

Лабораторная работа 1. Реализация удаленного импорта

Разместите представленный ниже код локально на компьютере и реализуйте механизм удаленного импорта. Продемонстрируйте в виде скринкаста или в текстовом отчете с несколькими скриншотами работу удаленного импорта. По шагам:

1/ Создайте файл `myremotemodule.py`, который будет импортироваться, разместите его в каталоге, который далее будет "корнем сервера" (допустим, создайте его в папке `rootserver`).

2/ Разместите в нём следующий код:

```
def myfoo():  
    author = "" # Здесь обозначить своё имя (авторство модуля)  
    print(f"{author}'s module is imported")
```

3/ Создайте файл Python с содержимым функций `url_hook` и классов `URLLoader`, `URLFinder` из текста конспекта лекции со всеми необходимыми библиотеками (допустим, `activation_script.py`).

4/ Далее, чтобы продемонстрировать работу импорта из удаленного каталога, мы должны запустить сервер `http` так, чтобы наш желаемый для импорта модуль "лежал" на сервере (например, в корневой директории сервера). Откроем каталог `rootserver` с файлом `myremotemodule.py` и запустим там сервер:

```
python3 -m http.server
```

5/ После этого мы запускаем файл, в котором содержится код, размещенный выше (обязательно добавление в `sys.path_hooks`).

```
python3 -i activation_script.py
```

6/ Теперь, если мы попытаемся импортировать файл `myremotemodule.py`, в котором размещена наша функция `myfoo` будет выведен `ModuleNotFoundError: No module named 'myremotemodule'`, потому что такого модуля пока у нас нет (транслятор про него ничего не знает).

7/ Однако, как только мы выполним код:

```
sys.path.append("http://localhost:8000")
```

добавив путь, где располагается модуль, в `sys.path`, будет срабатывать наш "кастомный" `URLLoader`. В `path_hooks` будет содержаться наша функция `url_hook`.

8/ Протестируйте работу удаленного импорта, используя в качестве источника модуля другие "хостинги" (например, `repl.it`, `github pages`, `beget`, `sprinthost`).

9/ Переписать содержимое функции `url_hook`, класса `URLLoader` с помощью модуля `requests` (см. комментарии).

Pycharm. Создаем файлы:

```
myremotemodule.py × activation_script.py test_import.py

1 def myfoo(): 1 usage (1 dynamic)
2     author = "Dasha"
3     print(f"{author}'s module is imported")
4     return f"Модуль от {author} успешно загружен!"
5
6 MODULE_VERSION = "1.0"
7
8 if __name__ == "__main__":
9     print("Этот модуль предназначен для удаленного импорта")
```

