
PyBites Platform

Release 0.0.1

Jul 26, 2023

1	Welcome to the PyBites CodeChallenges Platform	1
2	Platform Administration	3
3	How to Solve a Bite	5
4	Premium Tier	7
5	Recruiting talent via the PyBites CodeChallenges Platform	9
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
6	Using the Enterprise Tier	31

CHAPTER 1

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](#).

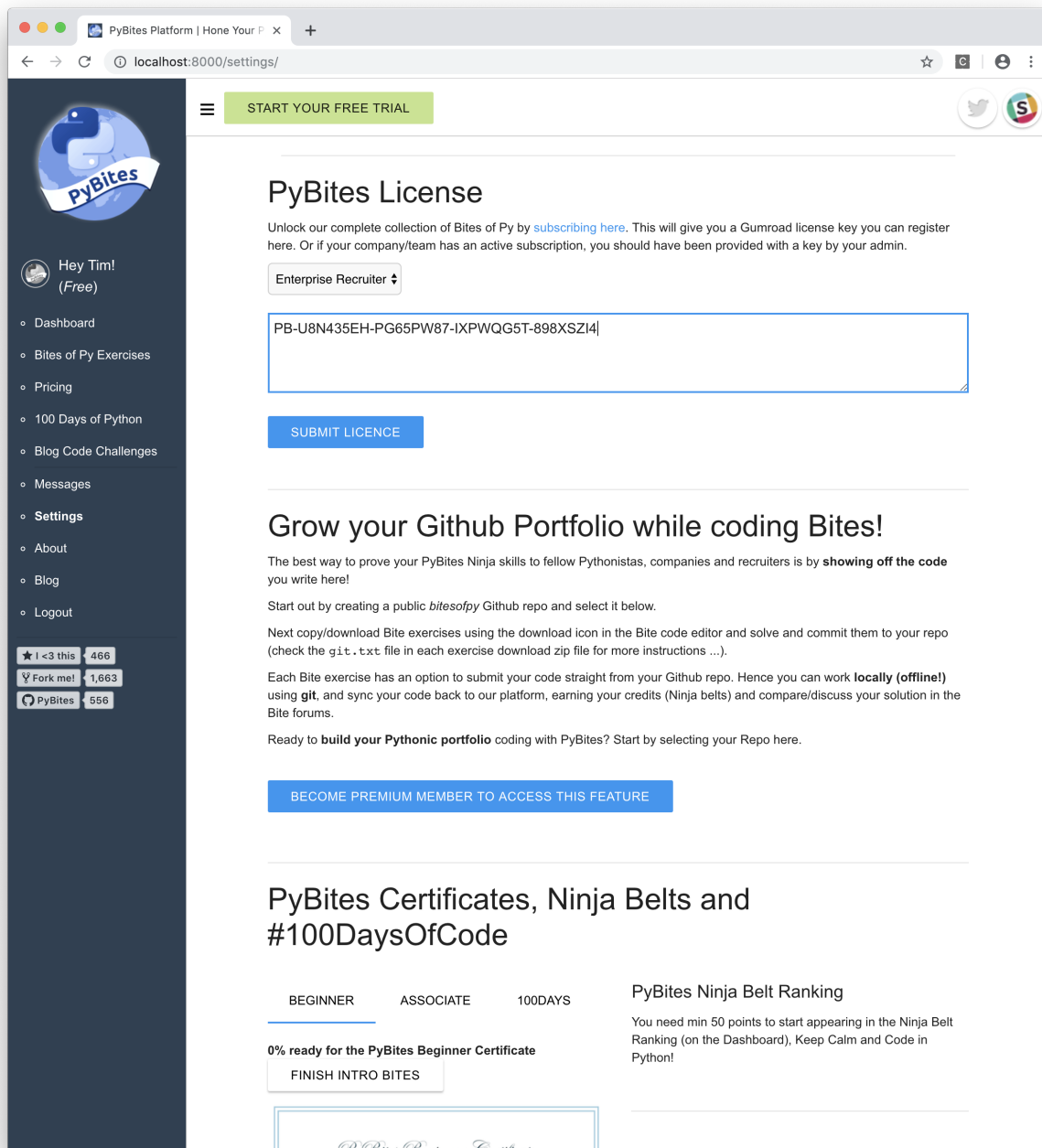
5.2 Pre-requisite

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.

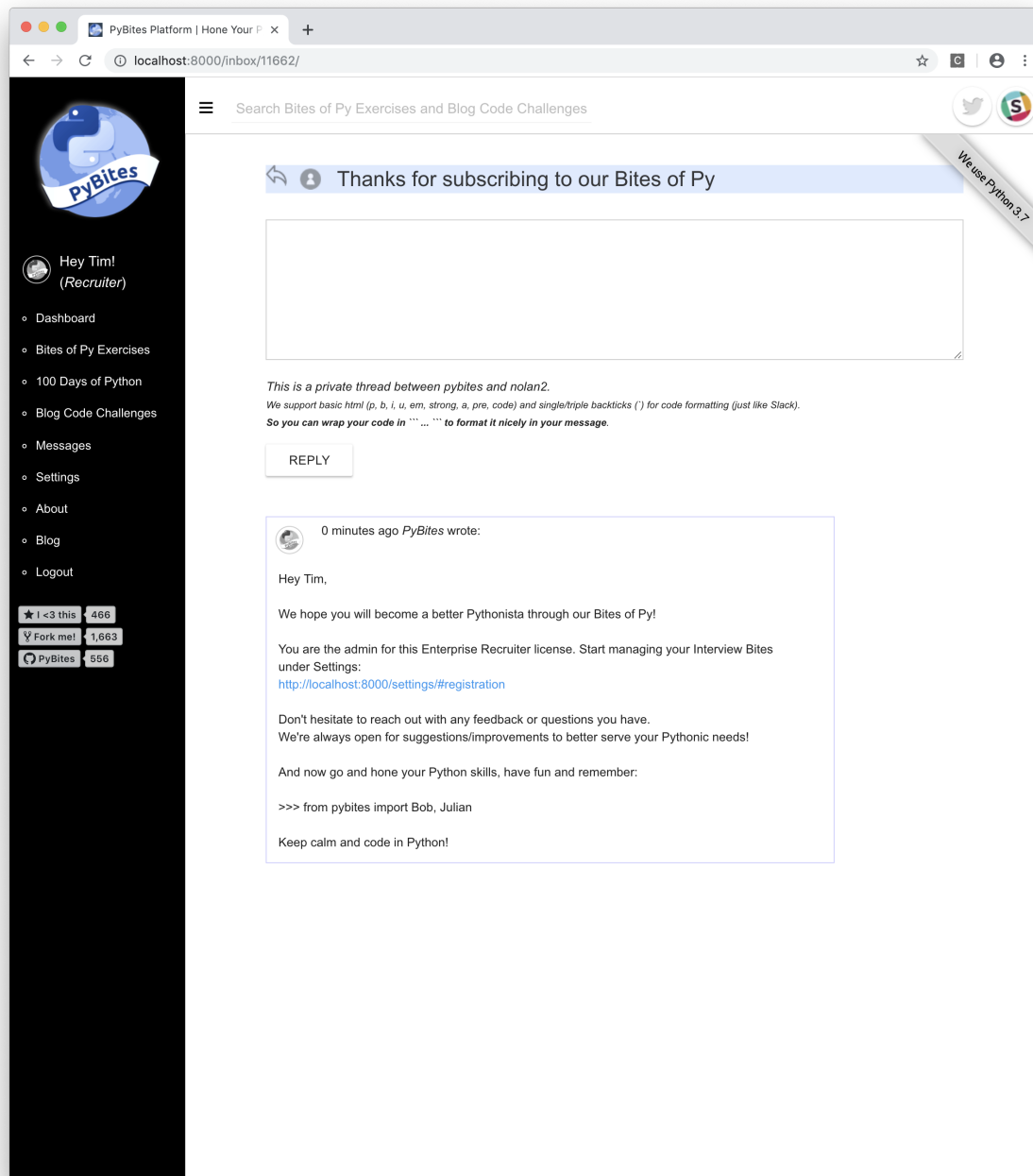
In this helper doc we have 3 users: Tim is a recruiter and Sara and Ramit are interviewees. We will show you how to set up an interview Bite as the recruiter; send out private interview links (tokens) to the interviewees and watch their submissions.

