

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



1

How to communicate with an API?

Welcome! We have been learning about web servers and HTTP. Today, we will learn how to communicate with an API by sending HTTP requests. APIs (Application Programming Interfaces) provide a structured way to ask for information. APIs are hosted on servers, and we (clients) can send requests to them to obtain information.

In this lab, we will go over two common request types: GET and POST. GET requests are used to retrieve information from the server based on some parameters (such as asking the weather in a city). POST requests are used to send information to the server (such as sending a private message).

2

Tutorial Structure

This tutorial is organized around common *patterns* in programs that send requests to APIs, also known as programming plans.

Each programming plan has a clear goal describing what it helps you achieve, a code template that you can modify for your own use case, and annotations on which areas to change.

Throughout the tutorial, you will first see real-world examples of how APIs are useful, and then you will see how these examples break down into plans.

Contents

- Before You Start The Activity (start_here.html)
- Example 1: Looking for Restaurant Ratings in Champaign-Urbana (integrated_1.html)
 - Plan 1: Set Target URL (set_target_url.html)
 - Plan 2: Set Parameters (set_parameters.html)
 - Plan 3: Make GET Request (make_get_request.html)
 - Plan 4: Show Result If Successful (show_result_if_successful.html)
- Example 2: Talking to your own ChatGPT (integrated_3.html)
 - Plan 1: Set Target URL (set_target_url.html)
 - Plan 5: Set Authentication Credentials (set_authentication_credentials.html)
 - Plan 6: Set Private Content (set_private_content.html)
 - Plan 7: Make POST Request (make_post_request.html)
 - Plan 4: Show Result If Successful (show_result_if_successful.html)
- Exercise: Guess the Number! (exercise_1.html)
- Before You Complete The Activity - 1 (end_here_comparison.html)



(start_here.html)

- Before You Complete The Activity - 2 (end_here_usability.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

```
{{ if request.application == 'runestone': }} {{ pass }}
```



1

2



(start_here.html)

{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Before You Start The Activity

Please rate how much you agree with the each statement. Your responses will be anonymized.

I am interested in understanding how to interact with and retrieve data from APIs.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I am confident in my ability to make API requests and show JSON responses.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

If you completed all the activities on this page, click on the arrow on the bottom right to continue.

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (http://runestoneinteractive.org/) 5.3.0.

| Back to

{{ if request.application == 'runestone': }} {{ pass }}



(index.html)



(integrated_1.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```

Example 1: Looking for Restaurant Ratings in Champaign-Urbana

Champaign and Urbana are two cities known for their vibrant food scene. Local residents and visitors alike are on the lookout for the best dining experiences in Champaign-Urbana. To facilitate this, we aim to fetch top ranking restaurants using an API that rates eateries in the area.

To achieve this, we will send a request to an API that returns the restaurant rankings in an area. You can think of this API to be similar to Google Maps or Yelp. However, for this example, we are using a custom API:

For Google Maps or Yelp, we would look for the **documentation** of the API by searching for “Google Maps API” or “Yelp API”. The documentation of an API provides us with the necessary information on how to use the API, including: - the **endpoint URL**, which is the URL that we will send a request to, - its **parameters**, which is a list of values that we can share with the API to explain our request, - and its **response**, which is what we get from the API after sending a request.

For our restaurant API, the endpoint is: <https://one02-api-fastapi.onrender.com/api/restaurants> (<https://one02-api-fastapi.onrender.com/api/restaurants>).

Our API accepts the following parameters:

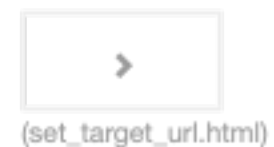
Parameter	Example Value	Explanation
location	‘champaign’	Specify the location that you will request ratings for
count	5	Specify how many of the top restaurants will be shown

Based on this information, we can send a GET request to the API to get the restaurant ratings in Champaign-Urbana.

Enter the URL for the
API you will use.

This example uses the following programming plans:

- Plan 1: Set Target URL (set_target_url.html)
- Plan 2: Set Parameters (set_parameters.html)
- Plan 3: Make GET Request (make_get_request.html)
- Plan 4: Show Result If Successful (show_result_if_successful.html)



Set Target URL: Enter the URL for the API you will use.

```
import requests
target_url = '{url}'
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Set Parameters: Enter parameters that will affect the response from the API.