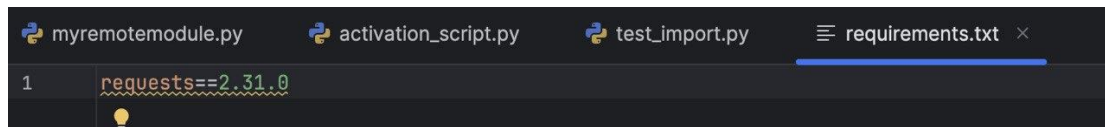
```
myremotemodule.py activation_script.py × test_import.py requirements.txt

1 import re
2 import sys
3 import requests
4 from importlib.abc import PathEntryFinder, Loader
5 from importlib.util import spec_from_loader
6
7
8 class URLLoader(Loader): 1 usage
9     def create_module(self, spec):
10         return None
11
12     def exec_module(self, module):
13         try:
14             print(f" Загружаем модуль из: {module.__spec__.origin}")
15             response = requests.get(module.__spec__.origin)
16             response.raise_for_status()
17             source = response.text
18             code = compile(source, module.__spec__.origin, mode="exec")
19             exec(code, module.__dict__)
20             print(" Модуль успешно загружен!")
21         except requests.exceptions.RequestException as e:
22             raise ImportError(f" Не удалось загрузить модуль: {e}") from e
23
24
25 class URLFinder(PathEntryFinder): 1 usage
26     def __init__(self, url, available):
27         self.url = url
28         self.available = available
```

```
myremotemodule.py  activation_script.py ×  test_import.py  requirements.tx
25  class URLFinder(PathEntryFinder): 1 usage
26
27  30  def find_spec(self, name, target=None):
31      if name in self.available:
32          origin = f"{self.url}/{name}.py"
33          loader = URLLoader()
34          return spec_from_loader(name, loader, origin=origin)
35      return None
36
37
38  def url_hook(some_str): 1 usage
39      if not some_str.startswith(("http", "https")):
40          raise ImportError
41
42      try:
43          print(f" Проверяем хост: {some_str}")
44          response = requests.get(some_str)
45          response.raise_for_status()
46          data = response.text
47          filenames = re.findall(pattern: r"[a-zA-Z_][a-zA-Z0-9_]*\.py", data)
48          modnames = {name[:-3] for name in filenames}
49          print(f" Найдены модули: {modnames}")
50          return URLFinder(some_str, modnames)
51      except requests.exceptions.RequestException as e:
52          print(f" Хост {some_str} недоступен: {e}")
53          raise ImportError(f"Хост недоступен: {some_str}") from e
54
```

```
55
56  # Добавляем хук в sys.path_hooks
57  sys.path_hooks.append(url_hook)
58  print(" URL hook активирован!")
```

```
myremotemodule.py  activation_script.py  test_import.py ×  require
1  import sys
2  import activation_script  # Активируем хук
3
4  print("=== ТЕСТ УДАЛЕННОГО ИМПОРТА ===")
5
6  # 1. Пытаемся импортировать без добавления пути
7  print("1. Попытка импорта без URL пути:")
8  try:
9      import myremotemodule
10
11      print(" Успешно!")
12  except ModuleNotFoundError as e:
13      print(f" Ожидаемая ошибка: {e}")
14
15  # 2. Добавляем URL путь
16  print("\n2. Добавляем URL путь:")
17  sys.path.append("http://localhost:8000")
18  print(" Путь http://localhost:8000 добавлен в sys.path")
19
20  # 3. Пытаемся импортировать с URL путем
21  print("\n3. Попытка импорта с URL путем:")
22  try:
23      import myremotemodule
24
25      print(" Модуль успешно импортирован удаленно!")
26
27      # 4. Тестируем функции модуля
28      print("\n4. Тестируем функции модуля:")
29      result = myremotemodule.myfoo()
30      print(f"Результат: {result}")
31      print(f"Версия модуля: {myremotemodule.MODULE_VERSION}")
32
33  except Exception as e:
34      print(f" Ошибка при импорте: {e}")
35
36  # 5. Тестируем обработку ошибок
37  print("\n5. Тестируем обработку недоступного хоста:")
38  try:
39      sys.path.append("http://nonexistent-domain-12345.com")
40      import some_module
41  except ImportError as e:
42      print(f" Обработка ошибок работает: {e}")
```



Устанавливаем зависимости. В терминале переходим в папку rootserver и запускаем сервер:

```
Terminal Local x + v
(.venv) tarassiky@MacBook-Air-Tarasova PyCharmMiscProject % pip install -r requirements.txt
Requirement already satisfied: requests==2.31.0 in ./venv/lib/python3.13/site-packages (from -r requirements.txt (line 1)) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in ./venv/lib/python3.13/site-packages (from requests==2.31.0->-r requirements.txt (line 1)) (3.4.3)
Requirement already satisfied: idna<4,>=2.5 in ./venv/lib/python3.13/site-packages (from requests==2.31.0->-r requirements.txt (line 1)) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in ./venv/lib/python3.13/site-packages (from requests==2.31.0->-r requirements.txt (line 1)) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in ./venv/lib/python3.13/site-packages (from requests==2.31.0->-r requirements.txt (line 1)) (2025.8.3)

[notice] A new release of pip is available: 25.1.1 -> 25.2
[notice] To update, run: pip install --upgrade pip

(.venv) tarassiky@MacBook-Air-Tarasova PyCharmMiscProject % cd rootserver

(.venv) tarassiky@MacBook-Air-Tarasova rootserver % python -m http.server 8000
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
```