5.3 Register your license key

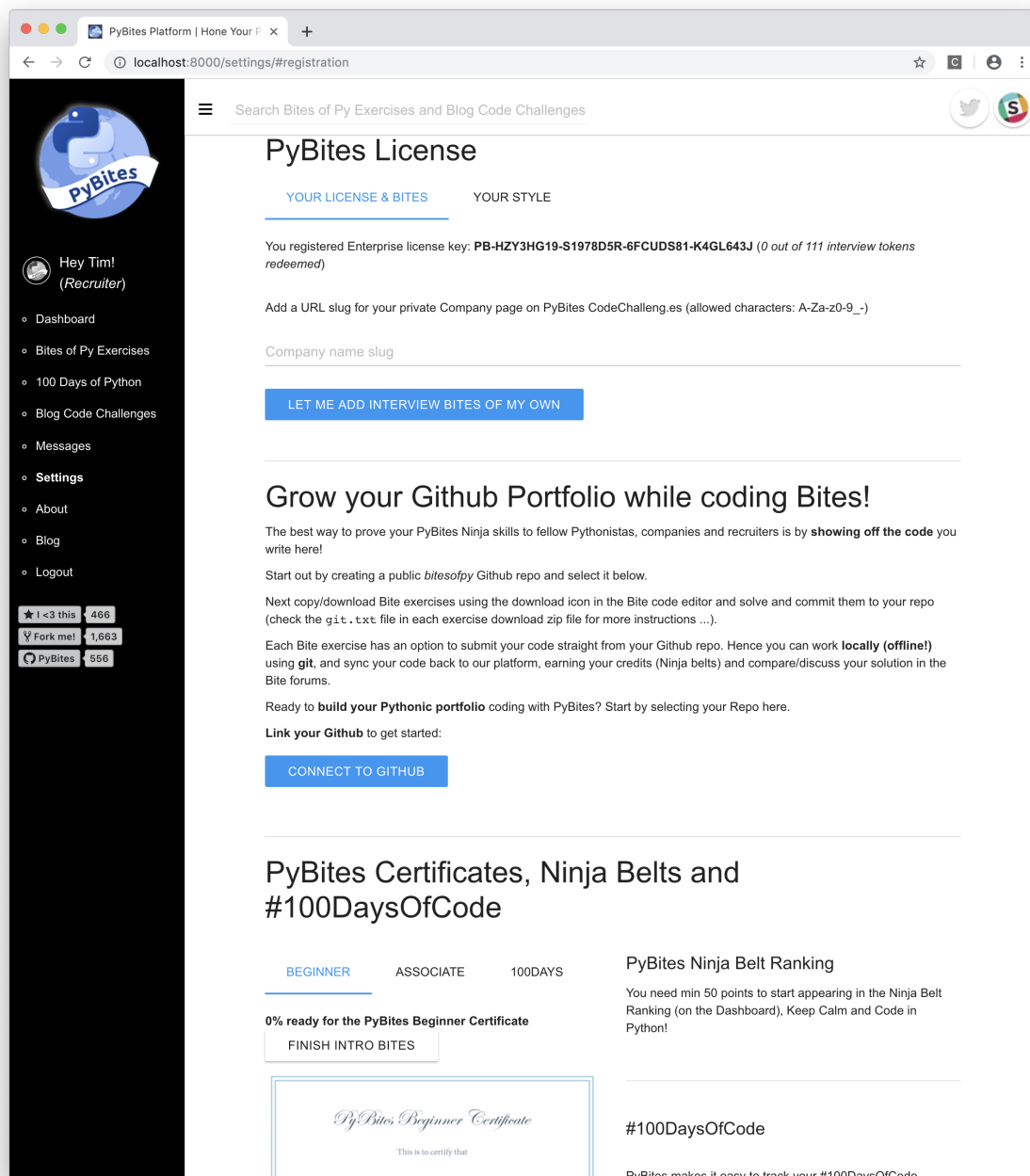
After buying a license key from our [Enterprise pricing page](#) register it under [Settings](#) or with the direct purchase link provided in the purchase email (e.g. https://codechalleng.es/purchase/access_code/PB-U8N435EH-PG65PW87-IXPWQG5T-898XSZI4).



