

Work Orders

Work Orders is a Backend Assessment organized by [Hatchways](#). This document shows how to run the API, as well as additional notes on decisions made. This assessment has been deployed as an Heroku app. You can find it at <https://hatchwaysassessment.herokuapp.com/api/>.

Installation (step by step)

- Unzip the compressed folder
- Prepare a virtual environment
- In your terminal, cd into the hatchways folder
- Run `pip install -r requirements.txt`. Note that some of the requirements were only essential in deployment, but it doesn't matter.
- Run `python manage.py runserver`. If the port 8000 is busy, switch ports using `python manage.py runserver [port_number]`

Usage (step by step)

- Launch your browser, open the link <http://127.0.0.1:8000/api/>. You'll see the API Root.
- You'll find a dictionary. In this dictionary, have Model class names as keys and links to access or work with them as values.

Working with the Worker model

- If you click the link <http://127.0.0.1:8000/api/workers/>, you'll get access to a list of dictionaries containing details about every worker in the database. There's also a form below to create or add to the list.
- Add a worker by populating the form fields with appropriate data, and hit the POST button. You'll get a dictionary containing details about the just added worker. Refresh to get the entire list again.
- To modify details about a particular worker in the list, or even delete the worker, look up the id of such worker and add it to the path - that is, <http://127.0.0.1:8000/api/workers/{id}/>. You'll get details for the worker and a prepopulated form field containing the worker's data. Feel free to edit data in the form fields and hit the PUT button. You'll get a dictionary containing modified worker data. To delete, there's a red DELETE button at the top. Then, go back to the list of workers in <http://127.0.0.1:8000/api/workers/> to see the user can't be found.

Working with the WorkOrder model

- If you click the link <http://127.0.0.1:8000/api/workorder/>, you'll get access to a list of dictionaries containing details about every work order in the database. There's also a form below to create or add to the list.
- Add a work order by populating the form fields with

appropriate data. Note that you can only add a maximum of 5 worker to a work order, so adding more than five will give an error message in dictionary form. The dictionary key being the field on which the error occurred and the value being the fact that not more than five workers can be added to a task. When you're done, hit the POST button. You'll get a dictionary containing details about the just added work order. Refresh to get the entire list again.

- When selecting workers in the 'assigned to' field, hold down the ctrl key so that you can select multiple workers at once.
- To modify details about a particular work order in the list, or even delete the work order, look up the id of such work order and add it to the path - that is, <http://127.0.0.1:8000/api/workorder/{id}/>. You'll get details for the work order and a prepopulated form field containing the work order data. Feel free to edit data in the form field and hit the PUT button. You'll get a dictionary containing modified work order data. To delete, there's a red DELETE button at the top. Then, go back to the list of work orders in <http://127.0.0.1:8000/api/workorders/> to see the work order can't be found.

Fetching all work orders

- To fetch all work orders for a specific worker, sorted by

deadline (as stated in the requirements for this assessment), look up the worker's id, then open the link <http://127.0.0.1:8000/api/worker/{id}/workorders/>

Routes Summary

- <http://127.0.0.1:8000/api> (api root)
- <http://127.0.0.1:8000/api/workers/> (get a list of all workers)
- <http://127.0.0.1:8000/api/workers/{id}/> (get a particular worker for modification or deletion)
- <http://127.0.0.1:8000/api/workorder/> (get a list of all work orders)
- <http://127.0.0.1:8000/api/workorder/{id}/> (get a particular work order for modification or deletion)
- <http://127.0.0.1:8000/api/worker/{id}/workorders/> (get a list of work orders for a particular worker)
- <http://127.0.0.1:8000/api/docs/> (interactive api documentation)
- <https://hatchwaysassessment.herokuapp.com/api/> (to use online)

Kindly replace <http://127.0.0.1:8000/> with

<https://hatchwaysassessment.herokuapp.com/> to use online

The project meets all requirements as specified in the assessment text and more. There's also an interactive documentation page (interactive because you can try things out on the page). Open the link <http://127.0.0.1:8000/api/docs/> to see this.

This project has been deployed as an herokuapp. You can find it at

<https://hatchwaysassessment.herokuapp.com/api/>. Follow the same instructions given above. Please note that this task was completed in two days. The time on the dashboard should be ignored, because I had to pick up the assessment weeks later for reasons I am willing to discuss, if you are interested.

Please send me a mail for further questions. Thank YOU!