

LU07.A01 - Discovery Service erstellen



Erstelle einen einfachen Discovery Service auf Basis von Sockets. Die Programmvorlage basiert auf dem Code von <https://realpython.com/python-sockets/>.

GitHub Classroom Assignment

Ausgangslage

Der Discovery-Service ist ein Python-Skript, welches die Services in unserem verteilten System verwaltet. Die Kommunikation mit diesem Service erfolgt über Sockets.

Das Python-Skript verwaltet die Services in der Klasse `Services`. Diese Klasse enthält eine Liste von Dictionaries. Für jeden Service wird ein Eintrag in der Liste erstellt.

Key	Value	Datentyp
<code>uuid</code>	UUID des Services	String
<code>type</code>	Die Art des Services	String
<code>ip</code>	Unter welcher IP-Adresse ist der Service erreichbar	String
<code>port</code>	Unter welchem Port ist der Service erreichbar	int
<code>heartbeat</code>	Zeitpunkt des letzten Herzschlags	datetime

Aktionen

register

Jeder Service muss sich beim Starten beim Discovery Service registrieren. Der Discovery-Service erzeugt eine neue UUID für diesen Service und trägt den Service in seine Serviceliste ein. Als `heartbeat` wird der aktuelle Zeitstempel eingetragen.

- Argumente:
 - `type`: Die Art des Service
 - `ip`: Unter welcher IP-Adresse ist der Service erreichbar
 - `port`: Unter welchem Port ist der Service erreichbar
- Response:
 - Die UUID des Services.
 - `ERROR`: Der Service konnte nicht registriert werden.

heartbeat

Jeder Service muss sich in regelmässigen Abständen (z.B. 5 Sekunden) beim Discovery Service melden. Mit dieser Aktion wird das Attribut `heartbeat` der bestehenden Registration aktualisiert.

- Argumente:
 - `service_uuid`: String
- Response:
 - **OK**: Aktualisierung war erfolgreich
 - **NOT FOUND**: Die `service_uuid` wurde nicht gefunden, der Service muss sich neu registrieren.
 - **ERROR**: Es ist ein Fehler aufgetreten.

query

Ein Client kann eine Liste aller Services für einen bestimmte Art von Service (`type`) anfordern. Der Discovery-Service durchsucht seine Serviceliste nach Services mit dem gewünschten `type`. Entspricht ein Service der gewünschten Art **UND** der letzte Herzschlag ist weniger als 5 Sekunden alt, so wird dieser Service in die Response aufgenommen. Jeder Service dessen letzter Herzschlag älter als 5 Sekunden ist, wird aus der Serviceliste entfernt.

- Argumente:
 - `type`: Die Art der gesuchten Services
- Response:
 - Eine Liste mit der IP-Adresse und dem Port pro Service.

Testen

Um den Discovery-Service zu testen, steht dir im Repository ein einfacher Socket-Client.

1. Starte deinen Discovery-Service in PyCharm oder über die Kommandozeile.
2. Öffne ein separates Terminal und navigiere zum Ordner mit dem Socket-Client.
3. Starte den Client mit `python socket_client.py <IP> <PORT>`.

M321-LU07



Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m321/learningunits/lu07/aufgaben/discovery?rev=1711631267>

Last update: **2024/03/28 14:07**

