

---

# PyBites Platform

*Release 0.0.1*

**Jul 26, 2023**



<b>1</b>	<b>Welcome to the PyBites CodeChallenges Platform</b>	<b>1</b>
<b>2</b>	<b>Platform Administration</b>	<b>3</b>
<b>3</b>	<b>How to Solve a Bite</b>	<b>5</b>
<b>4</b>	<b>Premium Tier</b>	<b>7</b>
<b>5</b>	<b>Recruiting talent via the PyBites CodeChallenges Platform</b>	<b>9</b>
5.1	Starting with why . . . . .	9
5.2	Pre-requisite . . . . .	9
5.3	Register your license key . . . . .	9
5.4	Your own space on PyBites . . . . .	12
5.5	Add Interview Bites . . . . .	14
5.6	Copy our Bites . . . . .	16
5.7	Interviewee links . . . . .	20
5.8	Interviewing people . . . . .	24
5.9	Interview submissions . . . . .	28
5.10	Download interview data . . . . .	29
5.11	Feedback . . . . .	30
<b>6</b>	<b>Using the Enterprise Tier</b>	<b>31</b>



---

## Welcome to the PyBites CodeChalleng.es Platform

---

Our goal with our coding platform is to get you coding. A lot!

We have grown a vibrant community of Pythonistas eager to learn every nook and cranny of the Python language and ecosystem.

However, just reading a book or watching a video tutorial is not going to cut it. You need real world, hands-on practice.

Enter PyBites CodeChallenges! We have almost 200 exercises on our platform that you can code in the comfort of your browser. All of these *Bites of Py*, as we call them, come with automated tests to verify your code submissions.

You can share your code on the forums, learning alongside other developers, earning Ninja belts and certificates as you go.

Sign up today, confirm your email and start your free 14 day trial: <https://codechalleng.es>



## CHAPTER 2

---

### Platform Administration

---

CodeChallenges Administration



## CHAPTER 3

---

### How to Solve a Bite

---

Here is a walk-through of how to solve a Bite on our platform. Although the design changed a bit, core functionality did not:

---

And here is another video where you see us solve [Bite #21](#):

---

If you get stuck on an exercise we recommend you do the following:

- Step away from your computer for a bit, take a walk, you may be surprised how effective this can be.
- Hop onto [our Slack](#) and ask in our [#codechallenges](#) channel (no spoilers please).



## CHAPTER 4

---

### Premium Tier

---

Using the Premium Tier



---

## Recruiting talent via the PyBites CodeChallenges Platform

---

PyBites CodeChallenges allows you to run your own Python based coding interviews using your own custom Bites (exercises) or from our library of Bites.

### 5.1 Starting with why

For some background on this technology / tier, read our article: [Why whiteboard interviews suck and what we're doing about it](#).

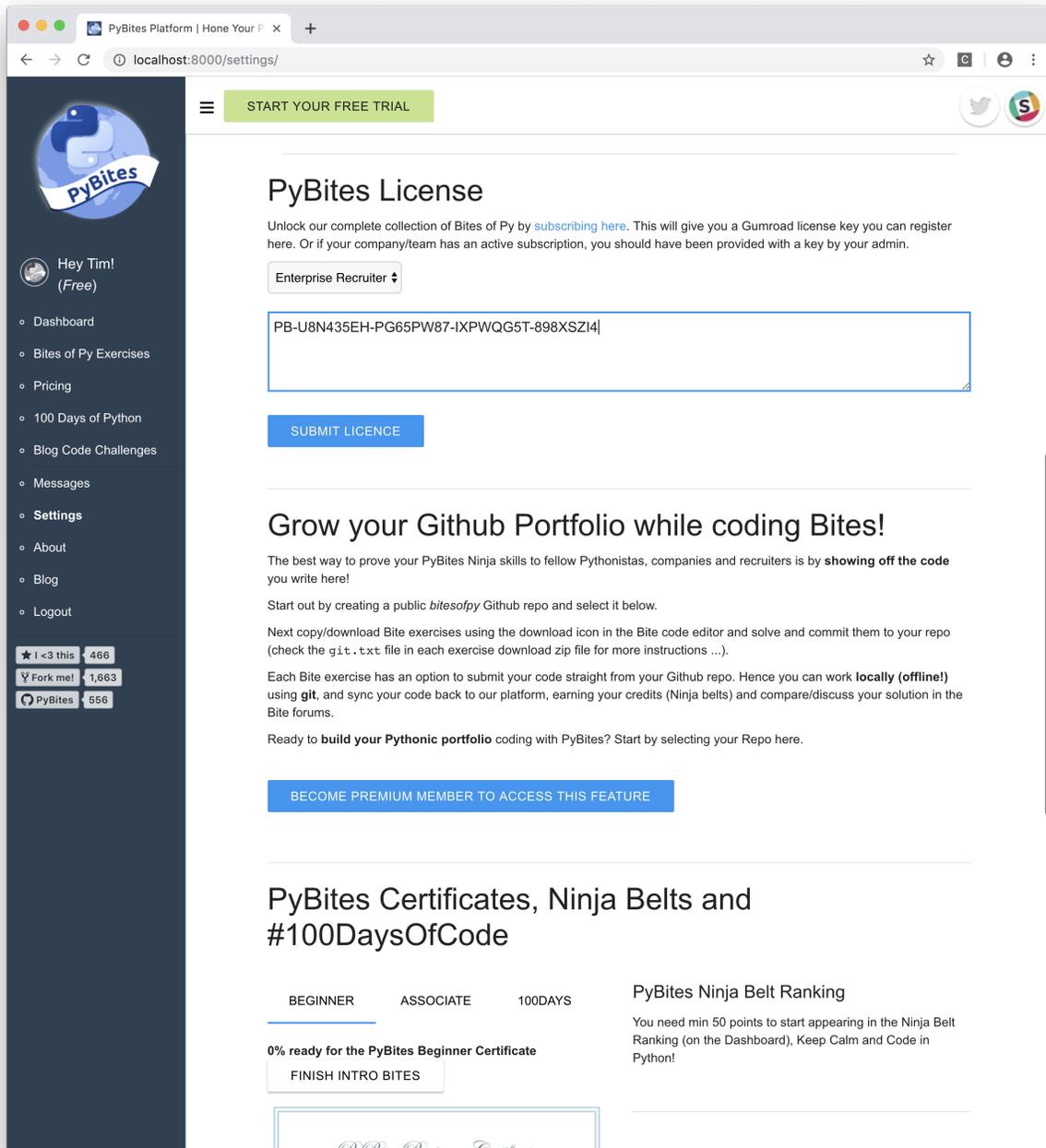
### 5.2 Pre-requisite

You have to be signed up to our platform with a username/email, or with a Github Account: <http://codechalleng.es/signup>. Make sure you confirm your email as this is required to register a license key.

