Embedded Sensor Cloud

HÖRMANN Daniel - if12b067

THORSTENSEN Benjamin - if12b066

08. Jänner 2014

1200 Wien, Höchstädtplatz

**Inhaltsverzeichnis**

Inhalt

[1 Architektur 3](#_Toc376967019)

[2 Beschreibung 4](#_Toc376967020)

[2.1 FirstForm.cs 4](#_Toc376967021)

[2.2 Html.cs 4](#_Toc376967022)

[2.3 IPlugins.cs 4](#_Toc376967023)

[2.4 LogFile.cs 4](#_Toc376967024)

[2.5 PluginManager.cs 4](#_Toc376967025)

[2.6 Program.cs 4](#_Toc376967026)

[2.7 Request.cs 4](#_Toc376967027)

[2.8 Response.cs 4](#_Toc376967028)

[2.9 Server.cs 5](#_Toc376967029)

[2.10 Url.cs 5](#_Toc376967030)

[2.11 WrongFilenameException.cs, WrongParameterException.cs 5](#_Toc376967031)

# Architektur

Program

Html

Temperatur

Static

Navi

RssFeed

Response

First

Url

Request

Manager

IPlugins

Server

# Beschreibung

## FirstForm.cs

Es wird eine Auswahl aller verfügbaren Plugins an den Browser geschickt.

## Html.cs

In dieser Klasse wird das HTML-Dokument zusammengestellt.   
Es werden CSS-Eigenschaften, JQuery, Javascript und der zu übermittelnde Inhalt zusammengeführt.

## IPlugins.cs

Hier wird das Interface für die verschiedenen Plugins definiert.

## LogFile.cs

Falls ein Fehler im Server auftritt, werden genauere Informationen in einem LogFile gespeichert. Diese sind jeweils mit Datum und Uhrzeit versehen.  
Wenn das LogFile eine bestimmte Größe erreich (zur Zeit: 1MB) wird dieses in einer chronologischen Reihenfolge umbenannt und ein neues LogFile erstellt.

## PluginManager.cs

Mit Hilfe von Reflections werden alle Plugins (von denen die DLL-Datei in einem bestimmten Ordner am Server gespeichert sind) geöffnet und überprüft, ob es sich um das gewünschte Plugin handelt. Dies geschieht dadurch, dass der Pluginname übergeben und mit dem Dateinamen der DLL-Datei verglichen wird. Ist das gewünschte Plugin vorhanden, werden weitere Funktionen im Plugin aufgerufen.

## Program.cs

Es wird die Klasse „Server“ aufgerufen und der Webserver gestartet. Das Programm wartet auf eine Eingabe. Geben Sie das Wort „Exit“ ein (es kann in jeder beliebigen Form geschrieben werden), wird das Programm beendet.

## Request.cs

Die Klasse unterscheidet zwischen einem GET-Request und einem POST-Request. Dementsprechend werden verschiedene Funktionen aufgerufen.

## Response.cs

Diese Klasse wird aufgerufen, wenn etwas vom Server an den Browser geschickt werden soll. Es werden die eigentlichen Inhalte mit einem HTML-Grundgerüst und Headerinformationen versehen.

## Server.cs

In dieser Klasse wird der Server in einem eigenen Thread gestartet.   
Beim Programmstart werden hier die Plugins „Temperatur“ und „Navi“ ohne Parameter gestartet.   
Dadurch wird ein Sensor ausgelesen und die Werte in eine Datenbank gespeichert. Des Weiteren wird ein OSM-File (OpenStreetMap) ausgelesen und intern abgespeichert.

## Url.cs

Diese Klasse wird aufgerufen wenn eine Anfrage von einem Browser zu diesem Server stattgefunden hat und bereits bekannt ist, um welche Art von Anfrage es sich dabei handelt (GET, POST). Es werden dementsprechend verschiedene Funktionen in dieser Klasse aufgerufen um die Inhalte aufzuteilen und für die Plugins zugänglich zu machen.

## WrongFilenameException.cs, WrongParameterException.cs

Diese zwei Klassen dienen dazu, genauere Aussagen über die Fehlerart treffen zu können.