```
my_params = {
    {key_value_pair},
}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Make GET Request: Call the API by making a GET request to the server.

```
response = requests.get(target_url, params={params})
```

► Show What To Write In Changeable Areas

(start_here.html)

Show Examples

Show Template



(set_target_url.html)

Show Result If Successful: Check if response was successful, and show the data or the error message depending on the result.

```
# Check the result of the request
if response.status_code == {status_code}:
    {process_result}
```

► Show What To Write In Changeable Areas

Show Examples

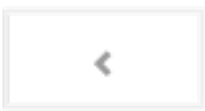
Show Template

You have attempted of activities on this page

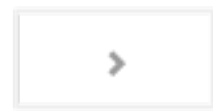
© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| [Back to top](#)

```
{{ if request.application == 'runestone:.' }} {{ pass }}
```



(start_here.html)



(set_target_url.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 1: Set Target URL

Set Target URL: Enter the URL for the API you will use.

```
import requests  
target_url = '{url}'
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This plan sets the URL address for the API you want to use.

Plan 1 - When to use this plan?

This plan is used before you send any requests.

Plan 1 - What parts can be customized to use this plan?

To use this plan, find the URL of the API you want to use online. These URLs might start with 'api' instead of 'www'. See the examples for the plan for some example URLs.

Plan 1 - Exercises



(integrated_1.html)



(set_parameters.html)

p2-1: The URL must be in quotation marks (e.g. `target_url = "api.example.com"`).

- True
- Correct!
- False
- When using Python, you need to wrap the URL address in quotes (either with 'single quotes' or "double quotes").

**Note**

Click here to go back to the main page (</index.html>)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

[Back to](#)

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(integrated_1.html)



(set_parameters.html)


```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 2: Set Parameters

Set Parameters: Enter parameters that will affect the response from the API.

```
my_params = {  
    {key_value_pair},  
}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This plan lists the parameters that will be shared with the API you are calling.

Plan 2 - When to use this plan?

This plan is used when you need to call an API with a set of parameters that do not contain private information. If you want to share private information with the API, you will need to use more secure methods that we will explore soon.

Plan 2 - What parts can be customized to use this plan?

To use this plan, check the documentation of the API to see what type of information is required to send a request. Then, add each piece of information according to the format described. You can Google for an API's documentation to see which parameters they accept.

Plan 2 - Exercises



(set_target_url.html)

(make_get_request.html)

p2-1: Look at the "request parameters" for a weather API in their documentation: <https://www.weatherapi.com/docs/> (<https://www.weatherapi.com/docs/>) According to the documentation, which one of the options would be an incorrect parameter to provide?

- "days": "yesterday"
- Correct!
- "days": 5
- This is a valid option for the number of days of forecast required, which must be between 1 and 14.
- "q": "Champaign"
- This is a valid option for the query parameter, as it is a city name.
- "q": 61820
- This is a valid option for the query parameter, as it is an US zip code.

p2-2: You can provide multiple parameters in the same API call.

- True
- Correct!
- False
- By providing multiple pairs of keys and values, you can pass many parameters to the API in the same call.

Note

Click here to go back to the main page (/index.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (http://runestoneinteractive.org/) 5.3.0.

| Back to

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(set_target_url.html)



(make_get_request.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 3: Make GET Request

Make GET Request: Call the API by making a GET request to the server.

```
response = requests.get(target_url, params={params})
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This is a plan when you want to make a GET request from an API to obtain information. Sending a GET request is similar to going to an URL on your website.

Plan 3 - When to use this plan?

This plan is used when your goal is to collect information (e.g. such as weather or exchange rates). You can use it if you want to make a request to the API where you do not need to share sensitive information.

Plan 3 - What parts can be customized to use this plan?

To use this plan, provide the target url you set earlier, and provide parameters if there are any parameters you would like to share with the API.

