Key Value Storage Plugin

Das plugin stellt Slots bereit, mit dennen man binäre Daten einem eindeutigen String zuordnen kann. Technisch Basis ist MapDb.

Konfiguration

Zu Konfuguration des Plugins dient das file "keyValueStorePlugin/src/main/resources/settings.conf". Hier müssen alle Werte bzgl. der Konfuguration der Pluginkonfiguration gepflegt werden.

Dazu sind die Sektionen

- appServerEndpoint
- akkaServer

notwendig

Desweitern muß ein Specherplatz im Dateisystem für die Persistenz des KeyValueStores angegebe werden. Die geschieht durch plugin.filenmame

Es muß ein Pfad mit Zielfile angegeben werden. Dabei ist unbedingt darauf zu achten das das Verzeichnis in dem sich das Zielfielbefindet die gleich Rechte hat wie der Benutzer, der das plugin ausführt. Sonst können keine Werte gespeichert werden.

Installation

das plugin kann als jar ausgeführt werden "java -jar keyvaluestoreplugin-assembly-0.1.jar" oder direkt durch "sbt run" gestartet werden.

Daraufhin verbindet sich das Plugin automatisch mit dem in den settings konfigurieren AppServer und registriert sich auch dort. Die Slots und die REST Services des Plugins sind dann ab sofort verfügbar.

Vorrausetzung hierfür ist natürlich ein das sie AppServer Instanz vollständig hochgefahren ist.

Das Plugin medldet sich auch automatisch ab, wenn es durch ein SigTerm beendet wird.

Einzig wenn der AppServer beendet wird, beendet sich das plugin nicht automatisch. Bei einem restart des Apservers also muss das Plugin auch erneut gestartet werden.

Services

AppServerSlot

folgeden Slots werden zur aussschießliche Verwendung durch den AppServer bereitgestellt:

get Datensatz holen.

in :

```
{
    "key" : "<searchkey>"
}
```

out :

```
{
    "success" : true,
    "result" : "<base64Endcoded binary result>"
}
```

```
{
    "success" : false,
    "message" : "ERROR MESSAGE"
}
```

put Datensatz anlegen. Dabei wird ein eventuell existierenden Datensatz überschrieben.

in :

```
{
    "key" : "<searchkey>",
    "content" : "<base64Endcoded binary content>"
}
```

out : N/A

delete Löscht einen Datensatz.

in :

```
{
    "key" : "<searchkey>"
}
```

out : N/A

exists Püft ob es einen datensatz mit dem betreffenden Schlüssel gibt.

in :

```
{
    "key" : "<searchkey>"
}
```

out :

```
{
    "success" : true,
    "result" : {
        "found" : true | false
    }
}
```

gracefullyShutdown beendet das plugin

in : N/A

out : N/A

Post Rest Services Slots

gethttp Datensatz holen.

in :

```
{
    "key" : "<searchkey>"
}
```

out :

```
{
    "success" : true,
    "result" : "<base64Endcoded binary result>"
}

{
    "success" : false,
    "message" : "ERROR MESSAGE"
}
```

puthttp Datensatz anlegen. Dabei wird ein eventuell existierenden Datensatz überschrieben.

in :

```
{
    "key" : "<searchkey>",
    "content" : "<base64Endcoded binary content>"
}
```

out : N/A

deletehttp Löscht einen Datensatz.

in :

```
{
    "key" : "<searchkey>"
}
```

out : N/A

existshttp Püft ob es einen datensatz mit dem betreffenden Schlüssel gibt.

in :

```
{
    "key" : "<searchkey>"
}
```

out :

```
{
    "success" : true,
    "result" : {
        "found" : true | false
    }
}
```

gracefullyShutdownhttp beendet das plugin

in : N/A

out : N/A