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

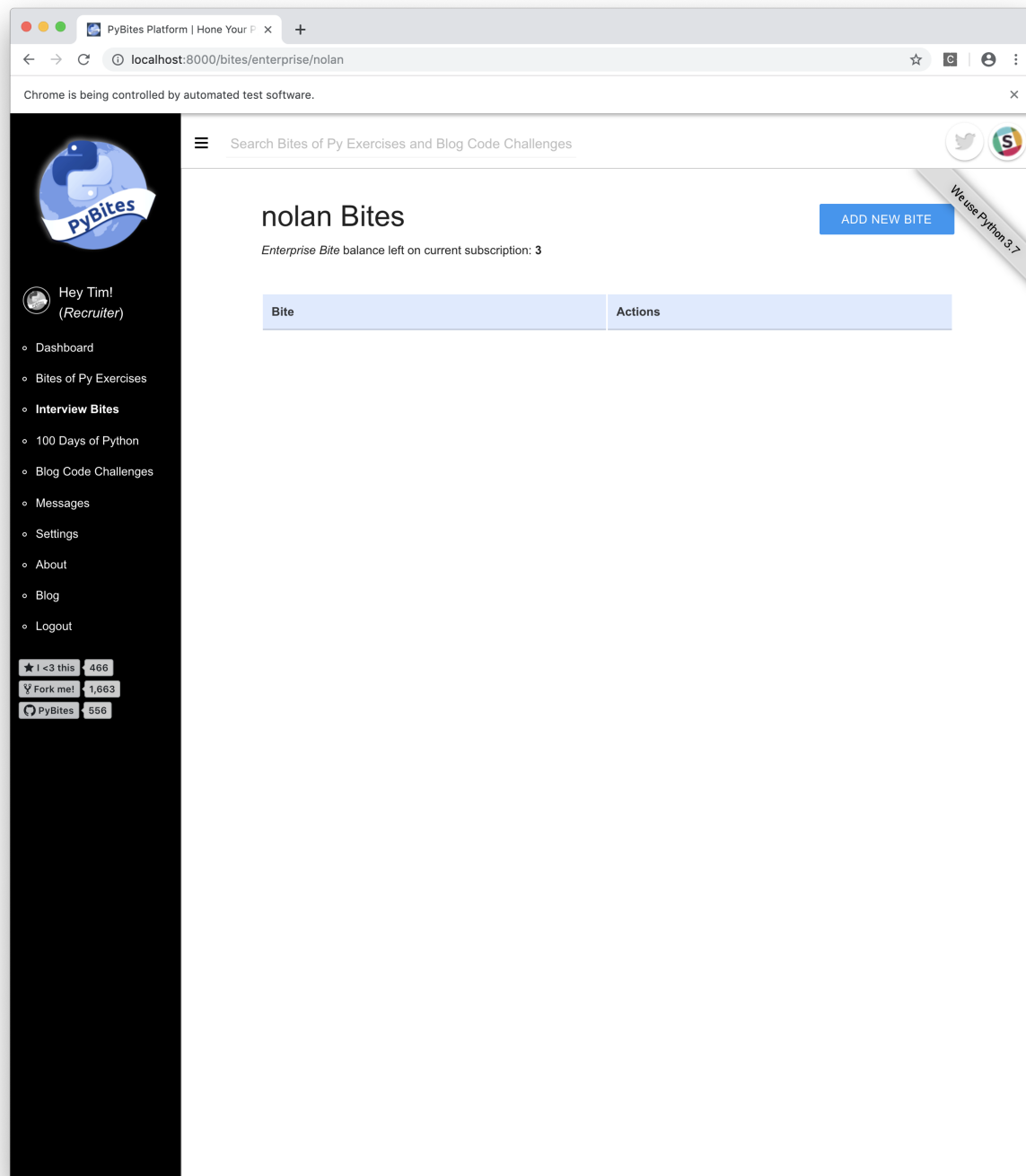
At this point you need to set a *company slug* for your Interview Bites. Consider it your private corner in the PyBites universe. As we're movie fans let's call our slug *nolan*:

The screenshot shows the PyBites Platform settings page for a user named Tim (Recruiter). The page is titled "PyBites License" and has two tabs: "YOUR LICENSE & BITES" and "YOUR STYLE". A green notification box says "Thanks, you can now add your own Interview Bites!". The license information shows a registered Enterprise license key: PB-K5XRNSJA-M1A8UWMS-XXYJLYG3-XW0F95OU (0 out of 111 interview tokens redeemed). The company slug is "nolan" and the full URL is "http://localhost:8000/bites/enterprise/nolan". There is a button "MANAGE YOUR INTERVIEW BITES".

Below the license section, there is a section titled "Grow your Github Portfolio while coding Bites!". It explains that the best way to prove PyBites Ninja skills is by showing off the code you write here. It suggests starting out by creating a public `bitesofpy` Github repo and selecting it below. It also mentions that each Bite exercise has an option to submit your code straight from your Github repo, earning credits (Ninja belts) and compare/discuss your solution in the Bite forums. A button "CONNECT TO GITHUB" is present.

The bottom section is titled "PyBites Certificates, Ninja Belts and #100DaysOfCode". It has three tabs: "BEGINNER", "ASSOCIATE", and "100DAYS". The "BEGINNER" tab is active, showing "0% ready for the PyBites Beginner Certificate" and a button "FINISH INTRO BITES". Below this is a placeholder for a "PyBites Beginner Certificate". To the right, there is a section for "PyBites Ninja Belt Ranking" which states that you need a minimum of 50 points to start appearing in the ranking. Below that is a section for "#100DaysOfCode" which mentions that PyBites makes it easy to track your progress and provides a link to "try it out for free".

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).

