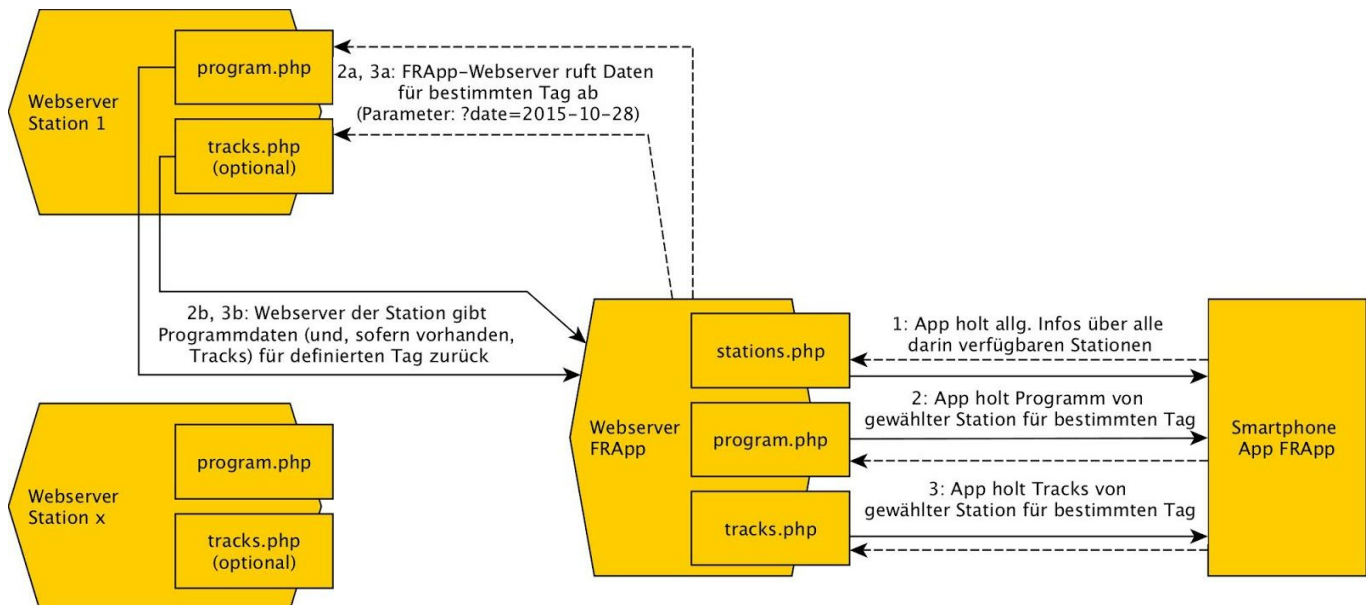


FRApp Datenanbindung v0.1

Prinzip Datenanbindung



Jede Station die in FRApp aufscheinen soll braucht also

1. einen (öffentlich zugänglichen) **Webserver**, darauf:
 - a. **Programmabfrage** mit Parameter **“date”** (im Bild: `program.php`)
 - b. *optional*: **Trackabfrage** mit Parameter **“date”** (im Bild: `tracks.php`)
2. **korrekt eingetragene Daten im FRApp-Backend** des FRApp-Webserver:
 - a. Kontaktdaten, Senderinfos, Frequenzen
 - b. URL Live-Stream (MP3)
 - c. URL Programmabfrage (im Bild: `program.php`)
 - d. *optional*: URL Trackabfrage (im Bild: `tracks.php`)

Regeln für JSON-Output (program.php / tracks.php)

1. Rückgabe: **UTF-8**, kein **HTML-Escaping**:

richtig: <code>ausgewählte Musik</code>	falsch: <code>ausgew&auml;l;hlte&nbsp;Musik</code>
---	--

2. Content-Type des Outputs: `application/json`

3. Request-Parameter `date` muss berücksichtigt werden (um Daten ausschließlich für einen bestimmten Kalendertag zurückzugeben)

4. zulässige **HTML-Tags** (alle anderen werden in der App entfernt, auch `..):`

- `
` bzw. `
` (Zeilenumbruch)
- `.. (Fetttext)`
- `.. (Fetttext)`

5. einzelne Einträge für Kategorien, Sendereien und Sendungen sind **immer indiziert** und nicht assoziativ:

<p>richtig (indiziert, also mit Index):</p> <pre>"categories": [{ "id": "cat0", "color": "FF0000", "namedisplay": "ausgewählte Musik", "description": "Beschreibung0" }, { "id": "cat1", "color": "00FF00", "namedisplay": "Politik", "description": "Beschreibung1" }]</pre>	<p>falsch (assoziativ, also mit Key):</p> <pre>"categories": { "cat0": { "id": "cat0", "color": "FF0000", "namedisplay": "ausgewählte Musik", "description": "Beschreibung0" }, "cat1": { "id": "cat1", "color": "00FF00", "namedisplay": "Politik", "description": "Beschreibung1" } }</pre>
---	--

