

Woche 7: Netzwerk, APIs, Web Scraping

Willkommen zur siebten Woche Ihrer Python-Lernreise! Diese Woche wird sich auf die Interaktion mit dem Web konzentrieren. Es handelt sich um ein äußerst relevantes Thema, da viele moderne Anwendungen die Kommunikation über das Internet und das Parsen von Webdaten erfordern.

Die Woche beginnt mit dem Studium von Sockets, die eine grundlegende Technologie zur Kommunikation zwischen Computern sind. Danach werden Sie sich mit RESTful APIs beschäftigen, welche ein übliches Muster zur Gestaltung von Netzwerkschnittstellen sind. Sie werden auch JSON und YAML kennenlernen, zwei weit verbreitete Datenformate, die oft in Verbindung mit APIs verwendet werden.

Sie werden auch die Requests-Bibliothek verwenden, ein leistungsstarkes Werkzeug, das Ihnen ermöglicht, HTTP-Anfragen einfach zu senden und zu bearbeiten. Ein weiterer wichtiger Teil dieser Woche ist BeautifulSoup, ein Python-Modul, das beim Parsen von HTML- und XML-Dokumenten hilfreich ist.

Ihre Aufgaben für diese Woche umfassen die Erstellung einer einfachen Client-Server-Anwendung, die Nutzung einer REST-API und das Extrahieren von Daten von einer Webseite. Dabei werden Sie auch praktische Erfahrungen mit HTML-Parsing sammeln.

Am Übungstag werden Sie Ihr Wissen vertiefen und haben die Möglichkeit, verschiedene Web-Frameworks wie Django, Flask und FastAPI kennenzulernen. Sie werden auch einen Ausblick auf das Konzept der asynchronen Programmierung erhalten, das besonders nützlich ist, wenn Sie mit Netzerkanwendungen arbeiten.

Bereiten Sie sich darauf vor, in die spannende Welt des Web-Scraping und der Netzwerkkommunikation einzutauchen. Viel Erfolg!

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
 - Sockets
 - RESTful APIs
 - JSON und YAML
 - Requests
 - Beautiful Soup
- Aufgabe:
 - Einfache Client-Server-App erstellen
 - Rest- API nutzen
 - Daten aus Webseite extrahieren
 - HTML parsen
- Tag 8:
 - Wiederholung
 - Vertiefung: requests
 - Vertiefung: Client-Server Anwendungen
 - Ergänzung: Überblick Django, Flask, FastAPI
 - Ergänzung: async/await,
 - Ausblick

Inhalte und thematische Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein.

Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

- Sockets
 - Grundlegende Begriffe: IP-Adresse, Port, TCP, UDP
 - Aufbau und Verwendung von Sockets in Python
 - Client-Server-Modell und Sockets
 - Fehlerbehandlung und sichere Programmierung mit Sockets
- RESTful APIs
 - Was ist eine API und warum sie wichtig ist
 - Konzepte von REST: Ressourcen, Zustandslosigkeit, Methoden
 - Arbeiten mit RESTful APIs in Python
- JSON und YAML
 - Überblick und Vergleich der Formate JSON und YAML
 - Verwendung in Python: Lesen, Schreiben, Parsen
 - Praktische Anwendung bei der Arbeit mit APIs
- Requests
 - Grundlagen der HTTP-Kommunikation
 - Verwendung der Requests-Bibliothek für HTTP-Anfragen
 - Arbeit mit HTTP-Antworten
- BeautifulSoup
 - Grundlagen des Webscrapings
 - HTML-Struktur und CSS-Selektoren
 - Nutzung von BeautifulSoup zur Extraktion von Daten aus HTML

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Beginnen Sie mit der Einführung in die Netzwerkprogrammierung in Python und den Grundlagen der Sockets. Erfahren Sie, wie IP-Adressen, Ports, TCP und UDP in diesem Kontext funktionieren.
2. Vertiefen Sie Ihr Verständnis von Client-Server-Modellen und der Verwendung von Sockets in Python. Üben Sie Fehlerbehandlung und sichere Programmierung mit Sockets.
3. Erschließen Sie sich das Wissen über APIs, insbesondere RESTful APIs. Verstehen Sie, was eine API ist und warum sie wichtig ist, und lernen Sie die Grundprinzipien von REST kennen.
4. Lernen Sie, wie man mit RESTful APIs in Python arbeitet. Üben Sie das Senden von Anfragen und das Empfangen und Verarbeiten von Antworten.
5. Setzen Sie sich mit den Datenformaten JSON und YAML auseinander. Verstehen Sie die Unterschiede und Gemeinsamkeiten zwischen den beiden und lernen Sie, wie man sie in Python liest, schreibt und parsed.
6. Vertiefen Sie Ihr Wissen im Umgang mit APIs durch die Arbeit mit der Requests-Bibliothek. Lernen Sie, HTTP-Anfragen zu senden und Antworten zu empfangen und zu verarbeiten.
7. Machen Sie sich mit den Grundlagen des Webscraping vertraut und lernen Sie die HTML-Struktur und CSS-Selektoren kennen.
8. Nutzen Sie BeautifulSoup zur Extraktion von Daten aus HTML. Üben Sie das Parsen von Webseiten und das Extrahieren von nützlichen Informationen.
9. Übertragen Sie Ihr neu erworbenes Wissen auf eine Reihe von Programmieraufgaben.