Открываем новый терминал (не закрывая сервер) и запускаем:

```
Terminal Local x Local (2) x + v
(.venv) tarassiky@MacBook-Air-Tarasova PyCharmMiscProject % cd ~/PyCharmMiscProject

(.venv) tarassiky@MacBook-Air-Tarasova PyCharmMiscProject % python test_import.py
URL hook активирован!
=== ТЕСТ УДАЛЕННОГО ИМПОРТА ===
1. Попытка импорта без URL пути:
Ожидаемая ошибка: No module named 'myremotemodule'

2. Добавляем URL путь:
Путь http://localhost:8000 добавлен в sys.path

3. Попытка импорта с URL путем:
Проверяем хост: http://localhost:8000
Найдены модули: {'myremotemodule'}
Загружаем модуль из: http://localhost:8000/myremotemodule.py
Модуль успешно загружен!
Модуль успешно импортирован удаленно!

4. Тестируем функции модуля:
Dasha's module is imported
Результат: 000000 000000 0X00 Dasha 0000000000000000 0-0-0*0000000000X!
Версия модуля: 1.0

5. Тестируем обработку недоступного хоста:
Проверяем хост: http://nonexistent-domain-12345.com
Хост http://nonexistent-domain-12345.com недоступен: HTTPConnectionPool(host='nonexistent-domain-12345.com', port=80): Max retries exceeded with url: / (Caused by NameResolutionError("
```

Реализация в Replit:

```
image-size-checker Upgrade r x t x
myremotemodule.py > ... Form

1 def myfoo():
2     author = "Dasha"
3     print(f"{author}'s module is imported")
4     return f"Модуль от {author} успешно загружен!"
5
6 MODULE_VERSION = "1.0"
7
8 if __name__ == "__main__":
9     print("Этот модуль предназначен для удаленного импорта")
```



```
image-size-checker Upgrade myremotemod... urlhook.py
urlhook.py > URLLoader > f exec_module > ...

1  import re
2  import sys
3  import requests
4  from importlib.abc import PathEntryFinder
5  from importlib.util import spec_from_loader
6
7  class URLLoader:
8      def create_module(self, target):
9          return None
10
11     def exec_module(self, module):
12         try:
13             print(f"    Загружаем модуль из: {module.__spec__.origin}")
14             response = requests.get(module.__spec__.origin)
15             response.raise_for_status()
16             source = response.text
17             code = compile(source, module.__spec__.origin, mode="exec")
18             exec(code, module.__dict__)
19             print("    Модуль успешно загружен!")
20         except requests.exceptions.RequestException as e:
21             raise ImportError(f"    Не удалось загрузить модуль: {e}") from e
22
23     class URLFinder(PathEntryFinder):
24         def __init__(self, url, available):
25             self.url = url
26             self.available = available
27
28         def find_spec(self, name, target=None):
29             if name in self.available:
30                 origin = "{}/{ }.py".format(self.url, name)
31                 loader = URLLoader()
32                 return spec_from_loader(name, loader, origin=origin)
33             else:
34                 return None
35
```

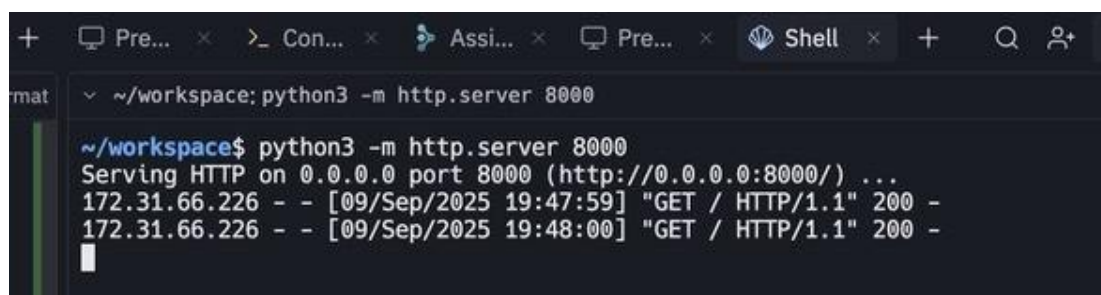
```

urlhook.py > URLLoader > f exec_module > ...
36 def url_hook(some_str):
37     if not some_str.startswith(("http", "https")):
38         raise ImportError
39
40     try:
41         print(f"    Проверяем хост: {some_str}")
42         response = requests.get(some_str)
43         response.raise_for_status()
44         data = response.text
45         filenames = re.findall("[a-zA-Z_][a-zA-Z0-9_]*.py", data)
46         modnames = {name[:-3] for name in filenames}
47         print(f"    Найдены модули: {modnames}")
48         return URLFinder(some_str, modnames)
49     except requests.exceptions.RequestException as e:
50         print(f"    Хост {some_str} недоступен: {e}")
51         raise ImportError(f"Хост недоступен: {some_str}") from e
52
53 # Добавляем хук в sys.path_hooks
54 sys.path_hooks.append(url_hook)
55 print("    URL hook активирован!")

