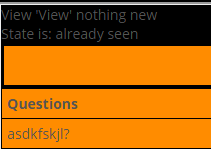
**Funktionalitäten**

Session

Sessions werden beim Login, sowie auch beim Chat verwendet.

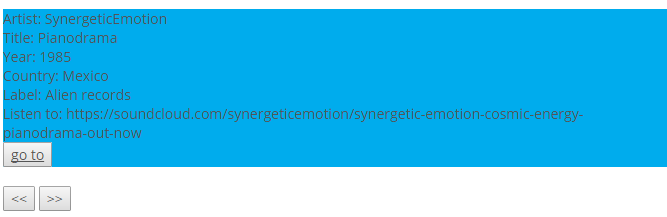
Cookie

Gibt an, ob die neusten Fragen bereits gesehen wurden oder nicht. Wird auf der main verwendet

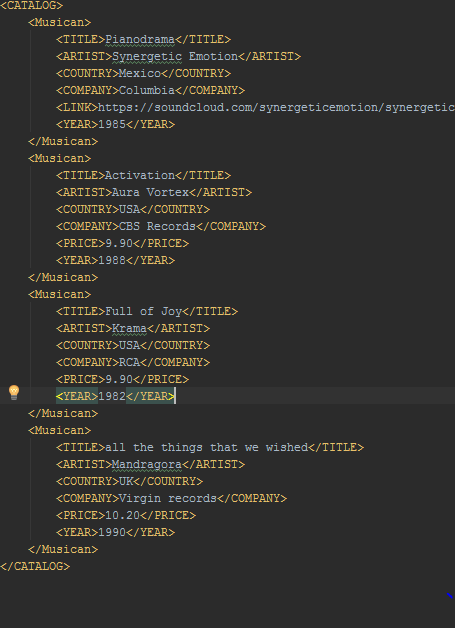


Ajax-Request

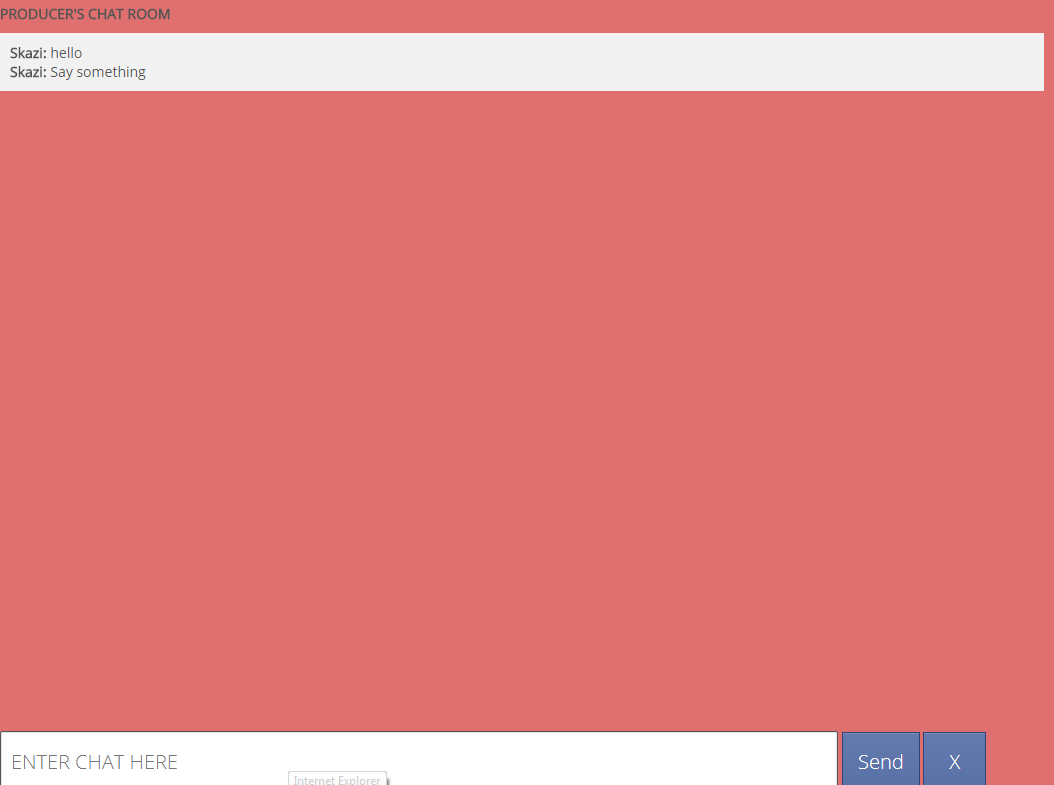
Werden auf «artists» verwendet, um ohne neues Laden zwischen den Artisten zu wechseln. Diese wiederum sind in einem .xml file gespeichert



- .XML Dokument



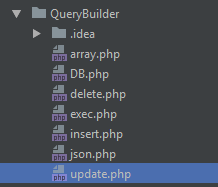
Ausserdem wird ajax für den Chat verwendet:



(Problem: der Chat funktioniert leider nachwievor nur richtig, wenn er einzeln gestartet wird->chatindex.php)