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. **Socket-Anwendung:** Erstellen Sie eine einfache Client-Server-Anwendung, bei der der Server aktuelle Zeitdaten an den Client sendet, sobald eine Verbindung hergestellt wird. Nutzen Sie dafür Sockets in Python.
2. **RESTful-API:** Verwenden Sie die Requests-Bibliothek, um Daten von einer RESTful-API zu holen. Wählen Sie eine API Ihrer Wahl (wie zum Beispiel die OpenWeatherMap API) und analysieren Sie die erhaltenen Daten.
3. **JSON-Verarbeitung:** Schreiben Sie ein Python-Programm, das eine JSON-Datei einliest und bestimmte Informationen daraus extrahiert. Zum Beispiel könnte die Datei Daten über verschiedene Länder enthalten und Ihr Programm soll die Bevölkerungszahl und die Hauptstadt jedes Landes ausgeben.
4. **Webscraping:** Wählen Sie eine Webseite aus und nutzen Sie BeautifulSoup, um bestimmte Informationen von der Seite zu extrahieren. Es könnte sich zum Beispiel um Nachrichtenartikel, Produktdetails oder Ähnliches handeln.
5. **HTML-Parsing:** Schreiben Sie ein Python-Programm, das eine heruntergeladene HTML-Seite einliest und alle Links auf der Seite extrahiert. Das Programm sollte die vollständigen URLs der Links ausgeben. Nutzen Sie BeautifulSoup für diese Aufgabe.

Abschluss-Quiz

Das Quiz soll Ihnen einen ersten Hinweis auf Ihren Lernfortschritt geben. Nach unserer Einschätzung sollten Sie diese Fragen alle beantworten können, wenn Sie den Stoff der Woche durchgearbeitet und verstanden haben. Natürlich gibt es noch sehr viel mehr mögliche Fragen, dazu wollen wir auf die Literatur und das Internet verweisen. Geben Sie gerne einmal „python quizzes“ bei Google ein.

1. Welche der folgenden Python-Bibliotheken wird zum Senden HTTP-Anfragen verwendet?
 1. BeautifulSoup
 2. Requests
 3. JSON
 4. Sockets
2. Welches Python-Modul würde man für das Parsing von HTML- und XML-Dokumenten verwenden?
 1. re
 2. json
 3. html.parser
 4. csv
3. Welche der folgenden Funktionen wird in Python für das Einlesen von JSON-Daten verwendet?
 1. json.dump()
 2. json.dumps()
 3. json.load()
 4. json.loads()
4. Welche Methode würde man in Python verwenden, um Daten an einen Server zu senden?
 1. socket.receive()
 2. socket.send()
 3. socket.connect()
 4. socket.bind()
5. Welche Python-Bibliothek wird zum Extrahieren von Daten aus HTML- und XML-Dateien verwendet?
 1. requests
 2. html.parser
 3. BeautifulSoup
 4. re
6. Was ist eine RESTful API?
 1. Ein Python Modul
 2. Ein Format für den Austausch von Daten zwischen Client und Server
 3. Eine Art von Webdienst, der das HTTP-Protokoll verwendet
 4. Ein Datenbankmanagement-System
7. Was macht die Methode json.loads() in Python?
 1. Sie konvertiert ein JSON-Objekt in einen Python-String
 2. Sie konvertiert einen Python-String in ein JSON-Objekt
 3. Sie liest eine JSON-Datei
 4. Sie schreibt in eine JSON-Datei
8. Welche Methode wird verwendet, um eine Verbindung zu einem Remote-Socket herzustellen?

1. `socket.bind()`
 2. `socket.connect()`
 3. `socket.listen()`
 4. `socket.accept()`
9. Wie extrahiert man alle Links einer HTML-Seite mit BeautifulSoup?
1. Mit dem Tag "a"
 2. Mit dem Tag "link"
 3. Mit dem Tag "href"
 4. Mit dem Tag "src"
10. Was ist der Hauptvorteil der Verwendung von RESTful APIs?
1. Sie erlauben Echtzeitkommunikation
 2. Sie stellen sicher, dass die Daten immer verschlüsselt sind
 3. Sie ermöglichen Kommunikation zwischen Anwendungen unabhängig von der Programmiersprache
 4. Sie verbessern die Geschwindigkeit der Datenübertragung

Antworten:

1/2 2/3 3/3 4/2 5/3 6/3 7/2 8/1 9/4 10/3

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und verstanden worden sein.

- Buch: Introduction to Python Network Automation: The First Journey:
 - Grundlagen Python (hat man in diesen Büchern immer!)
 - Aufbau einer Arbeitsumgebung mit Linux
 - Paramiko und Netmiko
 - CISCO Router
- Buch: Network Programming in Python: The Basic: A Detailed Guide to Python 3:
 - Umfangreiche Einführung in LowLevel Funktionalität im Netz
 - Alle wichtigen Protokolle
- Buch: Practical Web Scraping for Data Science:
 - Einführung in Webscraping
 - Viele Beispiele
- Video:
 - Python Web Requests
 - Building Restful Web Services with Python
- Lab and Course:
 - Lab: Building Restful Web Services with Python
 - Course: Learn Web Scraping with Beautiful Soup

Anhänge

Aktuell eine Leerseite