Plan 3 - Exercises

p2-1: The 'Make GET Request' plan is used to send sensitive data to a server by including it in the URL.
(set_parameters.html)

(show_result_if_successful.html)

- True
- No, 'Make GET Request' is primarily for retrieving data, not sending it.
- False

- Correct!

p2-2: You are tasked with retrieving data from an online weather API to display the current weather information for a city. Which parameter setting should you use in the code template to ensure the GET request is made without additional query parameters?

- None
- Correct!
- my_params
- Using 'my_params' would imply that you need to specify certain query parameters, which is not required for this task.
- default_params
- 'default_params' is not a recognized option in the given template.
- all_params
- 'all_params' suggests including all possible parameters, which is unnecessary for a simple GET request without specific queries.

Note

Click here to go back to the main page (/index.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (http://runestoneinteractive.org/) 5.3.0.

| Back to

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(set_parameters.html)



(show_result_if_successful.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 4: Show Result If Successful

Show Result If Successful: Check if response was successful, and show the data or the error message depending on the result.

```
# Check the result of the request  
if response.status_code == {status_code}:  
    {process_result}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This is a plan to check the response from the API. The response might indicate a successful operation (code 200), or might indicate some type of error.

Plan 4 - When to use this plan?

This plan is used when you want to handle cases where the request to the API may fail.

Plan 4 - What parts can be customized to use this plan?

To use this plan, look up the HTML error codes, and replace the status code with the code you need to watch out for. Then, you can process the result by printing it directly (if there are no errors), or by showing an error message.

Plan 4 - Exercises



(make_get_request.html)

(integrated_3.html)

p2-1: You are repeatedly sending requests to an API that has been not working for the past hour. You want to automatically print the data when the API is working again and your request is successfully completed. Which status code should you check for to ensure that the response

was successful, allowing you to process and display the JSON data received?

- 200
- Correct!
- 404
- 404 indicates that the requested resource could not be found, so you should handle this as an error.
- 500
- 500 indicates that there is a server error, thus the response is unsuccessful and should be treated as an error.



p2-2: You want to send a request to “api.weather.com/forecast”, but you made a typo while writing the target URL, resulting in the incorrect URL “api.weather.com/forcast”, which does not exist. Which status code would you need to look for to notice this error?

- 404
- Correct!
- 200
- 200 indicates a successful response, but your response would have the "Not Found" error as the mistyped URL does not point to an existing resource.
- 500
- 500 indicates an error due to the server, but in your case the error would be caused from requesting an address that cannot be found.

Note

[Click here to go back to the main page \(/index.html\)](/index.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

[| Back to](#)

```
{{ if request.application == 'runestone:'}} {{ pass }}
```



(make_get_request.html)



(integrated_3.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Example 2: Talking to your own ChatGPT

In this example, we will see how we can talk to ChatGPT using the API to ask a question and get a response.

To achieve this, you would need to buy an **access token** from OpenAI to authenticate your request. For today, you will use a custom large language model that we are providing to you, using the following endpoint URL: <https://one02-api-fastapi.onrender.com/api/chat> (<https://one02-api-fastapi.onrender.com/api/chat>).

The access token for this API is: ACCESS_CS102_GPT

To use the API, you will need to send your message in the body of your request, as “message”. Once authenticated, you can send a question to the ChatGPT API and receive a response. This example covers how to set each part of the request and print the response.

Enter the URL for the
API you will use.

This example uses the following programming plans:

- Plan 1: Set Target URL (set_target_url.html)
- Plan 5: Set Authentication Credentials (set_authentication_credentials.html)
- Plan 6: Set Private Content (set_private_content.html)
- Plan 7: Make POST Request (make_post_request.html)
- Plan 4: Show Result If Successful (show_result_if_successful.html)

Set Target URL: Enter the URL for the API you will use.

```
import requests  
target_url = '{url}'
```

(show_result_if_successful.html)



(set_authentication_credentials.html)

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Set Authentication Credentials: If the API requires a log in or a similar authentication, share your credentials

```
headers = {  
    'Authorization': {authorization_token}  
}
```

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Set Private Content: If you are calling an API on your data, send the data in the body of your message.

