

PLUGINS

Pluginit ovat helposti lisättäviä ja poistettavia ominaisuuksia. Pluginin asentaminen onnistuu siirtämällä plugin kansioon /plugins.

Pluginien rakenne ja luominen

Ensimmäiseksi uutta pluginia luodessa luo uusi kansio /plugins kansion sisään. Luomaasi kansioon tarvitset välttämättömänä config.js tiedoston.

config.js

ESIMERKKI 1.

```
module.exports = {  
  
  Name: "Testi plugin", // PLUGININ NIMI  
  Description: "Testausta varten", // PLUGININ KUVAUS  
  Versio: "1.0", // VERSIO  
  
  Temp: [  
  
    ],  
  
  Hooks: [  
  
    [  
      "MainTable",  
      "END_BEFORE_SCRIPT",  
      "../plugins/meal_ticket/hooks/loop_hook.html"  
    ],  
  ],  
  App: [  
  
    [  
      "/plugin.js",  
      "/meal_ticket"  
    ],  
  ],  
}
```

module.exports mahdollistaa että tiedoston variabellet on mahdollista saada toisten tiedostojen käyttöön.

Temp jätetään tyhjäksi. Temp kohtaa käytetään kun config.js exportataan applikaation käytettäväksi. Temp kohtaan tallennetaan databasesessa olevaa tietoa kiinnitysvaiheessa.

Hooks sisältää ns. koukut jotka kiinnitetään alkuperäiseen client puolen tiedostoihin ejs jälki javascriptilla (EJS -> Embanded JavaScript). Myöhemmin mahdolliset hook kohdat ja esimerkki kuinka lisätä uusia hook kohtia clientin puolelle. Ensimmäinen string kertoo mihin tiedostoon hook liitetään, toinen (jos sisältää 3 string) kertoo mihin kohtaan tiedostoa hook kiinnitetään ja viimeinen string kuvastaa aina liitettävää tiedostoa. Suosittelen käyttämään aina juuri polkua.

App on taas serveri puolelle kiinnitettävät koukut. Ensimmäinen string kuvastaa tiedostoa joka liitetään serveriin ja seuraava string kuvastaa polkua johon yhdistämällä toimintoja voidaan suorittaa. Esimerkki tapauksessa esim www.testi.fi/admin/meal_ticket toimii validina osoitteena.

Mahdolliset Hookit:

| | | |
|---------------|---|---------------------------------------------|
| Navbar | → | ["", ""] |
| Modify-client | → | ["", "CLIENT_ACTIONS,END, TABLE_BTN", ""] |
| ROUTE | → | ["", ""] |
| MainTable | → | ["", "END_BEFORE_SCRIPT, END", ""] |

Plugin kohdan lisääminen alkuperäisiin tiedostoihin.

Sovellus käyttää EJS modulaaristen sivujen luomisessa. EJS mahdollistaa javascriptin ajamisen sivuilla ennen kuin sivu lähetetään clientille.

Pluginit ladataan serverin käynnistettyä arrayhin josta sivua jakaessa ejs tarkistaa onko sivulle liitettävää pluginia.

Jos tarvitset uuden kohdan johon plugin liitetään avaa haluttu tiedosto ja lisää haluamaasi kohtaan:

```
<% if(plugins) { %>
<%for(var i = 0; i < plugins.length; i++){ %>
    <% if(plugins[i].Hooks != null && plugins[i].Hooks != undefined) { %>
        <% for(var j = 0; j < plugins[i].Hooks.length; j++){ %>
            <% var hook = plugins[i].Hooks[j]; %>

                // hook[0] == ensimmäinen string
                // hook[1] == toinen string
                // hook[2] == kolmas string

                // voit tarkistaa hook[0] == "TÄMÄ SIVU"
                <% if(hook[0] == "Navbar") {%>
                    //JOKO
                    <- include(hook[1])%>
                    //TAI
                    <- include(hook[2])%>
                <% } %>

            <%}%%>
        <%}%%>
    }%>
}<%}%%>
```

Pluginien hallitseminen.

Asetusvalikossa kohdassa **Plug & Play** avautuu ikkuna josta voi laittaa ominaisuuksia käyttöön/pois käytöstä. Klikkaamalla ominaisuuden nimeä ominaisuus vaihtaa tilaansa joko päälle/pois. Osa ominaisuuksista vaatii että serveri käynnistetään uudestaan joten on hyvä vaihdoksen jälkeen painaa **Käynnistä uudelleen!** nappia. Tämä syystä että osa ominaisuuksista sisältää serveripuolen ominaisuuksia jotka serverin käynnistyessä kiinnitetään pää applikaation ala-polkuun admin. Serveri applikaatio käyttää apunaan express js (npm) joka ei mahdollista järkevällä tavalla pää applikaatioon kiinnitettyjen ala-polkujen irrottamista.