Stand 09.03.2021 Ansgar

# einrad.hockey
# Allgemeine Dokumentation

einrad.hockey verwendet sein eigenes Framework. Dieses wurde von mir entwickelt, um Interessierten im Einradhockey nach einer möglichst kurzen Einrichtungs- und Einarbeitungsphase die Möglichkeit zu geben, an der Website mitzuarbeiten. Dafür sind nur grundlegende PHP- und/oder HTML/CSS-Kenntnisse notwendig, welche man sich schnell aneignen kann.

## Grundlegende Struktur auf dem Server:

| | |
|---|---|
| _localhost | Enthält wichtige Dateien für den Localhost, wie zB die Datenbank-Vorlage oder eine init_set.php Datei, welche PHP-Einstellungen für den Localhost vornimmt. |
| Classes | Dieser Ordner enthält alle Klassen, welche alle Funktionen bereitstellen. Jede SQL-Abfrage sollte unbedingt in einer dieser Klassen geschehen. Klassen werden in init.php automatisch geladen und müssen nicht included werden. Jede Klasse hat eine eigene Datei welche klassenname.class.php heißt. |
| frameworks | Enthält externe php-Frameworks. Diese werden manuell aufgerufen – zB für den E-Mail-Versand oder die Erstellung von Captchas |
| logic | Enthält Dateien für die Ablauflogik (Aufrufen der Klassen, Vorbereitungen zur Darstellung des HTML-Codes) welche immer wieder auftreten. |
| templates | Enthält HTML -Templates, welche wiederholt gebraucht werden. |
| public | Die Web-Root. Enthält die Js-, Css-Dateien, sowie Uploads, Bilder und Dokumente.<br>Wichtige Verzeichnisse.<br>liga -> öffentlicher Bereich der Website<br>ligacenter -> Ligacenter-Bereich<br>teamcenter -> Teamcenter-Bereich<br>*xml -> xml-Schnittelle für externe Anwendungen* |
| System | Enthält systembezogene Dateien: Backups und Logdateien. |
| init.php | init.php muss in jedes PHP-Skript welches in Public aufgerufen wird als erstes eingefügt werden. Es beinhaltet PHP-Einstellungen, Autoloader, Logs, Fehlerbehandlung, Security-Header etc. |
| example_env.php | Beispiel für die Einstellung der Entwicklungsumgebung auf XAMPP. |
| env.php | Enthält wichtige Einstellungen der Entwicklungsumgebung. Muss vom Entwickler selbst erstellt werden vor Verwendung. Siehe Anleitung in example_env.php. |

Stand 09.03.2021 Ansgar

# Jede PHP-Datei in public/liga, public/teamcenter, public/ligacenter welche vom User über die URL aufgerufen wird, hat die gleiche Struktur:

*////////Logik (Controller)////////*

Zuerst kommt der Logikbereich: Hier muss als erstes require first.logic.php für zb. den Autoloader der Klassen kommen. Als nächstes require session_la/team.logic.php für die Authentification für das Teamcenter oder Ligacenter (beides per require, nicht include). Anschließend alles was an Logik benötigt wird, um den angeforderten View des Users darzustellen zu können oder die Datenbank mit Hilfe der Funktionen der jeweiligen Klasse zu verändern (Formularverarbeitung). Im Logikteil sollte kein HTML-Code stehen – auch nicht als String.

*////////Layout (View)////////*

Anschließend kommt das Layout. Hier wird zunächst mit include header.tmp.php der Header, die Navigation und Info-/Fehler-/Hinweis-/Debug-Meldungen geladen. Anschließend werden Templates included, bzw der notwendige html-Code reingeschrieben. Hier werden mit Hilfe eines Arrays alle benötigten Daten aus der Logik (zb. Turnierdetails aus der DB) bereitgestellt. Hierbei wird sich an folgende Vorgehensweise orientiert: https://getkirby.com/docs/cookbook/templating/php-templates. Abschließend wird mit include footer.template.php der Footer eingefügt.