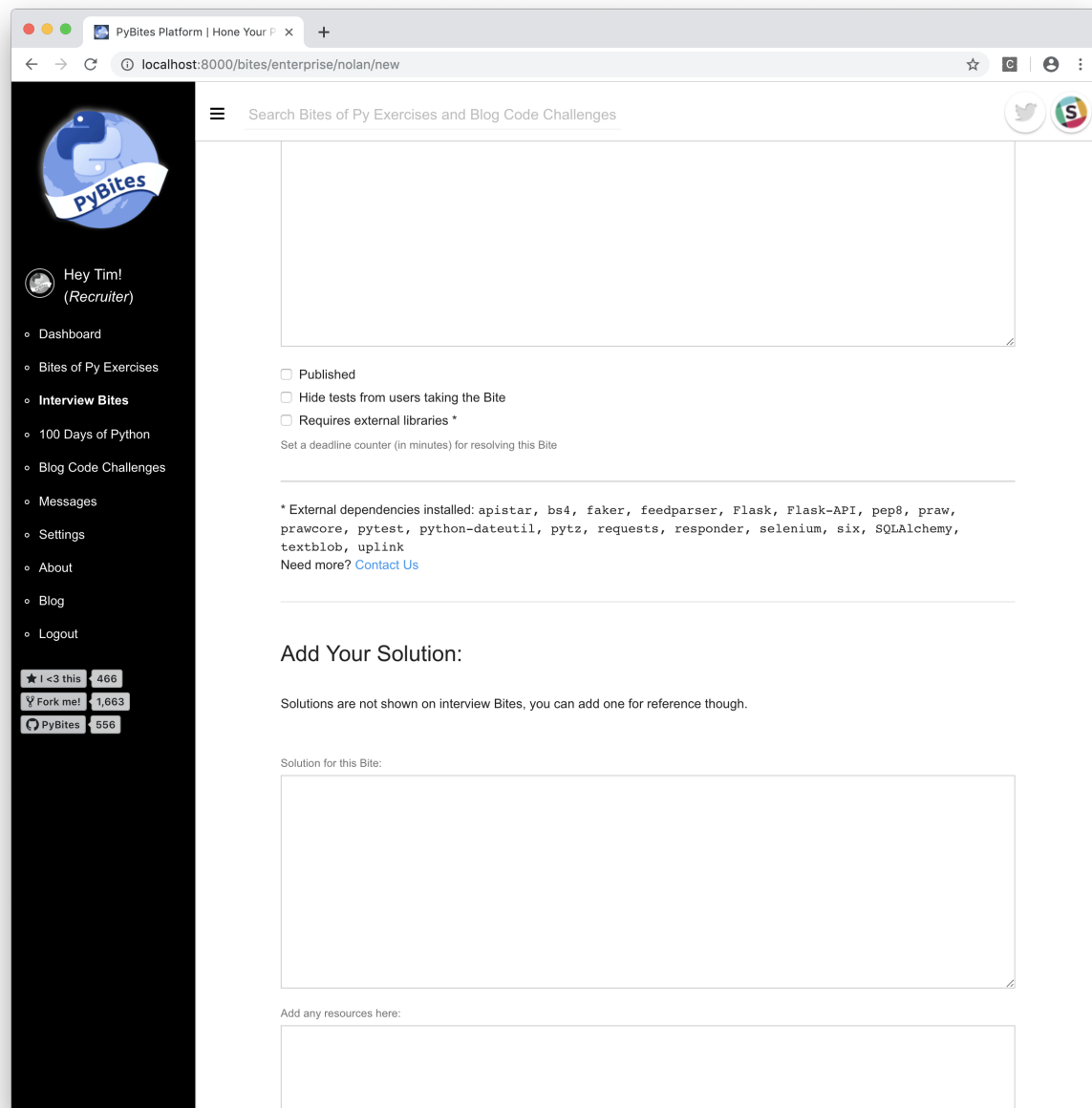
5.5 Add Interview Bites

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

The screenshot shows a web browser window with the URL `localhost:8000/bites/enterprise/nolan/new`. The page is titled "Create Bite" and features a sidebar on the left with the PyBites logo and a navigation menu. The main content area contains a form with the following fields:

- Title**: A text input field.
- Level**: A dropdown menu with a "-----" placeholder.
- Bite Description (supports HTML)**: A large text area for the description.
- Module file name (gets imported into tests!)**: A text input field.
- Template / Starter Code (use proper spacing)**: A large text area for the starter code.
- Validation pytest code (import your code module / file)**: A large text area for validation code.

At the top right of the form, there are two buttons: "CREATE" (blue) and "GO BACK" (white). A diagonal banner in the top right corner reads "We use Python 3.7". The sidebar on the left includes the PyBites logo, a user profile for "Hey Tim! (Recruiter)", and a list of navigation links: Dashboard, Bites of Py Exercises, Interview Bites, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, and Logout. At the bottom of the sidebar, there are three social media links: "I <3 this" (466), "Fork me!" (1,663), and "PyBites" (556).



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 a web browser at localhost:8000/bites/2/. The page title is 'Bite 2. Regex Fun'. The left sidebar contains a navigation menu with items like Dashboard, Bites of Py Exercises, Interview Bites, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, and Logout. The main content area has a search bar and a description of the bite: '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.' There are buttons for 'findall', 're', 'regular expressions', and 'COPY THIS BITE'. A badge indicates '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)'. Below the description are buttons for 'SAVE AND RUN TESTS (CTRL + ⌘ + ⌘)', 'RUN CODE ONLY (CTRL + ⌘)', 'SOLUTION: 1D 23H 59M 54S', and 'USE GITHUB'. The bottom section shows 'YOUR CODE (regex.py)' and 'TESTS (test_regex.py)' with a code editor displaying Python code for three functions: extract_course_times, get_all_hashtags_and_links, and match_first_paragraph.

PyBites Platform | Bite 2. Regex Fun

localhost:8000/bites/2/

Search Bites of Py Exercises and Blog Code Challenges

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.

findall re regular expressions +

COPY THIS BITE

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).

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

SOLUTION: 1D 23H 59M 54S USE GITHUB

YOUR CODE (regex.py) TESTS (test_regex.py)