```
body = {  
    {key_value_pair},  
}
```

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Make POST Request: Call the API with private data by making a POST request to the server.

(show_result_if_successful.html)

(set_authentication_credentials.html)


```
response = requests.post(target_url, headers={headers}, json={body})
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Show Result If Successful: Check if response was successful, and show the data or the error message depending on the result.

```
# Check the result of the request
if response.status_code == {status_code}:
    {process_result}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| [Back to](#)

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(show_result_if_successful.html)



(set_authentication_credentials.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 5: Set Authentication Credentials

Set Authentication Credentials: If the API requires a log in or a similar authentication, share your credentials

```
headers = {  
    'Authorization': authorization_token  
}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This is a plan to authenticate yourself to an API. This is done by sharing an access token (also called API key).

Plan 5 - When to use this plan?

This plan is used when you want to “log in” to an API. Since we do not have a login interface when working with APIs, we use these tokens or keys to let the API know who we are. In many cases, you might need to login to a website, pay for these access tokens, and then use them to authenticate your requests in your code.

Plan 5 - What parts can be customized to use this plan?

To use this plan, provide the access token you received from the API provider. In our examples, we use ‘ACCESS_CS102_GPT’, but this can change depending on the API you are using.

(integrated_3.html)

Note

Click here to go back to the main page (/index.html)

(set_private_content.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| Back to

```
{{ if request.application == 'runestone': }} {{ pass }}
```



(integrated_3.html)



(set_private_content.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 6: Set Private Content

Set Private Content: If you are calling an API on your data, send the data in the body of your message.

```
body = {  
    {key_value_pair},  
}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This is a plan to send sensitive content to an API. Instead of sending it as a parameter, which is visible in the URL, you send it in the body of the request.

Plan 6 - When to use this plan?

This plan is used when you want to send sensitive information to an API. For example, if you are sending a message to a chatbot or a private message to a friend, you would use this plan. If you submit a form on a website, you would also use this plan.

Plan 6 - What parts can be customized to use this plan?

To use this plan, provide the content you want to send in the body of the request. In our examples, we send the data as 'message', but this tag can change depending on the API you are using.

Plan 6 - Exercises

p2-1: To send a message to a chatbot using a POST request, you would use the 'Set Private

Content' plan that holds the message in the body of the request.

- True
- Correct!
- False
- No, you would use this plan as it prevents the message from being visible in the URL.

**Note**

Click here to go back to the main page (/index.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (http://runestoneinteractive.org/) 5.3.0.

| [Back to](#)

```
{{ if request.application == 'runestone:'}} {{ pass }}
```



(set_authentication_credentials.html)



(make_post_request.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Plan 7: Make POST Request

Make POST Request: Call the API with private data by making a POST request to the server.

```
response = requests.post(target_url, headers={headers}, json={body})
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

This is a plan when you want to make a POST request to an API to obtain information. Sending a POST request is different than the GET request, as you may communicate with the API by sending a message body, rather than just providing parameters.

Plan 7 - When to use this plan?

This plan is used when your goal is to modify information on the server side (e.g. processing the data you sent). You can use it if you want to make a request to the API where you want to share potentially sensitive information, like private messages.

Plan 7 - What parts can be customized to use this plan?

To use this plan, provide the target url you set earlier, provide any headers that you want to share such as your authorization credentials, and provide the body of the message that contains the information you would like to share with the API.

Plan 7 - Exercises

p2-1: In the context of sending a JSON payload to an API endpoint that requires authentication,

which part of the following code template is responsible for specifying the data being sent in the request body? `response = requests.post(target_url, headers=my_headers, json=my_body)`

- `my_body`
- Correct!
- `my_headers`
- Incorrect. '`my_headers`' is used to specify the headers of the request, which might include authentication information.
- `target_url`
- Incorrect. '`target_url`' specifies the endpoint to which the request is being sent.
- `requests.post`
- Incorrect. '`requests.post`' is the method used to send the POST request, but it does not specify the request body.

Note

[Click here to go back to the main page \(/index.html\)](/index.html)

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

[| Back to](#)

```
{{ if request.application == 'runestone': }} {{ pass }}
```



([set_private_content.html](#))



([exercise_1.html](#))

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Exercise: Guess the Number!

