

Leaderboard

```
@app.post("/scores", dependencies=[Depends(bearer)])
async def set_score(request: Request, score: int):

    data = database()
    score = int(request.query_params['score'])

    auth_token = request.headers.get('Authorization').split()[1]
    payload = JsonWebToken(auth_token).validate()

    if not data.get(payload['sub']): # new user
        data[payload['sub']] = [score,]
    else: # pre-existing user
        data[payload['sub']].append(score)

    with open('database.json', 'w') as f:
        f.write(json.dumps(data))

    return {"status": "ok", "message": "Data written successfully."}
```

Leaderboard

```
@app.get("/", include_in_schema=False)
async def top_players(request: Request):
    data = database()

    if data.__len__() < 10:
        top_players = sorted(
            data, key=lambda x: max(data[x]),
            reverse=True)[:data.__len__()]
    else:
        top_players = sorted(
            data, key=lambda x: max(data[x]),
            reverse=True)[:10]

    if top_players == []:
        return templates.TemplateResponse(
            "leaderboard.html",
            {"request": request, "top_players": {}}
        )
    # . . .
```