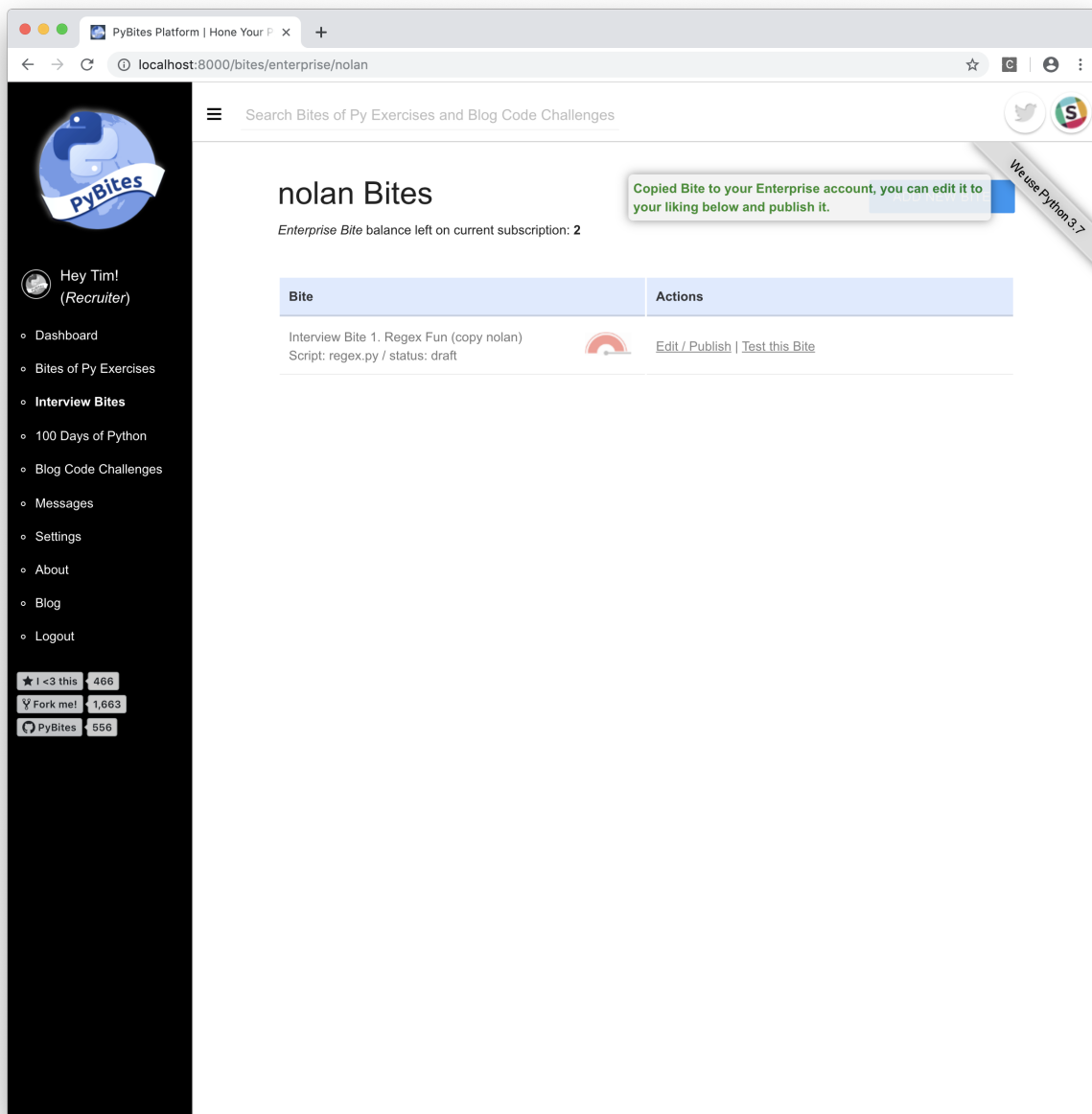
```

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
12     pass
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
21     pass
22
23 def match_first_paragraph():
24     """Write a regular expression that returns 'pybites != greedy' """
25

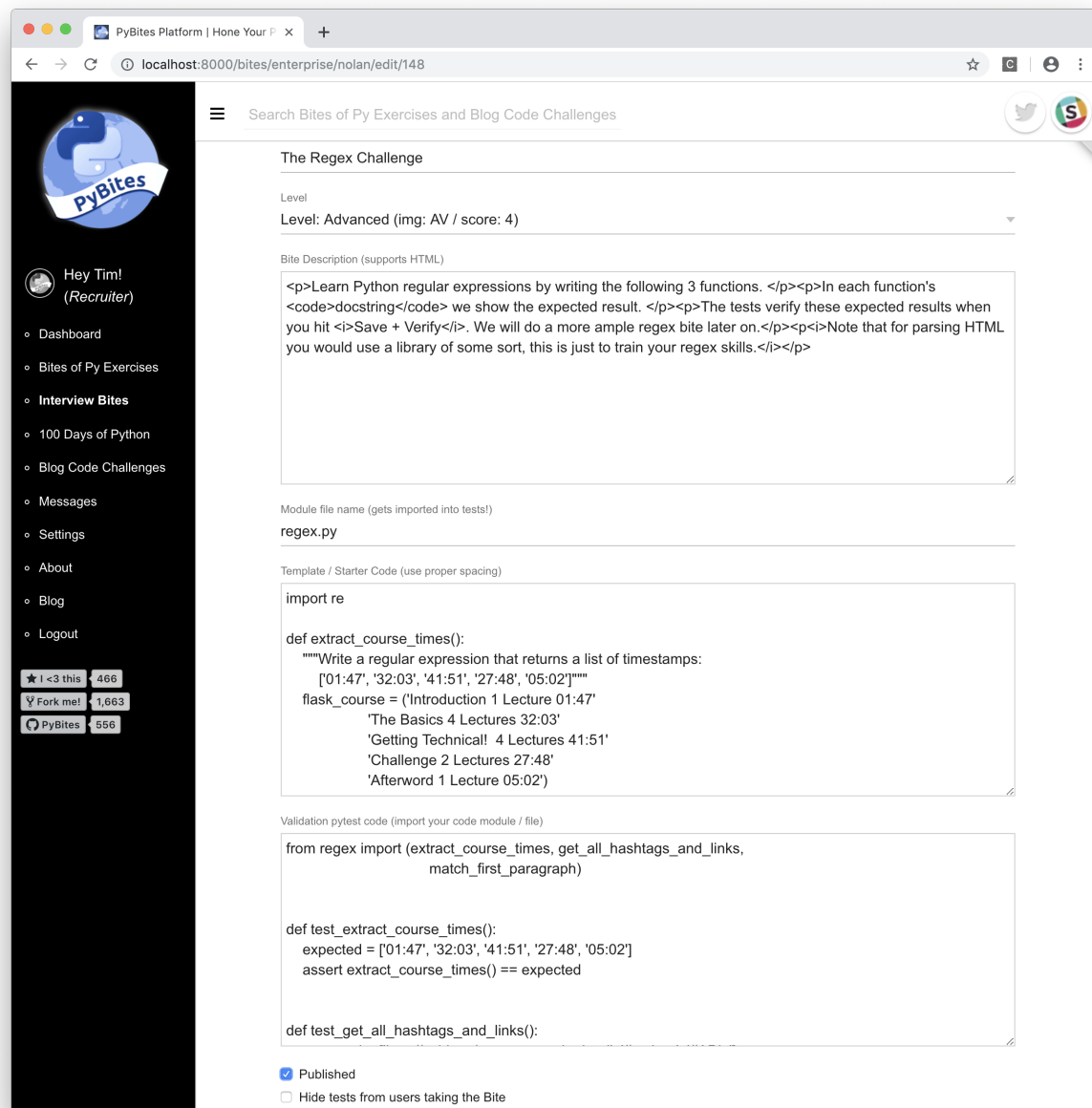
```

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.

