Python

REST: Flask Music Server

Building a REST server in Python

The music server case study

Goals

- Understanding how to design a REST service
- Implementing a REST service in Python
  - Using Flask
- Be concrete
  - Build upon previous knowledge (Lab)
  - REST Music Server: Flask Music Server

Flask Music Server

- A basic music server that:
  - Given a root directory scans all the sub-folders for music files (mp3 and flac, for the sake of simplicity)
  - For each file provides metadata such as:
    - Author, title, album, genre
  - Can play
    - Single tracks
    - Playlists

Resources

- Identify the resources to expose
  - Track
    - Represents a single song (file) with its own metadata
    - Each track is identified by a unique id
  - Tracks
    - The set of tracks available in the given directory
  - Player
    - A resource able to play music

REST service design

- ## URLs built on resources
  - `/music/api/v1.0/tracks` → the collection of Tracks
  - `/music/api/v1.0/player` → the Player resource

- ## How to represent single tracks
  - They are identified by a unique id
    - `/music/api/v1.0/tracks/134` → the Track with id=134

Tracks

- GET /music/api/v1.0/tracks
    - Returns all the tracks as json
- GET /music/api/v1.0/tracks/154
    - Returns track 154 as json
- Example

```
{
        "track": {
                "id": 154,
                "metadata": {
                        "album": "TRANSATLANTIC R.P.M.",
                        "artist": "Incognito",
                        "genre": "Acid Jazz",
                        "title": "Can't Get Enough«
                },
                "path": "/home/user/Music/Incognito/Incognito -Transatlantic R.P.M
(2010)/14. Incognito - Can't Get Enough.flac«
        }
}
```

Player

- A concrete resource
  - Represents a real music player
  - Has commands
- Commands are sent through PUT requests
  - PUT → resource update
  - Example:
    - PUT /music/api/v1.0/player

```
{
        "command" : "play",
        "playlist" : {
                    "tracks":[3, 55, 63, 77]
        }
}
```

Result

- Checkout on GitHub
  - https://github.com/AmI-2015/python-rest

Questions?

01PRD A<small>MBIENT</small> I<small>NTELLIGENCE</small>: T<small>ECHNOLOGY AND</small> D<small>ESIGN</small>

Dario Bonino

dario.bonino@polito.it