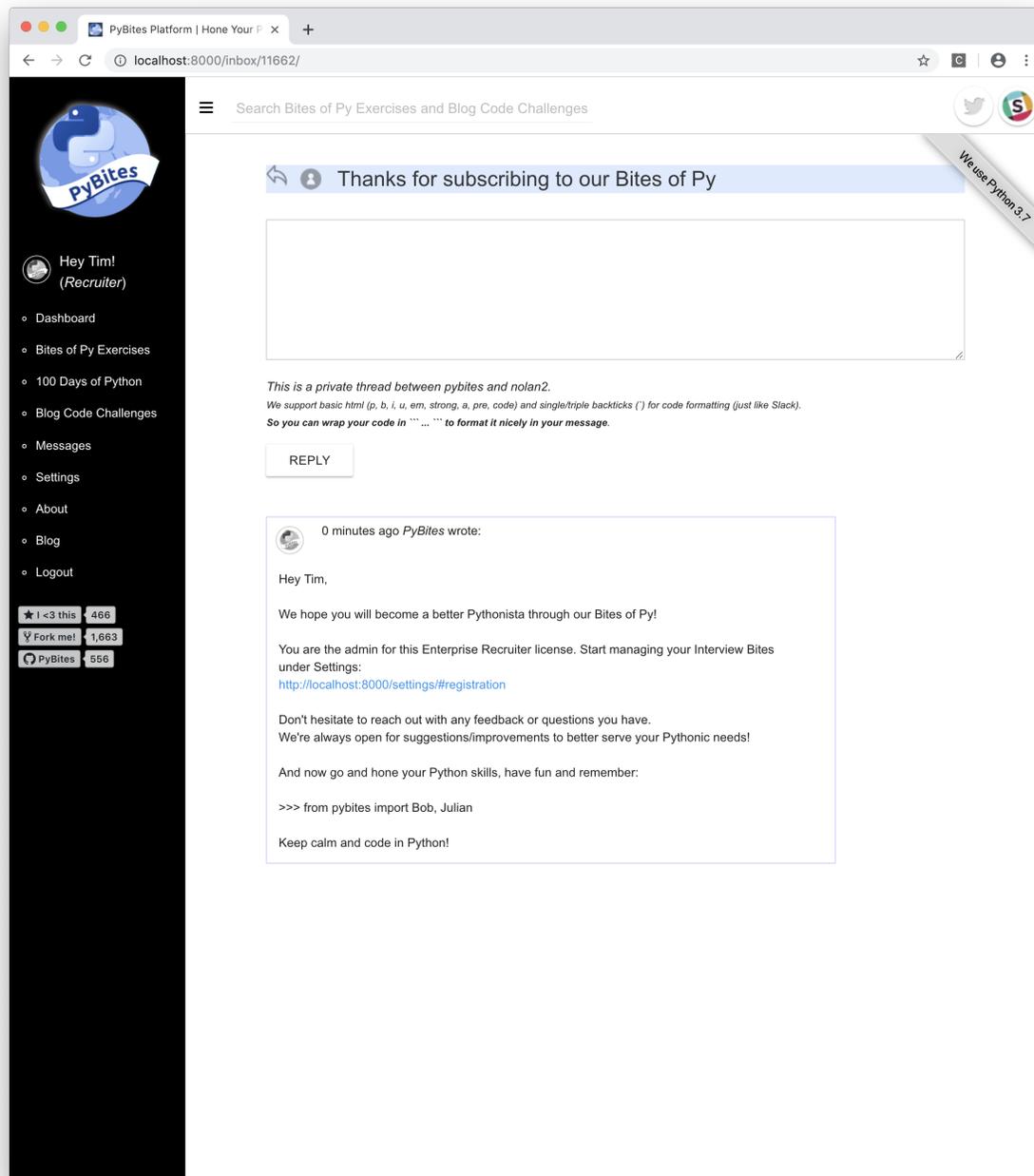
In this helper doc we have 3 users: Tim is a recruiter and Sara and Ramit are interviewees. We will show you how to set up an interview Bite as the recruiter; send out private interview links (tokens) to the interviewees and watch their submissions.

### 5.3 Register your license key

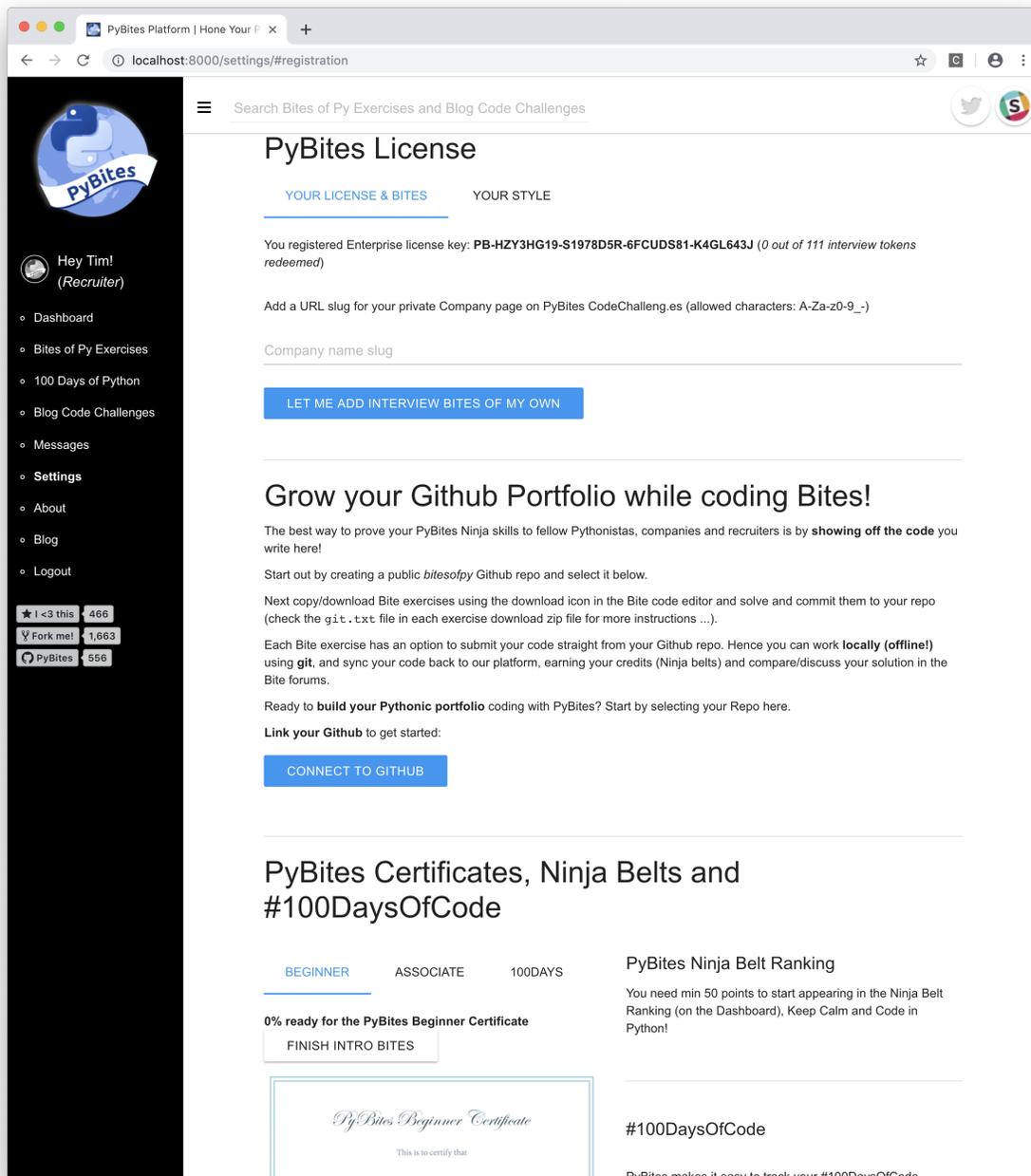
After buying a license key from our [Enterprise pricing page](#) register it under [Settings](#) or with the direct purchase link provided in the purchase email (e.g. [https://codechalleng.es/purchase/access\\_code/PB-U8N435EH-PG65PW87-IXPWQG5T-898XSZI4](https://codechalleng.es/purchase/access_code/PB-U8N435EH-PG65PW87-IXPWQG5T-898XSZI4)).



You'll then receive a confirmation/welcome message in the Messages section of the platform:



Click the settings link in the message and you'll be presented with the following screen showing your license keys and the amount of interview tokens purchased:

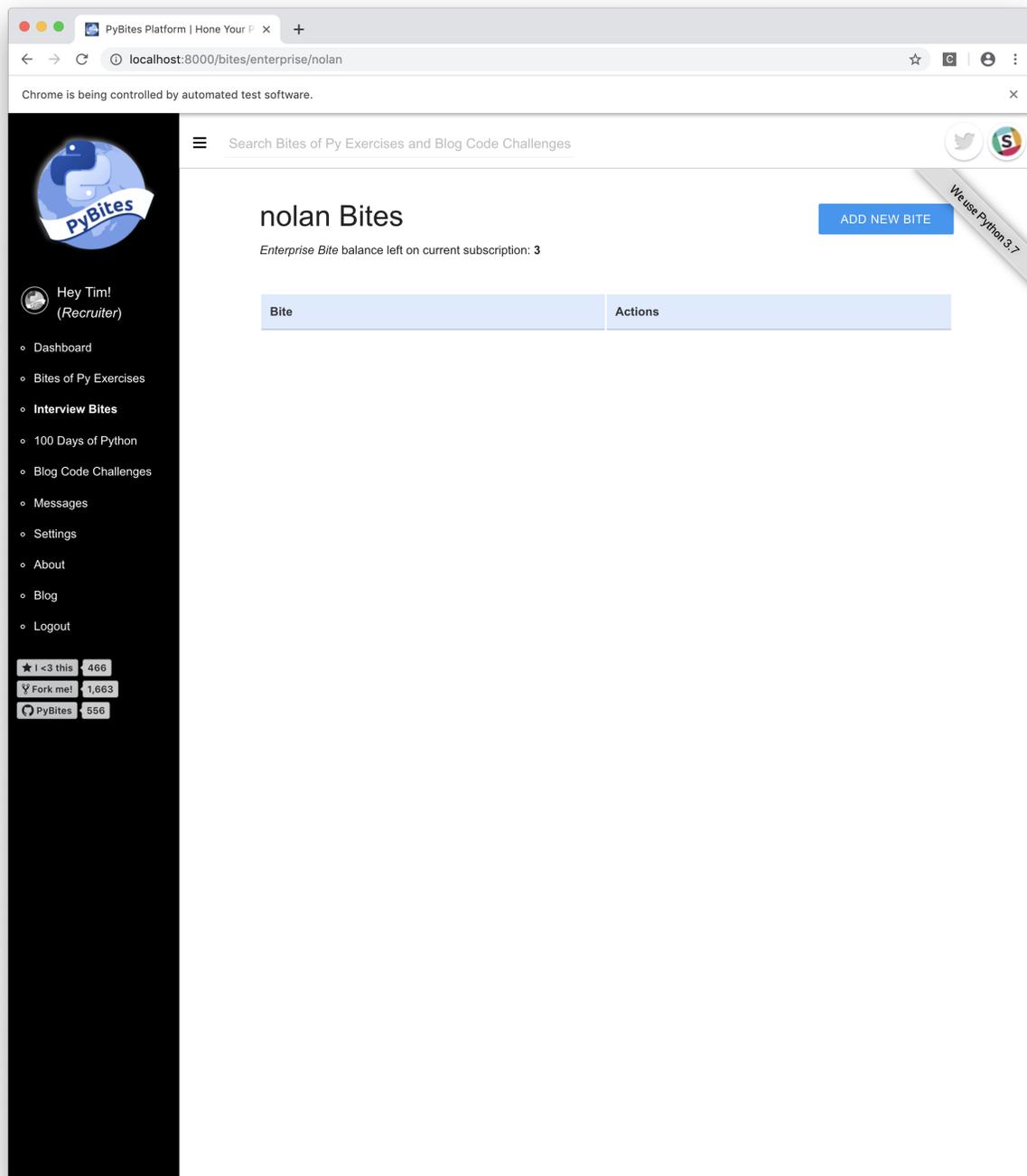


## 5.4 Your own space on PyBites

At this point you need to set a *company slug* for your Interview Bites. Consider it your private corner in the PyBites universe. As we're movie fans let's call our slug *nolan*:

The screenshot shows the 'PyBites License' settings page. The browser address bar indicates the URL is localhost:8000/settings/#registration. The page has a dark sidebar on the left with a navigation menu and social media stats. The main content area is titled 'PyBites License' and includes a search bar, a notification, and several sections: 'YOUR LICENSE & BITES' with a license key and company slug, 'MANAGE YOUR INTERVIEW BITES' button, 'Grow your Github Portfolio while coding Bites!' with instructions on how to link a Github repo, and 'PyBites Certificates, Ninja Belts and #100DaysOfCode' with progress indicators for Beginner, Associate, and 100Days levels.

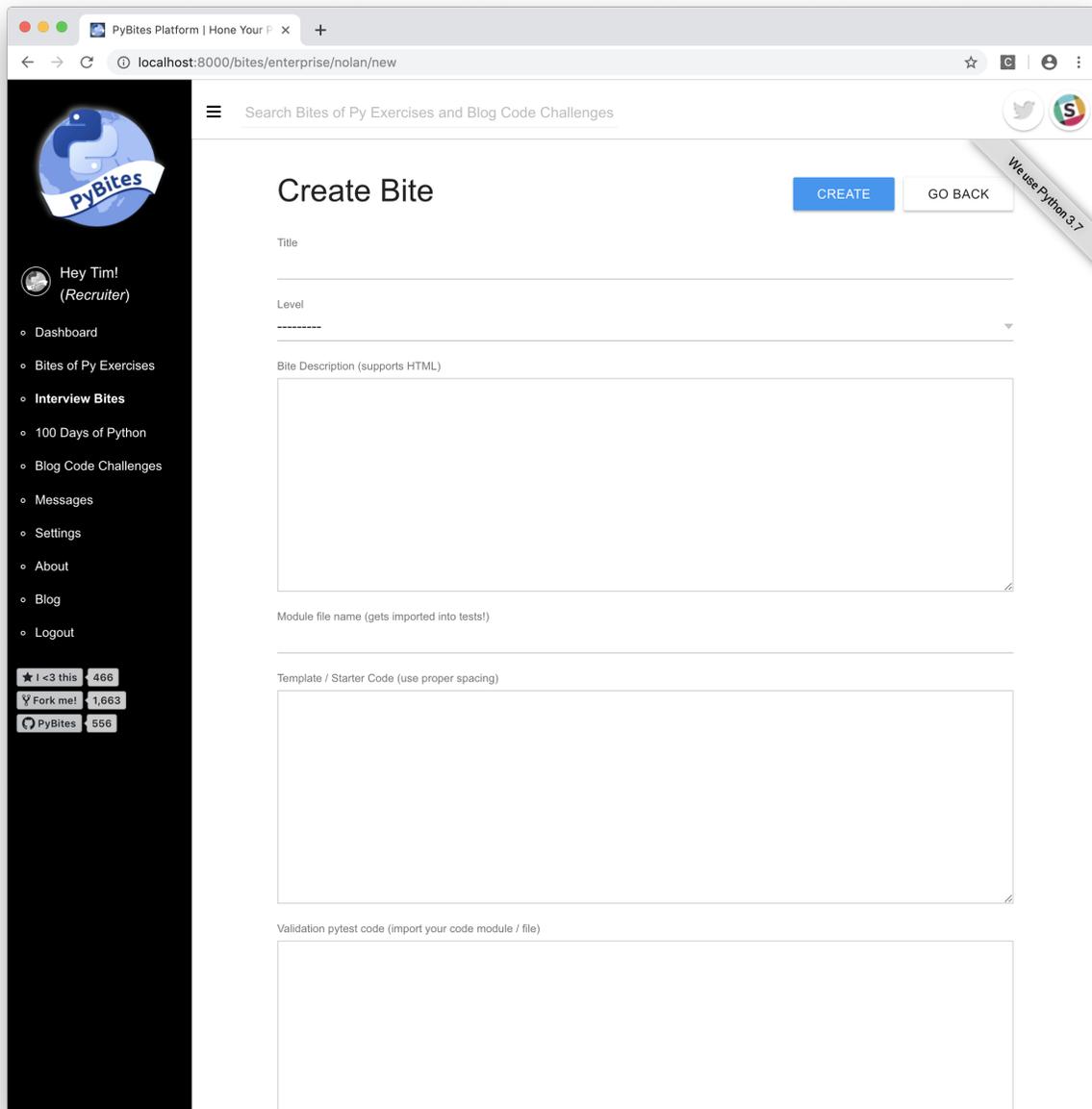
At this point you'll see your private *Interview Bites* link in the sidebar. Click on it or on the *Manage your interview bites* button while still in Settings:

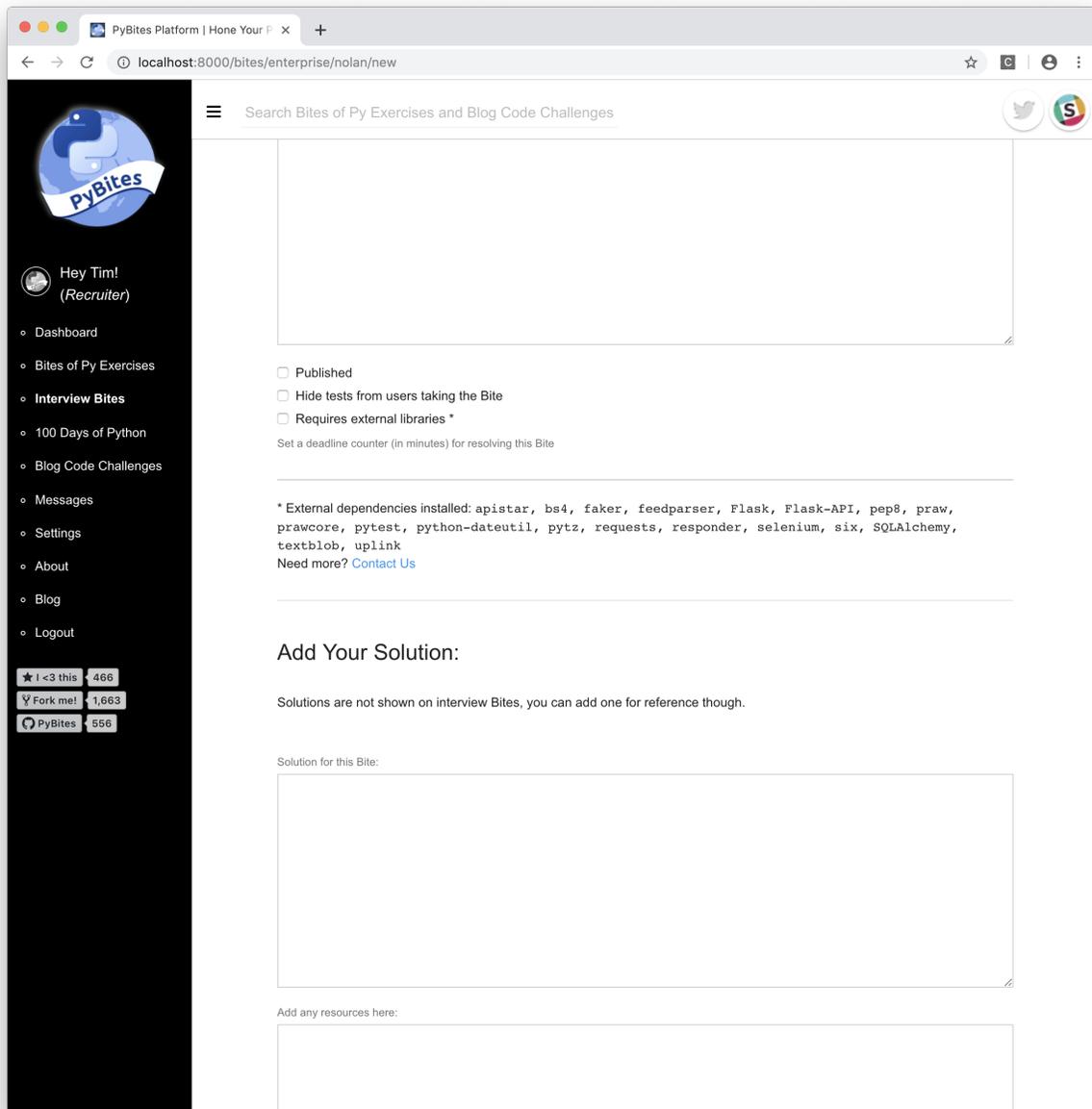


You can see the amount of Interview Bites available on your account (currently 1 for each 50 purchased tokens).

## 5.5 Add Interview Bites

You can add a new Interview Bite by clicking the *Add new Bite* button:





As you can see we support quite a few external libraries for you to use. You can also hide the tests to potentially make it more difficult and (exciting!) you can set a deadline in minutes to make it even more tough!

## 5.6 Copy our Bites

If you don't want to write your own Bites you're able to use our existing Bites for interviewing purposes.

After the previous setup you should see a *Copy this Bite* button on individual Bite pages. (Note: Intro Bites are not included)

The screenshot shows the PyBites Platform interface for the 'Bite 2. Regex Fun' challenge. The page is displayed in a browser window at localhost:8000/bites/2/. The interface includes a sidebar with navigation links, a search bar, and a main content area with instructions and a code editor. A 'COPY THIS BITE' button is visible in the top right corner.

**Bite 2. Regex Fun**

Learn Python regular expressions by writing the following 3 functions.

In each function's `docstring` we show the expected result.

The tests verify these expected results when you hit `Save + Verify`. We will do a more ample regex bite later on.

Note that for parsing HTML you would use a library of some sort, this is just to train your regex skills.

2 out of 2 users completed this Bite. Will you be Pythonista #3 to crack this Bite? It takes an average of ~44 minutes to solve this Bite (submissions < 4 hours).

**YOUR CODE (regex.py)**

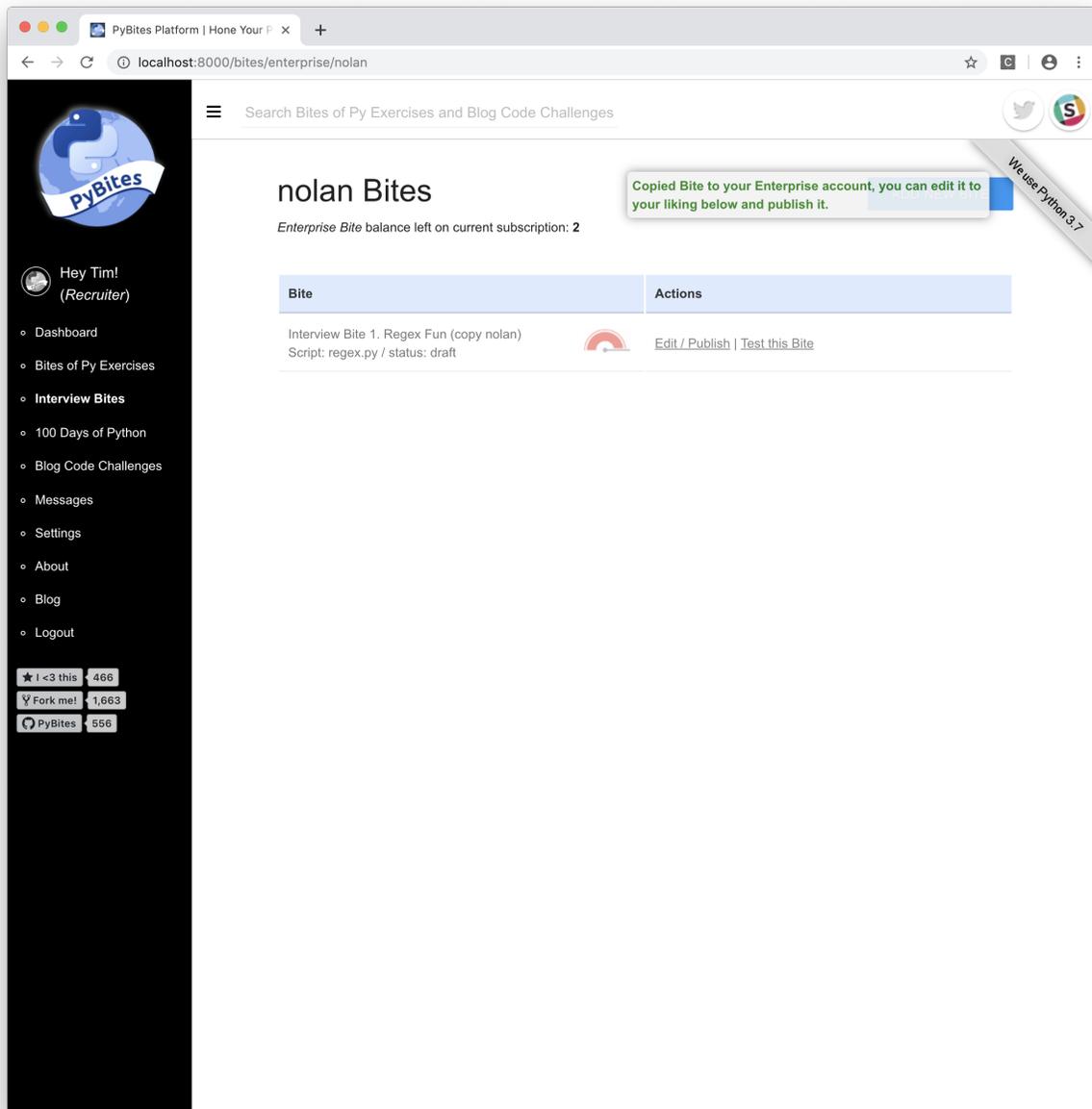
```

1 import re
2
3 def extract_course_times():
4     """Write a regular expression that returns a list of timestamps:
5     ['01:47', '32:03', '41:51', '27:48', '05:02']"""
6     flask_course = ('Introduction 1 Lecture 01:47'
7                     'The Basics 4 Lectures 32:03'
8                     'Getting Technical! 4 Lectures 41:51'
9                     'Challenge 2 Lectures 27:48'
10                    'Afterword 1 Lecture 05:02')
11     pass
12
13
14 def get_all_hashtags_and_links():
15     """Write a regular expression that returns this list:
16     ['http://pybit.es/requests-cache.html', '#python', '#APIs']"""
17     tweet = ('New PyBites article: Module of the Week - Requests-cache '
18            'for Repeated API Calls - http://pybit.es/requests-cache.html '
19            '#python #APIs')
20     pass
21
22
23 def match_first_paragraph():
24     """Write a regular expression that returns 'pybites != greedy' """
25

```

Here we've navigated to Bite 2 which has some kick-ass regex problems for the candidate to solve. Notice the "Copy this Bite" button. Click it.