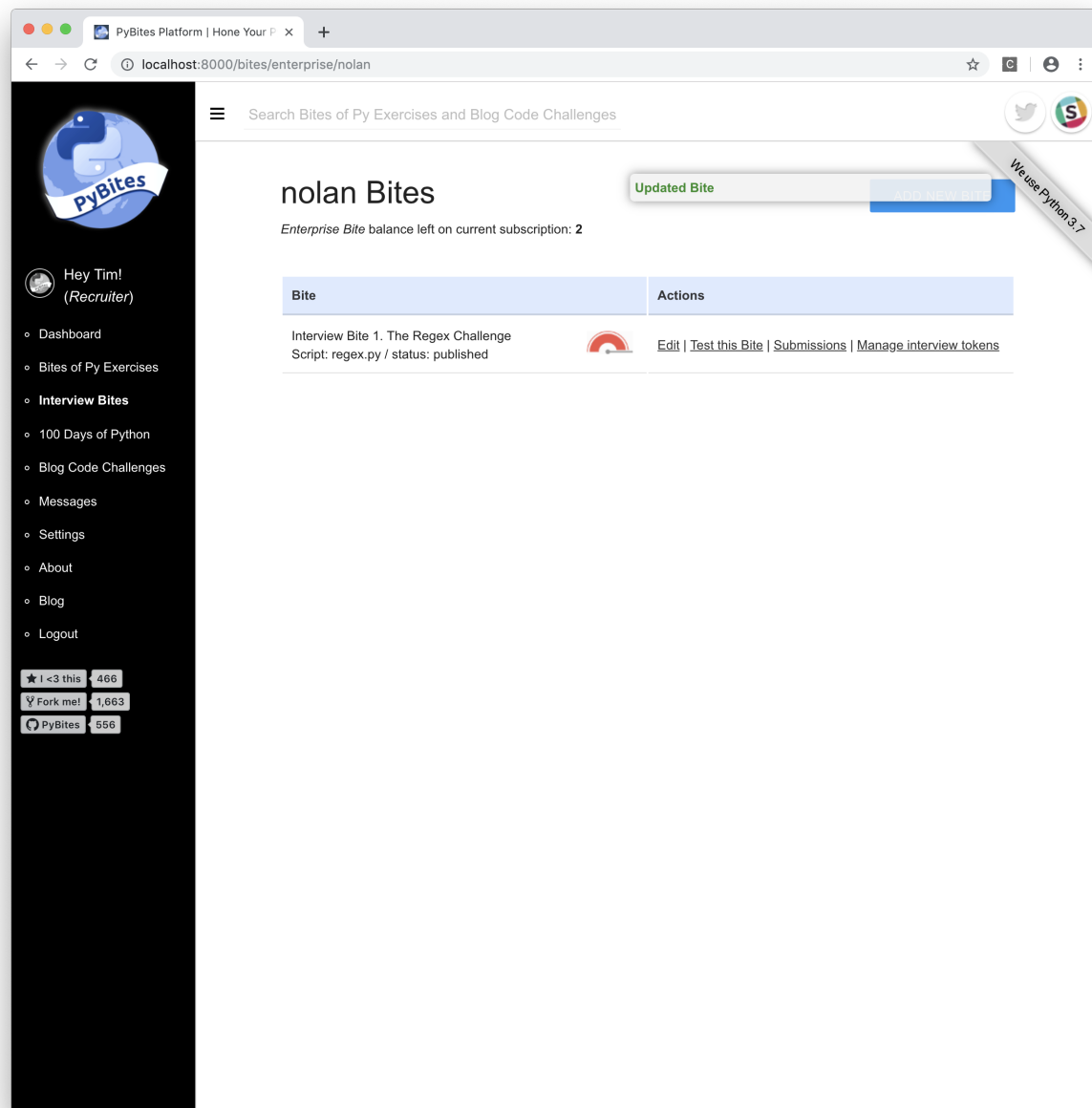
The Bite is now copied to your private Interview Bites!



