt.000e0M.km5f7pKjzN2h2Cl5N0rWsaqy",

"firstname": "Joel",

"lastname": "Holzer"

}

]