The Bite is now copied to your private Interview Bites!



Click on “Edit / Publish” to make any changes. Here we update the title and toggle “Published” to on:

The screenshot shows the PyBites Platform interface in a browser window. The URL is localhost:8000/bites/enterprise/nolan/edit/148. The page title is 'The Regex Challenge'. The challenge level is 'Advanced (img: AV / score: 4)'. The description asks the user to write three functions to parse timestamps from a string. The code editor shows the implementation of these functions. The test code uses pytest to verify the functions.

**Challenge Description:**

Learn Python regular expressions by writing the following 3 functions. In each function's docstring we show the expected result. The tests verify these expected results when you hit Save + Verify. We will do a more ample regex bite later on. Note that for parsing HTML you would use a library of some sort, this is just to train your regex skills.

**Module file name (gets imported into tests!)**

```
regex.py
```

**Template / Starter Code (use proper spacing)**

```
import re

def extract_course_times():
    """Write a regular expression that returns a list of timestamps:
    ['01:47', '32:03', '41:51', '27:48', '05:02']"""
    flask_course = ('Introduction 1 Lecture 01:47'
                    'The Basics 4 Lectures 32:03'
                    'Getting Technical! 4 Lectures 41:51'
                    'Challenge 2 Lectures 27:48'
                    'Afterword 1 Lecture 05:02')
```

**Validation pytest code (import your code module / file)**

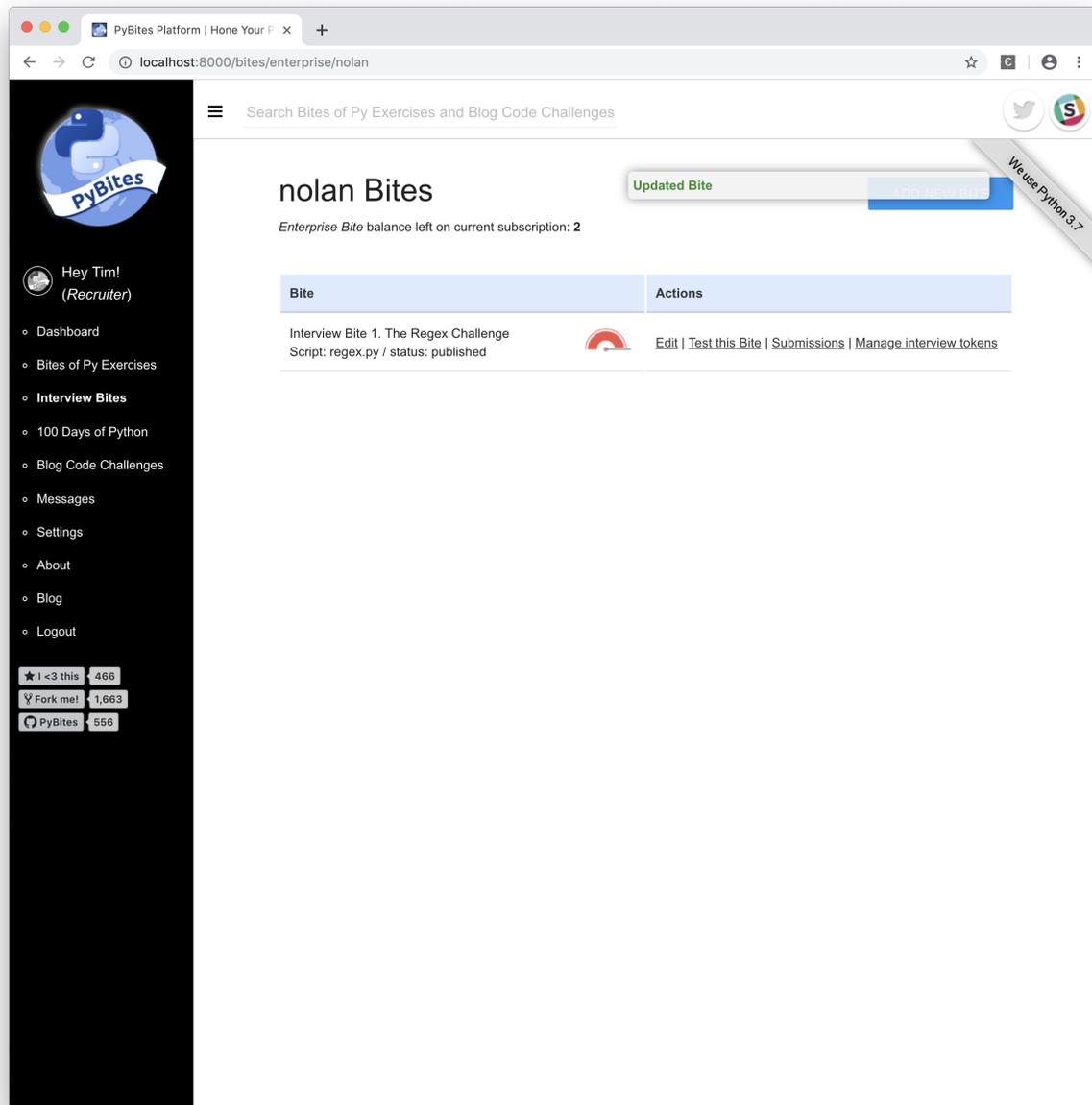
```
from regex import (extract_course_times, get_all_hashtags_and_links,
                  match_first_paragraph)

def test_extract_course_times():
    expected = ['01:47', '32:03', '41:51', '27:48', '05:02']
    assert extract_course_times() == expected

def test_get_all_hashtags_and_links():
```

Published  
 Hide tests from users taking the Bite

2 new links now appear on the Interview Bites page: *Submissions* and *Manage interview tokens*:



## 5.7 Interviewee links

Click on *Manage interview tokens* to generate private interviewee links (tokens):

The screenshot shows the PyBites Platform interface in a browser window. The URL is localhost:8000/bites/enterprise/nolan/edit/148#tokens. The page title is "Search Bites of Py Exercises and Blog Code Challenges".

**Left Sidebar:**

- PyBites logo
- Hey Tim! (Recruiter)
- Dashboard
- Bites of Py Exercises
- Interview Bites
- 100 Days of Python
- Blog Code Challenges
- Messages
- Settings
- About
- Blog
- Logout
- ★ I <3 this 466
- 🍴 Fork me! 1,663
- 🐍 PyBites 556

**Main Content Area:**

Search Bites of Py Exercises and Blog Code Challenges

```
import re

def extract_course_times():
    """Write a regular expression that returns a list of timestamps:
    ['01:47', '32:03', '41:51', '27:48', '05:02']"""
    flask_course = ('Introduction 1 Lecture 01:47'
                    'The Basics 4 Lectures 32:03'
                    'Getting Technical! 4 Lectures 41:51'
                    'Challenge 2 Lectures 27:48'
                    'Afterword 1 Lecture 05:02')
```

Add any resources here:

10 Tips to Get More out of Your Regexes  
<https://pybit.es/mastering-regex.html>

PyBites regex tag  
<https://pybit.es/tag/regex.html>

**UPDATE** **GO BACK**

### Interview Bite Tokens

Interview Bites are not publicly accessible by default. Create private token links for interviewees below. Each link is good for one interviewee coding one Bite exercise.