In this exercise, you will play a game with an API: the API has a secret number, and you will try to guess it. The API will tell you if your guess is too high or too low. The number is between 1 and 50.

The endpoint for the API is: <https://one02-api-fastapi.onrender.com/api/guess> (<https://one02-api-fastapi.onrender.com/api/guess>)

This endpoint will process your guess, so you will need to POST a request to it. Your guess should be in the body of the request, not in the URL.

Try to plan your solution below.

Q-1: Drag and drop the plans to send your guess using the correct request to the

Set Target URL

Set Private Content

Make POST Request

Show Result If Successful

Set Authentication Credentials #distractor

Make GET Request #distractor

Set Parameters #distractor

Optional: If you want, you can write the code for these plans by copy pasting the code from the plans! Let's see if you can guess the number!

The body of your response should contain two values: a "guess" and a "netid". For example, this could be the body for guessing 25: {"guess": 25, "netid": "katcun"}.

Copy and paste code
from the plans to send a



Here is a list of all programming plans you have learned today:

(end_here_comparison.html)

Set Target URL: Enter the URL for the API you will use.

```
import requests
target_url = '{url}'
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Set Parameters: Enter parameters that will affect the response from the API.

```
my_params = {
    {key_value_pair},
}
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Set Authentication Credentials: If the API requires a log in or a similar authentication, share your credentials

```
headers = {
    'Authorization': {authorization_token}
}
```

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Set Private Content: If you are calling an API on your data, send the data in the body of your message.

```
body = {  
    {key_value_pair},  
}
```

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Make GET Request: Call the API by making a GET request to the server.

```
response = requests.get(target_url, params={params})
```

► Show What To Write In Changeable Areas

[Show Examples](#)[Show Template](#)

Make POST Request: Call the API with private data by making a POST request to the server.



```
(make_request = requests.post(target_url, headers={headers}, json={body}))
```

► Show What To Write In Changeable Areas

Show Examples

Show Template



Show Result If Successful: Check if response was successful, and show the data or the error message depending on the result.

```
# Check the result of the request
if response.status_code == [status_code]:
    [process_result]
```

► Show What To Write In Changeable Areas

Show Examples

Show Template

Click on the arrow on the bottom right to continue.

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| [Back to](#)

```
{{ if request.application == 'runestone: }} {{ pass }}
```



(make_post_request.html)



(end_here_comparison.html)

```
{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}  
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}
```

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

```
{{ pass }}
```



Before You Complete The Activity - 1

Please rate how much you agree with the each statement. Your responses will be anonymized.

I am interested in understanding how to interact with and retrieve data from APIs.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I am confident in my ability to make API requests and show JSON responses.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

If you completed all the activities on this page, click on the arrow on the bottom right to continue.

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (http://runestoneinteractive.org/) 5.3.0.

| Back to

```
{{ if request.application == 'runestone:'}} {{ pass }}
```



(exercise_1.html)



(end_here_usability.html)

{{ if response.serve_ad and settings.adsenseid: }} {{ pass }}
{{ if settings.num_banners > 0 and settings.show_rs_banner: }}

Please Support Runestone

(/runestone/default/donate?banner={{=banner_num}})

{{ pass }}



Before You Complete The Activity - 2

Rate how much you agree with the given statement based on your experience with this interface.

Overall, I am satisfied with how easy it is to use this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I felt comfortable using this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

I believe I learned this material effectively using this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

The information provided with the system was effective in helping me complete the tasks & scenarios.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree

- 6. Agree
- 7. Strongly Agree



I liked using the interface of this system.

- 1. Strongly Disagree
- 2. Disagree
- 3. Somewhat Disagree
- 4. Neither Agree nor Disagree
- 5. Somewhat Agree
- 6. Agree
- 7. Strongly Agree

Q-6: If you would to participate in an OPTIONAL 30-minute interview about your experience, please provide your email address below.

You have reached the end of the activity. Complete your worksheet.

You have attempted of activities on this page

© Copyright 2013-2020 Runestone Interactive LLC. Created using Runestone (<http://runestoneinteractive.org/>) 5.3.0.

| [Back to](#)

```
{{ if request.application == 'runestone:'}} {{ pass }}
```



(end_here_comparison.html)