6. Farbwerte (Kategorien, Sendereien) sind immer als Hex-Codes anzugeben:

richtig: <code>FF0000</code>	falsch: <code>#FF000</code> oder <code>rgb(255, 0, 0)</code>
------------------------------	--

7. Links werden in der App automatisch klickbar gemacht (alle anderen werden entfernt):

- `http://` bzw. `https://`
- `name@domain.com`

8. **Keys** für assoziative Arrays sind **immer in Kleinbuchstaben**

Programmdaten (program.php)

Dieses PHP-File gibt alle

1. Kategorien (falls vorhanden),
2. Sendereihen (zwingend notwendig) und
3. Sendungen (falls vorhanden)

für einen bestimmten Tag im JSON-Format zurück.

Beispiel für program.php: [Output*](#) | [Quellcode](#)

** um JSON im Browser lesbar anzuzeigen: zuerst [Firefox-Addon](#) bzw. [Chrome Extension](#) installieren*

Schema Programmdaten

Die Programmdaten müssen einem bestimmten Schema entsprechen, um von der App korrekt ausgelesen und darin richtig dargestellt werden zu können. Grundsätzlich sieht das Schema so aus:

Programm

- **Kategorien**
 - Kategorie 1
 - Kategorie-ID
 - Farbe
 - Bezeichnung
 - Beschreibung
 - Kategorie 2
 - (gleiche Felder wie Kategorie 1)
 - ...
- **Sendereihen**
 - Sendereihe 1
 - Sendereihen-ID
 - Kategorie-ID (Zugehörigkeit zu Kategorie)
 - Bezeichnung
 - Beschreibung
 - Ausstrahlungs-Infos
 - Farbe
 - Webseite
 - Bild
 - Meta-Infos
 - Sendereihe 2
 - (gleiche Felder wie Sendereihe 1)
 - ...
- **Sendungen**
 - Sendung 1
 - Sendungs-ID
 - Sendereihen-ID
 - Bezeichnung
 - Beschreibung
 - Startzeitpunkt
 - Endzeitpunkt
 - Website
 - Link zu MP3 (für vergangene & archivierte Sendungen)
 - Sendung 2
 - (gleiche Felder wie Sendung 1)
 - ...

Kategorien (“categories”)

- Eine Kategorie entspricht einer **Art von Sendereihe**, wie etwa **“Politik”** oder **“Kultur”**.
- Beispiel: [Kategorien von Radio FRO](#)
- Sie können **auch händisch eingetragen** werden (im PHP-File `program.php`), da es sich um Daten handelt, die sich im Laufe der Zeit nicht großartig ändern bzw. **relativ statisch** sind.
- Kategorien müssen **nicht zwingend** vorhanden sein, wenn sie nicht definiert sind, sind aber folgende Features für die Station nicht in der App vorhanden:
 - Kategorien-Übersicht im Tab “Programm”
 - Kategorie-Detailansicht im Tab “Programm”
 - Farben für Sendereihen im Tab “Programm”

Datenschema für Kategorien

- jede Kategorie ist ein assoziatives (key-value) Array mit bestimmten **keys** (* = Pflichtfeld):
 - `id` * eindeutige, gleichbleibende ID für die Kategorie
 - `color` Farbe
 - `namedisplay` * Bezeichnung
 - `description` Beschreibung

Sendereihen (“series”)

- Eine Sendereihe ist ein **Eintrag im Wochen- oder Monatsprogramm einer Station**.
- Durch die Wiederholung in einem bestimmten Zyklus sind Sendereihen (im Gegensatz zu Sendungen, siehe unten) nicht mit eindeutigen Zeitpunkten versehen, sondern mit Informationen zu deren Ausstrahlungszeitpunkten (wie etwa **“Jeden 1. Montag, 10 - 12 Uhr”**).
- Sie können **auch händisch eingetragen** werden (im PHP-File `program.php`), da es sich um Daten handelt, die sich im Laufe der Zeit nicht großartig ändern bzw. **relativ statisch** sind.
- Sendereihen können in der App **abonniert** werden, die damit einhergehenden **Erinnerungen** erfolgen nach dem selben Prinzip wie die Ausstrahlungszeitpunkte selbst - für das Beispiel werden User etwa **“Jeden 1. Montag, 10 Minuten vor 10 Uhr”** erinnert.
- Sendereihen müssen **zwingend** in den Programmdateien vorhanden sein.