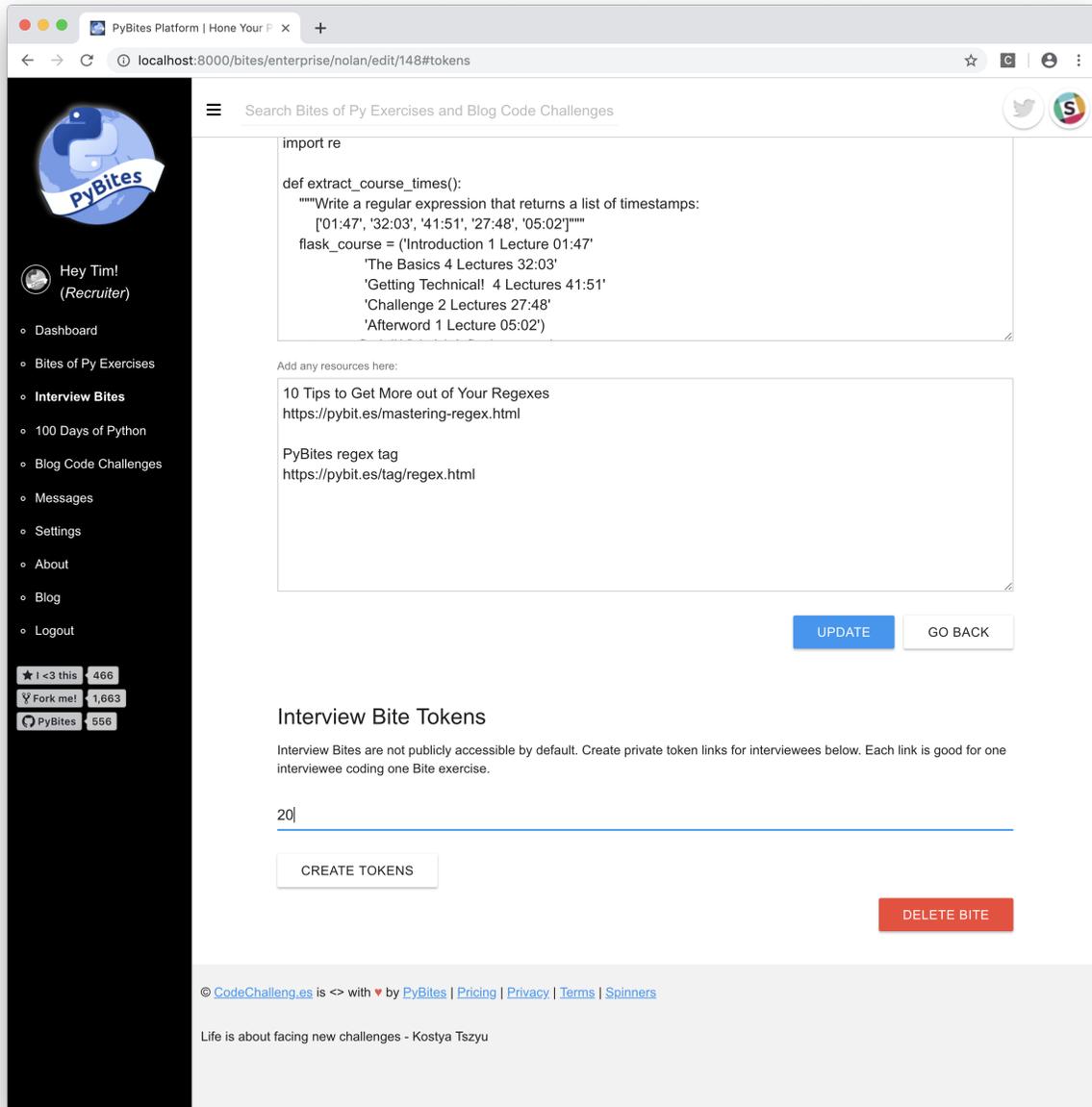
Number of tokens (111 left)

**CREATE TOKENS** **DELETE BITE**

© CodeChalleng.es is <> with ♥ by PyBites | Pricing | Privacy | Terms | Spinners

Life is about facing new challenges - Kostya Tszzy

Let's generate 20 interview links for this one:



PyBites Platform | Hone Your P x

localhost:8000/bites/enterprise/nolan/edit/148#tokens

Search Bites of Py Exercises and Blog Code Challenges

## Interview Bite Tokens

20 tokens created

Interview Bites are not publicly accessible by default. Create private token links for interviewees below. Each link is good for one interviewee coding one Bite exercise.

Number of tokens (91 left)

CREATE TOKENS

Share the following private links with the interviewees (each link is private to the user that redeems it):

Interviewee links	DELETE
<a href="http://localhost:8000/bites/interview/1TVQ460JAKU30CUZXGR8/nolan-1/">http://localhost:8000/bites/interview/1TVQ460JAKU30CUZXGR8/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/7THR2B09CU8EEJZ7G4TD/nolan-1/">http://localhost:8000/bites/interview/7THR2B09CU8EEJZ7G4TD/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/21L9UIN9EOEBBD4P89C8/nolan-1/">http://localhost:8000/bites/interview/21L9UIN9EOEBBD4P89C8/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/ZDP7S5Z9VQS2V8YS5G71/nolan-1/">http://localhost:8000/bites/interview/ZDP7S5Z9VQS2V8YS5G71/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/DOUKXGIDB3ZDZ8B2C64R/nolan-1/">http://localhost:8000/bites/interview/DOUKXGIDB3ZDZ8B2C64R/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/AKREV872PUJ4NTZ3ZT4B/nolan-1/">http://localhost:8000/bites/interview/AKREV872PUJ4NTZ3ZT4B/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/1DE1RDJBAQ3CKY5NVUGN/nolan-1/">http://localhost:8000/bites/interview/1DE1RDJBAQ3CKY5NVUGN/nolan-1/</a>	<input type="checkbox"/>
<a href="http://localhost:8000/bites/interview/QFX03SZYTZZRAEWG78LI/nolan-1/">http://localhost:8000/bites/interview/QFX03SZYTZZRAEWG78LI/nolan-1/</a>	<input type="checkbox"/>

Hey Tim!  
(Recruiter)

- Dashboard
- Bites of Py Exercises
- Interview Bites
- 100 Days of Python
- Blog Code Challenges
- Messages
- Settings
- About
- Blog
- Logout