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

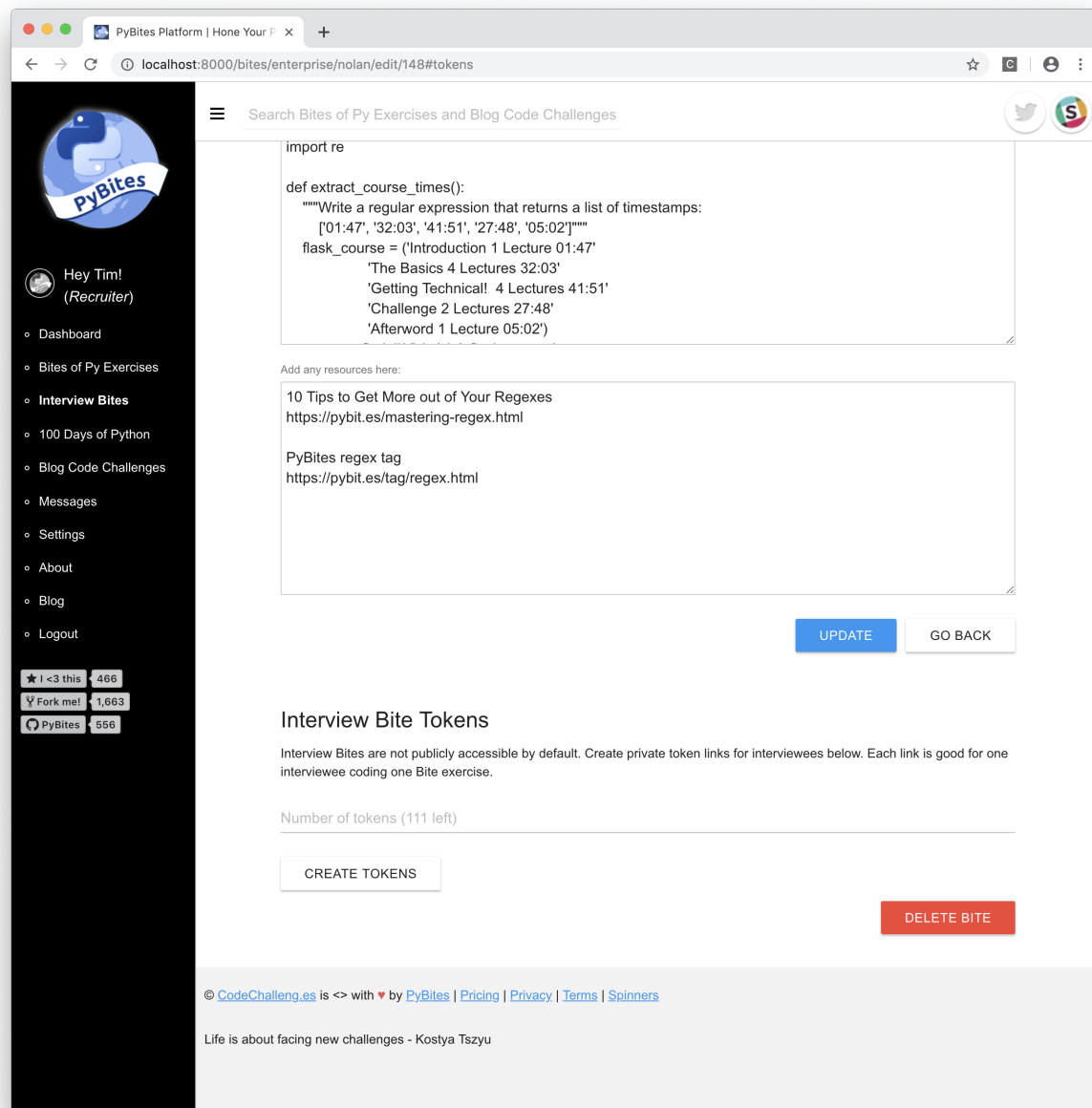


2 new links now appear on the Interview Bites page: *Submissions* and *Manage interview tokens*:



5.7 Interviewee links

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



Let's generate 20 interview links for this one:

The screenshot displays the PyBites Platform interface in a web browser. The browser's address bar shows the URL `localhost:8000/bites/enterprise/nolan/edit/148#tokens`. The interface is divided into a dark sidebar on the left and a main content area on the right.

Sidebar:

- Logo: PyBites
- User: Hey Tim! (Recruiter)
- Navigation links: Dashboard, Bites of Py Exercises, Interview Bites, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, Logout.
- Stats: 466 (stars), 1,663 (forks), 556 (PyBites).

Main Content Area:

- Search bar: Search Bites of Py Exercises and Blog Code Challenges.
- Code Editor: Contains Python code for `extract_course_times()` and a list of timestamps.
- Resources: A section titled "Add any resources here:" with links to "10 Tips to Get More out of Your Regexes" and "PyBites regex tag".
- Buttons: "UPDATE" and "GO BACK".
- Section: "Interview Bite Tokens".
- Text: "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."
- Form: A text input field containing "20|".
- Buttons: "CREATE TOKENS" and "DELETE BITE".

Footer:

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

Life is about facing new challenges - Kostya Tszzyu

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
http://localhost:8000/bites/interview/1TVQ460JAKU30CUZXR8/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/7THR2B09CU8EEJZ7G4TD/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/21L9UIN9EOEBBD4P89C8/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/ZDP7S5Z9VQS2V8YS5G71/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/DOUKXGIDB3ZD28B2C64R/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/AKREV872PUJ4NTZ3ZT4B/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/1DE1RDJBAQ3CKY5NVUGN/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/QFX03SZYTZZRAEWG78LI/nolan-1/	<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

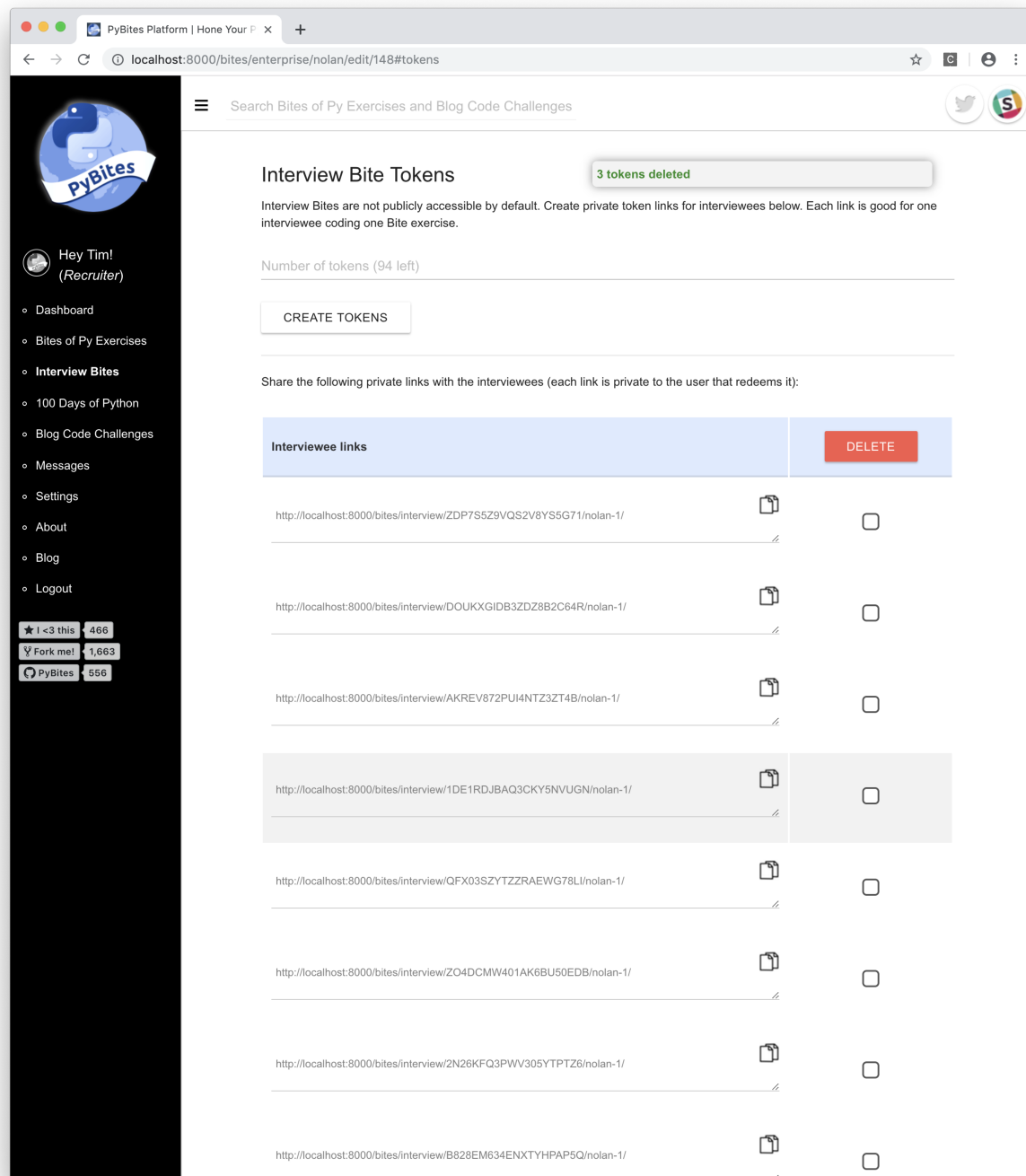
I <3 this 466

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:

The screenshot shows a web browser window at localhost:8000/bites/148/code/. The page title is 'PyBites Platform | Interview Bite 1. The Regex Challenge'. A notification banner at the top right says 'Successfully redeemed interview token'. The left sidebar shows a user profile for 'Hey Sara! (Free)' with a 'START YOUR FREE TRIAL' button. The main content area has the title 'Interview Bite 1. The Regex Challenge' and instructions: '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.' Below the instructions are two buttons: 'SAVE AND RUN TESTS (CTRL + ⌘ + ↵)' and 'RUN CODE ONLY (CTRL + ↵)'. The code editor shows the user's code for 'regex.py' and the tests for 'test_regex.py'. The code includes three functions: 'extract_course_times()', 'get_all_hashtags_and_links()', and 'match_first_paragraph()'. The tests verify the output of these functions. A yellow box at the bottom contains instructions: '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!'

PyBites Platform | Interview Bite 1. The Regex Challenge

START YOUR FREE TRIAL

Hey Sara! (Free)

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

★ I <3 this 466

🍴 Fork me! 1,663

🐍 PyBites 556

Interview Bite 1. The Regex 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 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.

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:

The screenshot shows the PyBites Platform web interface. The header includes a navigation menu with options like Dashboard, Bites of Py Exercises, Pricing, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, and Logout. A sidebar on the left displays user statistics for 'Hey Ramit! (Free)'. The main content area is titled 'Hone Your Python Skills' and features a search bar and a list of exercises categorized by difficulty: Intro, Beginner, Intermediate, and Advanced. A table lists several exercises with their descriptions and tags.

Bite	Description & Tags
f-strings and a simple if/else	The latest way to print to the screen in Python ... f-strings if/else
Infinite loop, input, continue and break	In this Bite we'll get you to take user input ... break continue input mock.patch while
Loop through a dictionary and pluralise a word	You're given a dictionary of people and the number of ... dict looping
Split and join	Split up the message on newline (\n) using the split ... join split
Slice and dice	Take the block of text provided and strip off the ... replace slicing split string module strip
Strip out vowels and count the number of replacements	In this Bite we'd like you to loop over the ... Zen of Python counting regular expressions replace string manipulation
Filter numbers with a list comprehension	Complete the function below that receives a list of numbers ... list comprehensions modulo
Loop over a dict of namedtuples calculating a total score	In this Bite you calculate the total amount of points ... counting dict namedtuple

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:

PyBites Platform | Interview Bite 1. Regex Fun

localhost:8000/bites/148/code/

Chrome is being controlled by automated test software.

START YOUR FREE TRIAL

Hey Ramit! (Free)

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

I <3 this 466

Fork me! 1,663

PyBites 556

Interview Bite 1. Regex Fun (copy not paste)

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.

SAVE AND RUN TESTS (CTRL + ⌘ + S) **RUN CODE ONLY (CTRL + ⌘)**

YOUR CODE (regex.py) TESTS (test_regex.py) **TIME LEFT: 00 0H 59M 58S**

```

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!

He now has a timer!

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

PyBites Platform | Hone Your P x

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

Chrome is being controlled by automated test software.

Search Bites of Py Exercises and Blog Code Challenges

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 (94 left)

CREATE TOKENS

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

Interviewee links	DELETE
JXKYIV28O2BW48C147EX was redeemed by nolan3 1 minute ago	
YP6B4RXIO271WJCADQDA was redeemed by nolan4 1 minute ago	
http://localhost:8000/bites/interview/OEO2FHFX2GX26SDPC0ZY/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/Z1PGVFURRD60VEL3EY63/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/ZRSM8RDYUYPCA1M71VXW/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/CHUZVHL2BB1LHDXBWIW/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/S3AX4M3YTQH28V7FRX79/nolan-1/	<input type="checkbox"/>
http://localhost:8000/bites/interview/TVHPV3Q5GEXZ1VMOUGFK/nolan-1/	<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

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 PyBites Platform interface in a web browser. The page title is "Code Submissions for *The Regex Challenge*". On the left is a dark sidebar with the PyBites logo and a list of navigation items: Dashboard, Bites of Py Exercises, Interview Bites, 100 Days of Python, Blog Code Challenges, Messages, Settings, About, Blog, and Logout. Below the sidebar are social media links for GitHub, Fork me!, and PyBites. The main content area has a search bar and a "DOWNLOAD DATA" button. It displays two submission entries for the challenge. Each entry shows the user's name and stats, a list of emojis, and the final code. The first entry is for "User: nolan4 (Ramit Interviewee2)" with a score of 0 and a duration of 1:08:56. The second entry is for "User: nolan3 (Sara Interviewee1)" with a score of 0. Both entries show the same Python code for the challenge.

Code Submissions for *The Regex Challenge*

[DOWNLOAD DATA](#)

User / stats

- User: nolan4 (Ramit Interviewee2)
- Score: 🍌 0
- Total saves: 4
- Duration: 1:08:56

Final code and last 5 revisions

FINAL CODE 1 2 3 4

```
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')
    return re.findall(r'(\d+:\d+)', flask_course)

def get_all_hashtags_and_links():
    """Write a regular expression that returns this list:
    ['http://pybit.es/requests-cache.html', '#python', '#APIs'] """
    tweet = ('New PyBites article: Module of the Week - Requests-cache '
            'for Repeated API Calls - http://pybit.es/requests-cache.html '
            '#python #APIs')
    return re.findall(r'((?:http|https|ftp)://.*|#[a-zA-Z0-9_]+)', tweet)

def match_first_paragraph():
    """Write a regular expression that returns 'pybytes != greedy' """
    html = ('<p>pybytes != greedy</p>'
            '<p>not the same can be said REgarding ...</p>')
    return re.sub(r'<p>(.*?)</p>.*', r'\1', html)
```

FINAL CODE 1

```
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')
    return re.findall(r'(\d+:\d+)', flask_course)
```

FINAL CODE 1

User: nolan3 (Sara Interviewee1)

Score: 🍌 0


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