Datenschema für Sendereihen

- jede Sendereihe ist ein assoziatives (key-value) Array mit bestimmten **keys** (* = Pflichtfeld):
 - `id` * eindeutige ID
 - `categoryid` ID der Kategorie, der Sendereihe zugehörig ist (falls vorhanden)
 - `color` Farbe (wenn nicht definiert, wird - falls vorhanden - Farbe der Kategorie verwendet)
 - `namedisplay` * Bezeichnung
 - `description` Beschreibung
 - `url` Link zur Website der Sendereihe
 - `image` Link zu Bild der Sendereihe
 - `broadcastinfos` Ausstrahlungszeitpunkt(e) der Sendereihe
(Achtung: nur Sendereihen mit broadcastinfos werden in der App angezeigt)
 - `metainfos` Zusatzinfos für die Sendereihe: assoziatives (key-value) Array

Broadcastinfos

Jeder einzelne Eintrag eines Ausstrahlungszeitpunkts von Sendereihen muss folgendem Schema entsprechen, **mehrere Einträge selbst werden Strichpunkte (;) getrennt**

Wiederholungen	Abtrennung	Wochentag(e)	Abtrenn.	Startzeit	“bis”	Endzeit	“Uhr”
optional	optional	zwingend	zwing.	zwing.	zwing.	zwing.	zwing.
Wiederholung	: oder jeweils:	jeden Sonntag oder jeden 1. Montag oder jeden 2. Dienstag oder jeden geraden Montag oder jeden ungeraden Dienstag oder werktags oder taglich	,	10 oder 10:00	- oder bis	12 oder 12:00	Uhr

Beispiele:

richtig	falsch
Jeden 1. Sonntag, 10 - 12 Uhr	Jeden 1. Sonntag, 10 - 12 ("Uhr" hinten fehlt)
Jeden Dienstag, 20 - 21 Uhr	dienstags, 20 - 21 Uhr ("dienstags" wird nicht erkannt)
Wiederholung: Mittwoch, 10 - 12 Uhr	Wiederh.: Mittwoch, 10 - 12 Uhr ("Wiederh." wird nicht erkannt)
Jeden 1. Sonntag, 10 - 12 Uhr, Jeden 3. Sonntag, 10 - 12 Uhr	Jeden 1. Sonntag, 10 - 12; Jeden 3. Sonntag, 10 - 12 ("Uhr" hinten fehlt, Beistrich statt Strichpunkt)

Metainfos

- Metainfos sind Zusatzinfos zu Sendereihen, wie etwa ProduzentIn, Kontakt-Email oder alles andere, das gesondert in der App (also nicht in Bezeichnung & Beschreibung) aufscheinen soll.
- Beispiel siehe Sendereihe mit ID "series1" im Programminfo-Dummy ([Output](#) | [Quellcode](#))

Sendungen (“shows”)

- Eine Sendung stellt **Zusatzinformationen zu einer Sendereihe** bzw. **zu einem bestimmten Zeitpunkt** dar.
- Ein Beispiel wäre hier etwa eine Sondersendung zur Landtagswahl in OÖ im Oktober 2015.
- Sendungen müssen **nicht zwingend** in den Programinfos vorhanden sein.

Datenschema für Sendungen

- jede sendung ist ein assoziatives (key-value) Array mit bestimmten **Keys** (* = *Pflichtfeld*):
 - **id** eindeutige ID
- wenn nicht vorhanden, wird diese in der App auf Basis von **datetimestart** berechnet
 - **seriesid** ID der Sendereihe, der Sendung zugehörig ist
- wenn nicht vorhanden, wird diese in der App auf Basis von **datetimestart** einer Sendereihe zugeordnet
 - **namedisplay** Bezeichnung, falls nicht vorhanden wird Bezeichnung der zugehörigen Sendereihe verwendet
 - **description** * Beschreibung - muss definiert werden, da sie ja logischerweise von der Beschreibung der Sendereihe abweicht
 - **datetimestart** * Startdatum & -zeit, Format: DD.MM.YYYY HH:MM:SS
 - **datetimeend** * Enddatum & -zeit, Format: DD.MM.YYYY HH:MM:SS
 - **url** Link zu Website der (Sonder-)Sendung
 - **urlmp3** Link zu archiviertem MP3 der Sendung

zu urlmp3:

- wurde eine Sendung archiviert und als MP3 am Webserver abgespeichert, kann man mittels Angabe einer URL zu dieser Datei in der App ermöglichen, diese nachzuhören (bzw. sie dort auch offline verfügbar zu machen)

Tracks (tracks.php)

Dieses PHP-File gibt alle gespielten Tracks für einen bestimmten Tag im JSON-Format zurück.

Beispiel für tracks.php: [Output*](#) | [Quellcode](#)

** um JSON im Browser lesbar anzuzeigen: zuerst [Firefox-Addon](#) bzw. [Chrome Extension](#) installieren*

Datenschema für Tracks

- jeder Track ist ein assoziatives (key-value) Array mit bestimmten **Keys** (* = *Pflichtfeld*):
 - **datetimestart** * Startdatum & -zeit, Format: DD.MM.YYYY HH:MM:SS
 - **artist** Name des/der Artists
 - **title** Name des Tracks