

Developers

Wie funktioniert moosrdata?

Das moosrdata Backend greift im Hintergrund über 40 verschiedene Provider für dich zu und sucht die gewünschten Informationen zusammen. Dabei spricht moosrdata über 25 Provider an und greift hierfür auf über 40 verschiedene Implementierungen zu. Die hierzu verwendete library ist libglyr. ([Link](#))

Moosrdata Metadata Provider

Folgende Onlineprovider werden von moosrdata genutzt:

- bbcmusic
- chartlyrics
- chordie
- discogs
- elyrics
- flickr
- generated
- google
- htbackdrops
- jamendo
- lastfm
- lipwalk
- lyrdb
- lyricsreg
- lyricstime
- lyricsvip
- lyricswiki
- lyrix
- magistrix
- metallum
- metrolyrics
- musicbrainz
- picsearch
- rhapsody
- singerpictures
- slothradio

Lokale Provider:

- local (sqlcache)
- musictree

Moosrdata API Usage:

Die moosrdata engine kann auf mehrere Varianten angesprochen werden. Über eine ganz normale HTML query oder zur maschinellen Auswertung über die api Schnittstelle, die die gewünschten Metadaten im Jason format liefert.

HTML Version:

```
http://www.moosr.org/query/<type>/<number>/<query>
```

Json Version:

```
http://www.moosr.org/api/<type>/<number>/<query>
```

Suchparameter

- <type>, Art der Metadaten, lyrics, cover, artistbio oder artistphoto
- <number>, Anzahl der gewünschten Ergebnisse
- <query>, Die zu suchenden Parameter, hierbei ist darauf zu achten das bei der Lyricssuche oder der Coversuche neben dem Künstler ein zweiter Parameter wie Album/Titel zwingend anzugeben ist (siehe Beispiel)

Beispiel für die Suche des Liedes "Innocence" von Avril Lavigne:

```
http://www.moosr.org/api/lyrics/3/Avril Lavigne+Innocence
```

Beim Zugriff der API über "query" wird ein Json Dokument zurückgeliefert. Bei folgender beispielhafter Anfrage

```
http://www.moosr.org/query/artistphoto/3/Avril Lavigne
```

wird nach Artistphotos der Künstlerin "Avril Lavigne" gesucht, die Anzahl der Ergebnisse wurde hierbei auf drei beschränkt. Die Anfrage würde folgendes Json Dokument zurückliefern:

```
{
  "album": "",
  "artist": "Avril Lavigne",
  "get_type": "artistphoto",
  "number": 3,
  "results": [
    {
      "image_format": "png",
      "is_cached": true,
      "md5sum": "155519bb42f2e38e9cc924d4382779c8",
      "provider": "lastfm",
      "size": 128765,
      "url": "http://userserve-ak.last.fm/serve/252/46759181.png"
    },
    {
      "image_format": "jpeg",
      "is_cached": true,
      "md5sum": "045a6db8d06840edee476f4cfd50e1b",
      "provider": "lastfm",
      "size": 36431,
    }
  ]
}
```

```
        "url": "http://userserve-ak.last.fm/serve/252/15470717.jpg"
    },
    {
        "image_format": "jpeg",
        "is_cached": true,
        "md5sum": "0b302fda58e94757b1a3b5cf7d46e31a",
        "provider": "lastfm",
        "size": 18836,
        "url": "http://userserve-ak.last.fm/serve/126/2127140.jpg"
    }
],
"title": ""
}
```

Das zurückgelieferte Json Objekt enthält neben den Links zu den gewünschten Metadaten, das Dateiformat, MD5-Summe, den Provider u.a. Daten, je nach Anfragetyp. Das Json Format eignet sich sehr gut zur maschinellen Verarbeitung und sollte deshalb bei der Implementierung von moosrdata in den eigenen Player dem HTML Format vorgezogen werden.