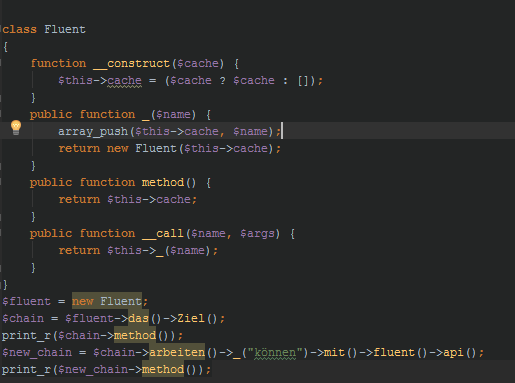
Query Builder

Dieser ist in der Lage Daten der DB als Array oder JSON auszugeben, Daten zu updaten oder zu löschen und neue Daten hinzuzufügen.



Fluent API

C:\Users\lucas.wymann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\fluentapi.png

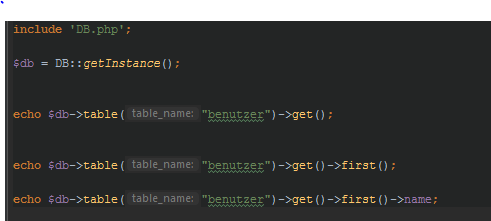


JSON

Auf « profile » lassen sich Benutzerinformationen als JSON ausgeben.

(Query Builder JSON: Der Query Builder ist ebenfalls in der Lage JSON auszugeben)

QueryBuilder:



Profile:

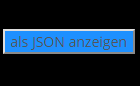


Bild hochladen und verwenden

Auf «images» (imgUploader.php) ist es möglich in der Bildergalerie zu stöbern und auch eigene Bilder hochzuladen (diese werden automatisch der Bildergalerie hinzugefügt.)

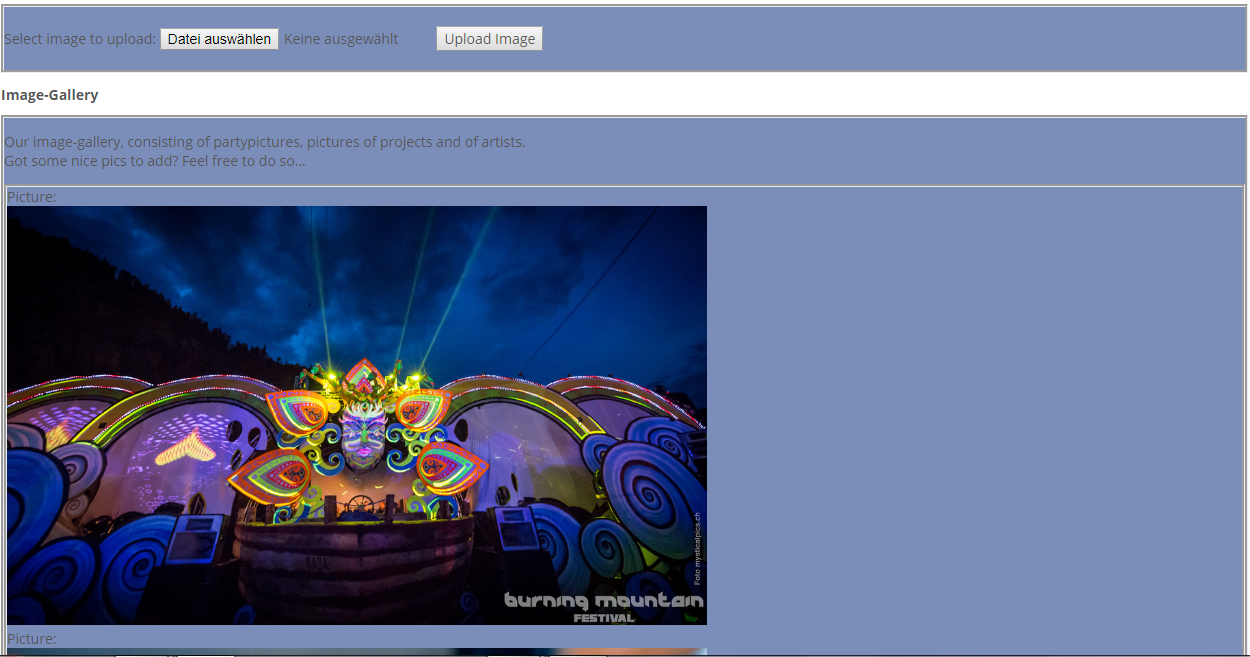
Wenn ein eigenes Bild hochgeladen werden soll, werden diverse Überprüfungen gemacht:

-Handelt es sich wirklich um ein Bild?

-Existiert das Bild bereits?

-Ist das Bild von einem unterstützten Datentyp?

-Konnte das Bild hochgeladen werden?



Klassenbasiert

Der Aufbau des Projekts ist in diverse Unterkategorien und Klassen unterteilt, um das Ganze möglichst erweiterbar zu gestalten.

Router

Der Router kümmert sich um die Verwendung der /user/ Aufrufe.

Responsiveness

Die Browserseite lässt sich nach Belieben vergrössern, bzw. verkleinern und passt sich dem entsprechend an.