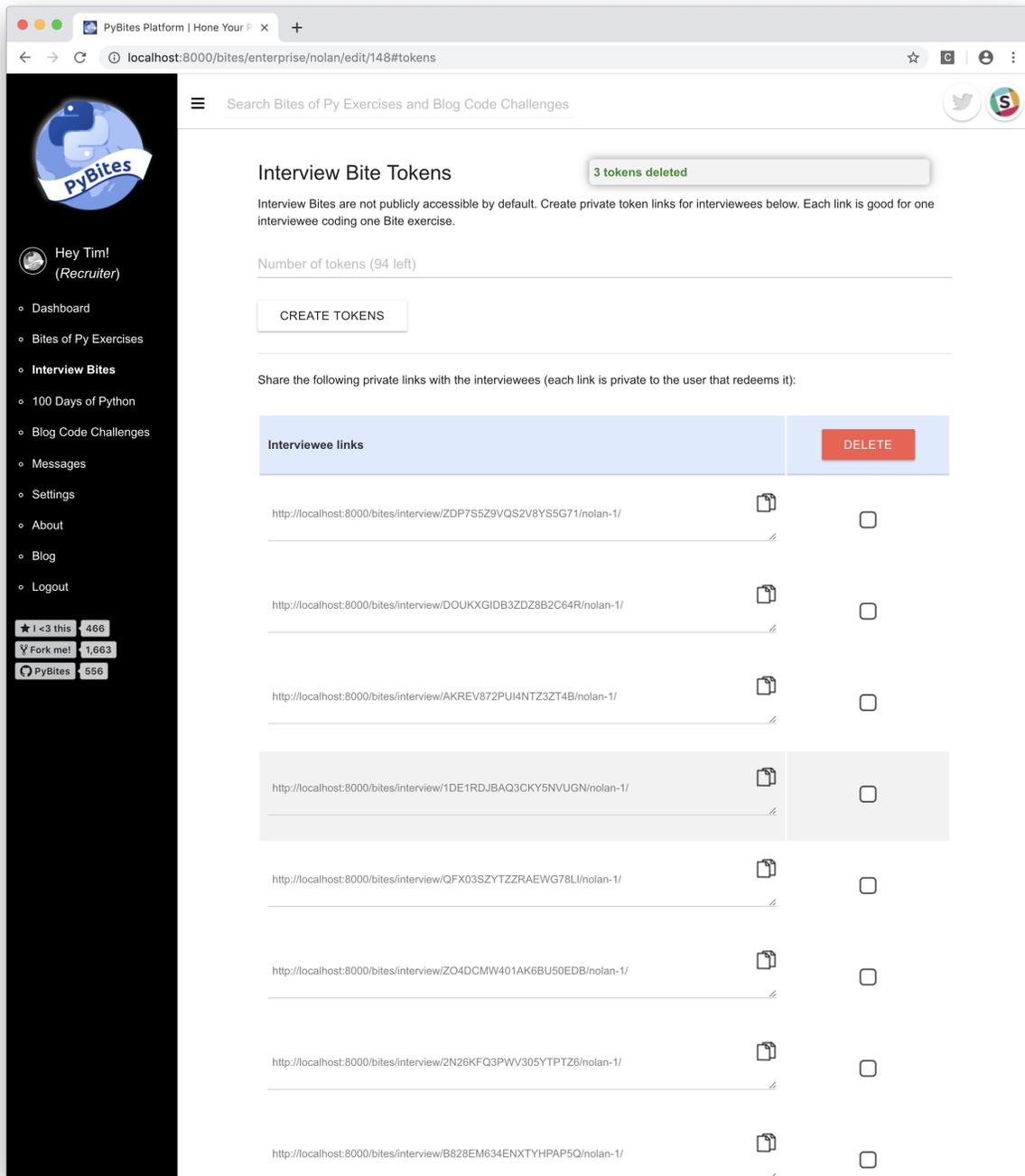
★ I <3 this 466

Fork me! 1,663

PyBites 556

The balance goes down by 20. Token spots can be recovered by deleting them. **Tokens are only consumed when interviewees redeem the links.**

For example, here we delete 3 tokens, recovering some of the balance:



## 5.8 Interviewing people

Let's send Ramit and Sara an interview link. Click the copy button on the first two token rows. Sara enters the interview token in her browser and is presented with our typical Bite interface:

PyBites Platform | Interview Bite 148

localhost:8000/bites/148/code/

Chrome is being controlled by automated test software.

START YOUR FREE TRIAL

Interview Bite 1. The Regexp Challenge Successfully redeemed interview token

Learn Python regular expressions by writing the following 3 functions.

In each function's `docstring` we show the expected result.

The tests verify these expected results when you hit `Save + Verify`. We will do a more ample regexp bite later on.

Note that for parsing HTML you would use a library of some sort, this is just to train your regexp skills.

SAVE AND RUN TESTS (CTRL + ⌘ + ↵)    RUN CODE ONLY (CTRL + ↵)

YOUR CODE (regex.py)    TESTS (test\_regex.py)

```

1 import re
2
3 def extract_course_times():
4     """Write a regular expression that returns a list of timestamps:
5     ['01:47', '32:03', '41:51', '27:48', '05:02']"""
6     flask_course = ('Introduction 1 Lecture 01:47'
7                    'The Basics 4 Lectures 32:03'
8                    'Getting Technical! 4 Lectures 41:51'
9                    'Challenge 2 Lectures 27:48'
10                   'Afterword 1 Lecture 05:02')
11     pass
12
13
14 def get_all_hashtags_and_links():
15     """Write a regular expression that returns this list:
16     ['http://pybit.es/requests-cache.html', '#python', '#APIs']"""
17     tweet = ('New PyBites article: Module of the Week - Requests-cache '
18            'for Repeated API Calls - http://pybit.es/requests-cache.html '
19            '#python #APIs')
20     pass
21
22
23 def match_first_paragraph():
24     """Write a regular expression that returns 'pybites != greedy' """
25     html = ('<p>pybites != greedy</p>')

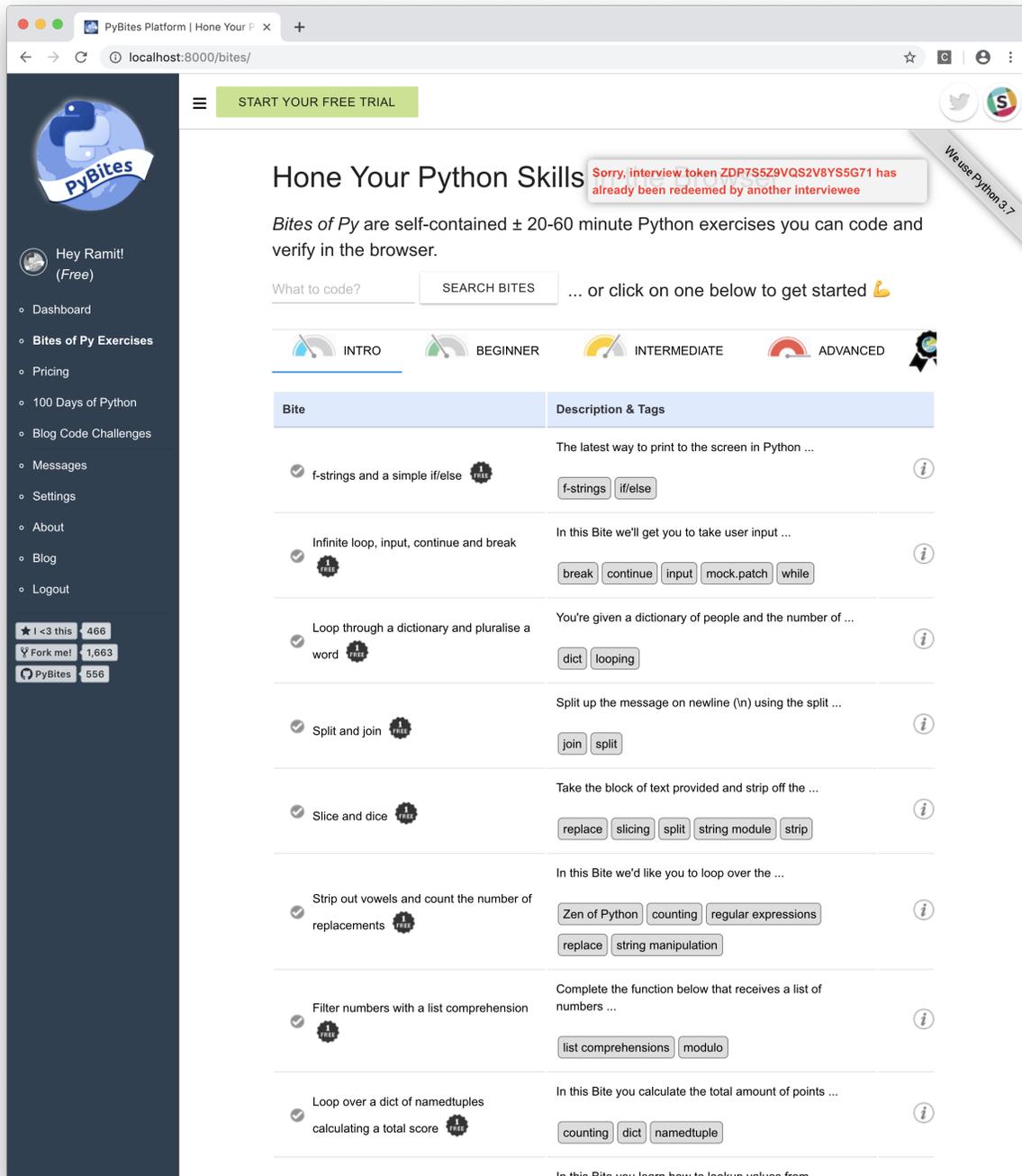
```

Hit **SAVE AND RUN TESTS** as many times as you need. You can also use the **RUN CODE ONLY** console option. Once you pass the tests, it becomes visible to the interviewer.

Not happy with your code? Just click **TRY AGAIN > TAKE THIS BITE AGAIN** - visible on a *completed* Bite - and tweak your code. Then share your updated code hitting **SAVE AND RUN TESTS** again.

Keep Calm and Code More Python ... good luck!

To show you that each token is unique to an interviewee here is what happens when Ramit redeems the same link:



Let's compare this with the timer view. Tim has set 60 minutes on the Bite. Now, let's get Ramit to open the Bite:

The screenshot shows a web browser window at localhost:8000/bites/148/code/. The page title is "Interview Bite 1. Regex Fun (copy no)". A green banner at the top says "START YOUR FREE TRIAL". The user profile is "Hey Ramit! (Free)". The left sidebar contains navigation links: Dashboard, Bites of Py Exercises, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, and Logout. Social media stats are shown: 466 likes, 1,663 forks, and 556 PyBites.

The main content area displays the bite title and instructions. A green notification says "Successfully redeemed interview token" and another says "Note you have 60 minutes to solve this Bite, good luck!". The instructions state: "Learn Python regular expressions by writing the following 3 functions. In each function's docstring we show the expected result. The tests verify these expected results when you hit Save + Verify. We will do a more ample regex bite later on. Note that for parsing HTML you would use a library of some sort, this is just to train your regex skills."