## Wichtige Dateien:

| | |
|---|---|
| **init.php** | init.php muss in jedes PHP-Skript welches in Public aufgerufen wird als erstes eingefügt werden. Es beinhaltet PHP-Einstellungen, Autoloader, Logs, Fehlerbehandlung, Security-Header etc. |
| **header.tmp.php** | Enthält den Header und die Navigation für Desktop und Mobil. Anschließend werden Fehler- und Infomeldungen dargestellt. |
| **footer.tmp.php** | Enthält den Footer |
| **session_la.logic.php** | Diese Datei muss required werden. Wenn sie eingefügt wird, kann ein User auf den Inhalt nur zugreifen, wenn er im Ligacenter angemeldet ist. |
| **session_team.logic.php** | Diese Datei muss required werden. Wenn sie eingefügt wird, kann ein User auf den Inhalt nur zugreifen werden, wenn er im Teamcenter angemeldet ist. |
| **env.php** | Wichtige Einstellungen für die Entwicklerumgebung, um die Seite zum Laufen zu bringen. |
| **W3.css** | Ein simples aber erstmal effektives Css-Framework (https://www.w3schools.com/w3css/default.asp)<br>Dieses sollte man sich anschauen, wenn man HTML-Code auf der Seite schreibt. |

Stand 09.03.2021 Ansgar

## Brauchbare Klassen:

**Alle Datenbankzugriffe werden in einer der Klassen vorgenommen. Außerhalb von Klassen ist es theoretisch möglich Querys vorzunehmen, dies sollte aber nicht getan werden.**

| | |
|---|---|
| **config.class.php** | Erhält Konstanten, welche für den Ligabetrieb wichtig sind. Zb was die aktuelle Saison ist (Config::SAISON === 26) => true für 2020/2021 |
| **feiertage.class.php** | Klasse zum Erkennen von bundesweiten Feiertagen in Abhängigkeit vom Saisonjahr. |
| **db.class.php** | Klasse um Datenbank-Querys auszuführen. Datenbankabfrage-Funktionen werden aus dbwrapper.class.php geladen. |
| **html.class.php** | Klasse zum Einfügen von HTML-Code |
| **helper.class.php** | Hilfsfunktionen, welche das Leben einfacher machen. |
| **nav.class.php** | Hier kann man die Links der Navigationsmenüs ändern. |
| Und viele weitere Klassen. Die Klassen und Funktionen sind alle Kommentiert. Also schaut am besten zB in die Klasse Turniere, falls ihr zb Daten über Turniere braucht. | |

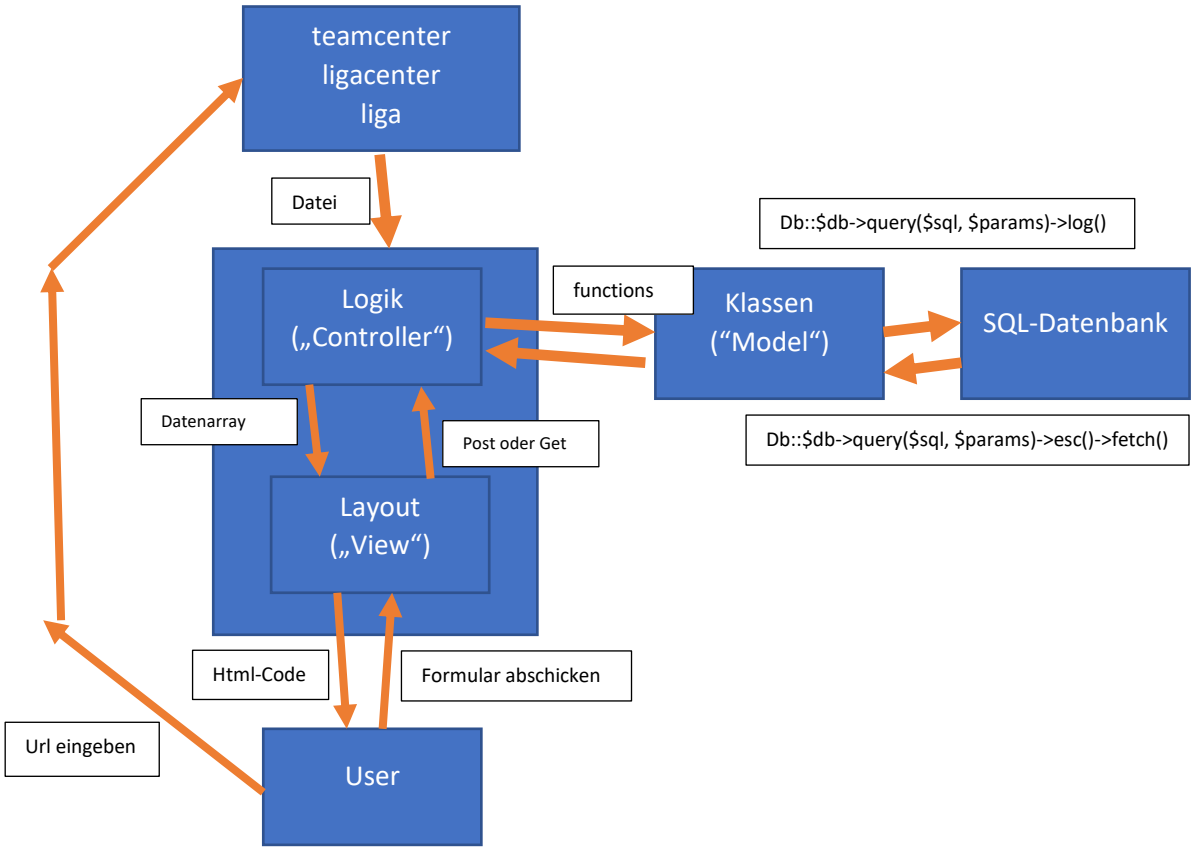
## Brauchbare Funktionen:

| | |
|---|---|
| **db::escape($foo)** | Escape von returns gegen css-hacking)<br>db::escpae($_SERVER[PHP_SELF]) ist zum Beispiel sehr wichtig |
| **db::debug($foo)** | Sehr sehr nützliche Funktion. Erstellt eine Übersicht der Variable $foo. Sehr sinnvoll um die Struktur der Daten-Arrays zu erfassen, mit welchen die Templates gefüttert werden. Gibt auch File und Line der Variable aus. |
| **db::$db->query($sql, …$params)…** | Führt die in $sql als String gespeicherte SQL-Befehl als Prepared Statement mit den Parametern $params. Siehe **dbwrapper.class.php** für die verfügbaren Funktionen, oder sie dir Beispielcode (teams.class.php oder turnier.class.php) an. |

In _localhost/tutorial.php gibt es ein Tutorial mit einer Einleitung wie man eine Seite auf der Website erstellt und weitere wichtige Funktionen.

Stand 09.03.2021 Ansgar

# Funktionsweise der Webseite




Beispiel für die Funktionsweise der Webseite zum Anzeigen des Teamkaders:

1. Der User gibt eine Url in seinem Browser ein.
2. Die Url verweist auf eine PHP-Datei, zum Beispiel einrad.hockey/teamcenter/tc_kader.php
3. Diese PHP-Datei hat einen Logik-Teil und einen Layout-Teil. Im Logik-Teil werden alle notwendigen PHP-Vorgänge vorgenommen, also die Validierung, ob man im Teamcenter angemeldet ist und anschließend ruft sie die Funktion Spieler::get_teamkader($_SESSION[‚logins ‘][‚team‘][‚id‘]) aus der Klasse Spieler auf, welche ein assoziatives Array mit dem Teamkader des angemeldeten Teams zurückgibt. Diese Funktion greift wiederrum mit der Klasse db auf die SQL-Datenbank zu, um die notwendigen Daten zu bekommen.
4. Anschließend kommt der Layout-Teil des in der URL aufgerufenen PHP-Dokuments: Hier wird zuerst das Template der Navigation geladen. Anschließend kommt der HTML-Code/Template mit der Tabelle für die Darstellung des Kaders. In diesem Html-Code werden jetzt die Einträge des assoziativen Arrays mit den Spieler-Daten eingefügt. Abschließend wird das Template des Footers in die Webseite eingefügt.
5. PHP gibt jetzt eine HTML-Seite an den User aus.