```

Запускаем HTTP сервер. В терминале Replit выполняем:

```
python3 -m http.server 8000
```



```

+ Pre... x >_ Con... x Assi... x Pre... x Shell x + Q +
mat ~ /workspace: python3 -m http.server 8000

~/workspace$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
172.31.66.226 - - [09/Sep/2025 19:47:59] "GET / HTTP/1.1" 200 -
172.31.66.226 - - [09/Sep/2025 19:48:00] "GET / HTTP/1.1" 200 -

```

Тестируем удаленный импорт. Открываем второй терминал в Replit и создаем файл test_import.py:

```
python3 test_import.py
```

test_import.py > ...

```
1  # test_import.py
2  import sys
3
4  # Сначала активируем хук
5  import urlhook
6
7  print("=== ТЕСТ УДАЛЕННОГО ИМПОРТА ===")
8
9  # 1. Пытаемся импортировать без добавления пути
10 print("1. Попытка импорта без URL пути:")
11 try:
12     import myremotemodule
13     print("    Успешно!")
14 except ModuleNotFoundError as e:
15     print(f"    Ожидаемая ошибка: {e}")
16
17 # 2. Добавляем URL путь
18 print("\n2. Добавляем URL путь:")
19 sys.path.append("http://localhost:8000")
20 print("    Путь http://localhost:8000 добавлен в sys.path")
21
22 # 3. Пытаемся импортировать с URL путем
23 print("\n3. Попытка импорта с URL путем:")
24 try:
25     import myremotemodule
26     print("    Модуль успешно импортирован удаленно!")
27
28     # 4. Тестируем функции модуля
29     print("\n4. Тестируем функции модуля:")
30     result = myremotemodule.myfoo()
31     print(f"Результат: {result}")
32     print(f"Версия модуля: {myremotemodule.MODULE_VERSION}")
33
```

test_import.py > ...

```
33
34 except Exception as e:
35     print(f"    Ошибка при импорте: {e}")
36
37 # 5. Тестируем обработку ошибок
38 print("\n5. Тестируем обработку недоступного хоста:")
39 try:
40     sys.path.append("http://nonexistent-domain-12345.com")
41     import some_module
42 except ImportError as e:
43     💡 print(f"    Обработка ошибок работает: {e}")
```



```
~/workspace: python3 test_import.py 2

~/workspace$ python3 test_import.py
URL hook активирован!
=== ТЕСТ УДАЛЕННОГО ИМПОРТА ===
1. Попытка импорта без URL пути:
Успешно!

2. Добавляем URL путь:
Путь http://localhost:8000 добавлен в sys.path

3. Попытка импорта с URL путем:
Модуль успешно импортирован удаленно!

4. Тестируем функции модуля:
Dasha's module is imported
Результат: Модуль от Dasha успешно загружен!
Версия модуля: 1.0

5. Тестируем обработку недоступного хоста:
Проверяем хост: http://localhost:8000
Найдены модули: {'myremotemodule', '_', 'test_import', 'urlhook'}
Проверяем хост: http://nonexistent-domain-12345.com
Хост http://nonexistent-domain-12345.com недоступен: HTTPConnectionPool
(host='nonexistent-domain-12345.com', port=80): Max retries exceeded with
url: / (Caused by NameResolutionError("<urllib3.connection.HTTPConnectio
n object at 0x7f5876d65710>: Failed to resolve 'nonexistent-domain-12345.
com' ([Errno -2] Name or service not known)"))
Обработка ошибок работает: No module named 'some_module'
~/workspace$
```

Ссылка на Replit:

<https://replit.com/@tarasovadasha5/image-size-checker#myremotemodule.py>