Buttons for "SAVE AND RUN TESTS (CTRL + ⌘ + ↵)" and "RUN CODE ONLY (CTRL + ↵)" are visible. Below the buttons, the code editor shows the following Python code:

```

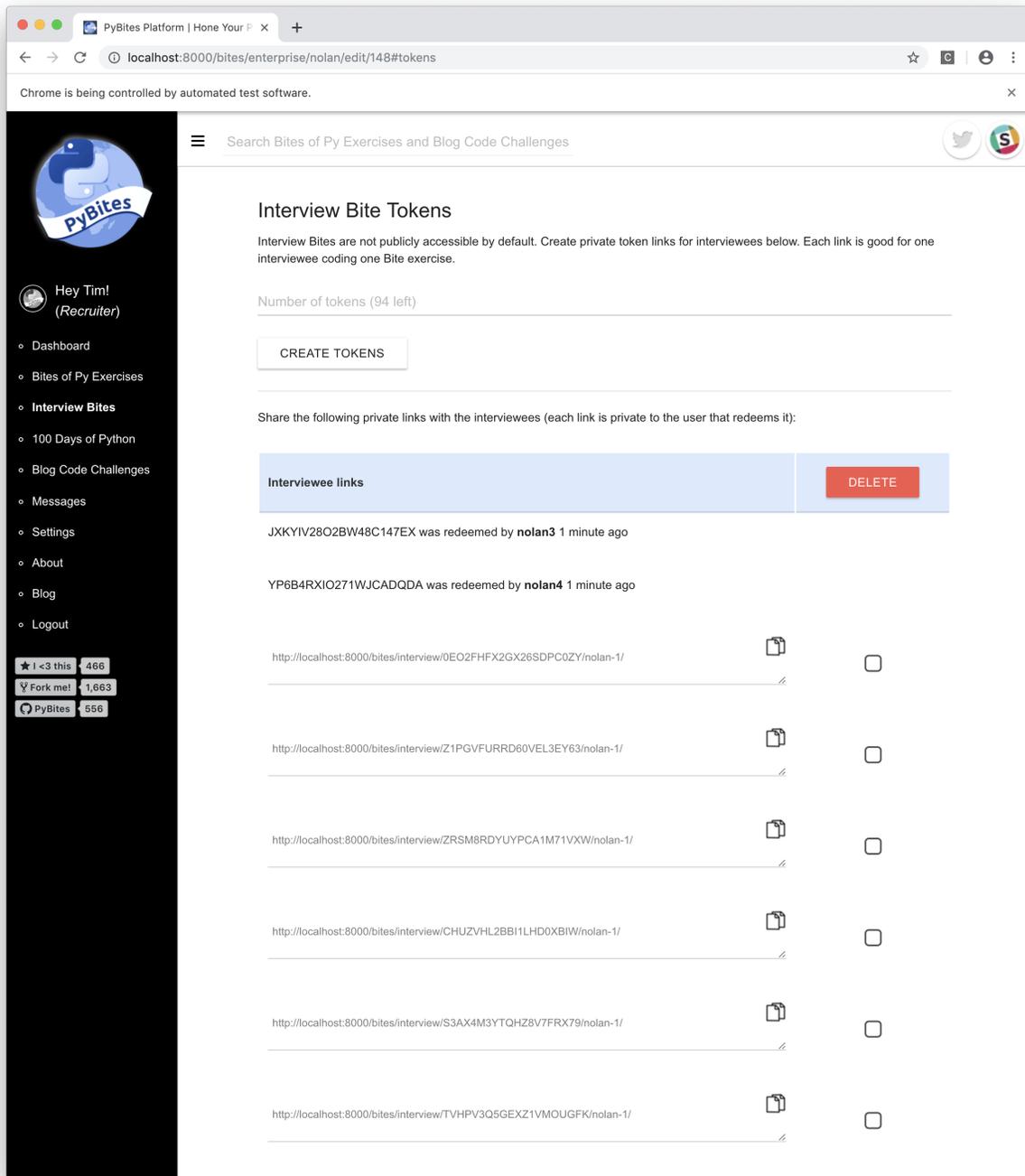
1 import re
2
3 def extract_course_times():
4     """Write a regular expression that returns a list of timestamps:
5     ['01:47', '32:03', '41:51', '27:48', '05:02']"""
6     flask_course = ('Introduction 1 Lecture 01:47'
7                    'The Basics 4 Lectures 32:03'
8                    'Getting Technical! 4 Lectures 41:51'
9                    'Challenge 2 Lectures 27:48'
10                   'Afterword 1 Lecture 05:02')
11     pass
12
13
14 def get_all_hashtags_and_links():
15     """Write a regular expression that returns this list:
16     ['http://pybit.es/requests-cache.html', '#python', '#APIs']"""
17     tweet = ('New PyBites article: Module of the Week - Requests-cache '
18            'for Repeated API Calls - http://pybit.es/requests-cache.html '
19            '#python #APIs')
20     pass
21
22
23 def match_first_paragraph():
24     """Write a regular expression that returns 'pybites != greedy' """
25     html = (<p>pybites != greedy</p>

```

At the bottom of the code editor, a yellow box contains the following text: "Hit SAVE AND RUN TESTS as many times as you need. You can also use the RUN CODE ONLY console option. Once you pass the tests, it becomes visible to the interviewer. Not happy with your code? Just click TRY AGAIN > TAKE THIS BITE AGAIN - visible on a completed Bite - and tweak your code. Then share your updated code hitting SAVE AND RUN TESTS again. Keep Calm and Code More Python ... good luck!"

He now has a timer!

Refresh the page as Tim (the Recruiter) and you can see the tokens being redeemed:



## 5.9 Interview submissions

Fast forward a bit, this is what the Interview Bites page looks like after Sara and Ramit submit some code:

The screenshot shows the 'Code Submissions for The Regex Challenge' page. The interface includes a sidebar with navigation options like 'Dashboard', 'Bites of Py Exercises', and 'Interview Bites'. The main content area features a table with two columns: 'User / stats' and 'Final code and last 5 revisions'. The first submission is by 'nolan4 (Ramit Interviewee2)' with a score of 0, 4 total saves, and a duration of 1:08:56. The second submission is by 'nolan3 (Sara Interviewee1)' with a score of 0. The 'Final code and last 5 revisions' column shows Python code for three functions: `extract_course_times()`, `get_all_hashtags_and_links()`, and `match_first_paragraph()`. A 'DOWNLOAD DATA' button is located in the top right corner of the submission area.

You can see their final code and last submits. Interview Bites are not scored at this point but you see some emojis if a timer was set and the deadline was hit or missed.

## 5.10 Download interview data

There's also a download feature to ensure you don't lose any data upon deleting Interview Bites. This will be necessary to free up Bite slots on your license (currently you get 1 for each 50 purchased interview tokens).

To download a zip file with all the interviewee's submissions just click the button on the Bite:

```
2. bash
[bobbelderbos@imac Downloads]$ unzip -l pybytes_interview_data_bite_nolan-1.zip
Archive:  pybytes_interview_data_bite_nolan-1.zip
 Length      Date    Time    Name
-----
  128      06-05-2019  11:13   README.md
   80      06-05-2019  11:13   nolan4/stats.txt
 1146      06-05-2019  11:13   nolan4/final_code.py
 1022      06-05-2019  11:13   nolan4/submissions/rev1_20190605-111006.py
 1064      06-05-2019  11:13   nolan4/submissions/rev2_20190605-111056.py
 1103      06-05-2019  11:13   nolan4/submissions/rev3_20190605-111126.py
 1146      06-05-2019  11:13   nolan4/submissions/rev4_20190605-111140.py
   79      06-05-2019  11:13   nolan3/stats.txt
 1146      06-05-2019  11:13   nolan3/final_code.py
 1022      06-05-2019  11:13   nolan3/submissions/rev1_20190605-110653.py
-----
  7936                                 10 files
[bobbelderbos@imac Downloads]$
```



## 5.11 Feedback

We hope this new way of interviewing (pre-screening) makes the hiring process more agile and will save you a lot of time. As always we're very open to your feedback to make the platform better, just ping us [on the platform](#).

## CHAPTER 6

---

### Using the Enterprise Tier

---

Placeholder