

# PETROL PRICE CALCULATOR

Softwaredokumentation

Von Christian Hahn

## KURZBESCHREIBUNG

Diese Software wurde im Zusammenhang der Code Competition „Zapfsäulenkönig“ erstellt. Es wurde als Programmiersprache C# zusammen mit dem Framework ASP.NET MVC genutzt.

## REQUIREMENTS

Entwicklungsumgebung:

- Windows 7 (64-bit)
- Microsoft Visual Studio 2015
- .NET Framework 4.5.2
- IIS Version 6.1 SP 1

Getestet auf:

- Internet Explorer 11
- Firefox 42
- Chrome Version 51.0.2704.103 m (64-bit)

## ARCHITEKTUR

Grundlegend ist die Solution in 2 große Projekte eingeteilt:

1. Petrolcalculator
2. Petrolcalculator.Core

Im Projekt „Petrolcalculator“ wird die reine Funktionalität zur Ausgabe der Daten implementiert. D.h. in einem MVC Projekt sind dort sämtliche Controller, View Models, Views, CSS-Files und JS-Files zu finden.

Im Projekt „Petrolcalculator.Core“ ist die gesamte Logik für die Aufgabenstellung implementiert worden. Diese wird in einer Servicearchitektur zur Verfügung gestellt. Es gibt für verschiedene Aufgaben, wie Mappings durchführen, Requests senden, Restriktionen überprüfen usw. jeweils eigene Services. All diese Services wurden gemäß ihres Zwecks jeweils unter einer großen Servicefacade mit Interfaces zusammengefasst, um den Zugriff zu vereinfachen und zu vereinheitlichen. Viele Services wurden zusätzlich, wenn es möglich war, generisch aufgebaut. Dies erlaubt z.B. eine einfache Adaption eines Datenmodells, indem dieses bspw. einfach ausgetauscht werden kann. Mit solch einer Architektur kann man sehr dynamisch auf nötige Änderungen der Implementierungen einzugehen, z.B. wenn sich Parameter oder die Rückgabe einer Schnittstelle ändern. Zusätzlich ist der Implementierungsaufwand bei Änderungen oder Zusatzanforderungen deutlich geringer.

Diese Gesamtarchitektur erlaubt es, die jeweils genutzte Ausgabetechnologie wie bspw. MVC, Webforms, WPF, Console etc. einfach zu ändern, da jeweils nur das Projekt „Petrolcalculator.Core“ eingebunden und die jeweilige Servicefacade referenziert werden muss.

## INSTALLATIONSANWEISUNG

Um das ASP.NET MVC Projekt nutzen zu können, ist es nötig dieses auf einem Webserver laufen zu lassen. Im Folgenden werden zwei einfache Wege erläutert, das Projekt aufzusetzen.

## VISUAL STUDIO

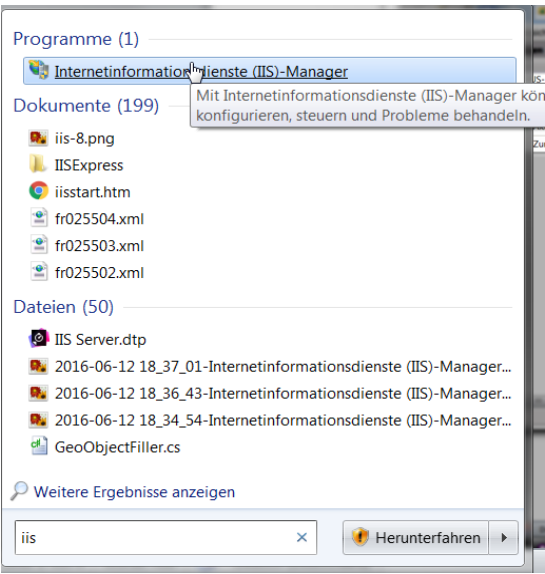
Möchte man das Projekt mittels Visual Studio starten, so nutzt man automatisch den in Visual Studio integrierten Webserver „IIS Express“. Hierzu ist es lediglich nötig das Solution File „\Petrolcalculator\Petrolcalculator.sln“ aufzurufen. Hierdurch öffnet sich Visual Studio und lädt die gesamte Solution. Daraufhin reicht es die Option „Start Without Debugging“ (Strg + F5) auszuwählen. Es öffnet sich die Website im ausgewählten Browser.

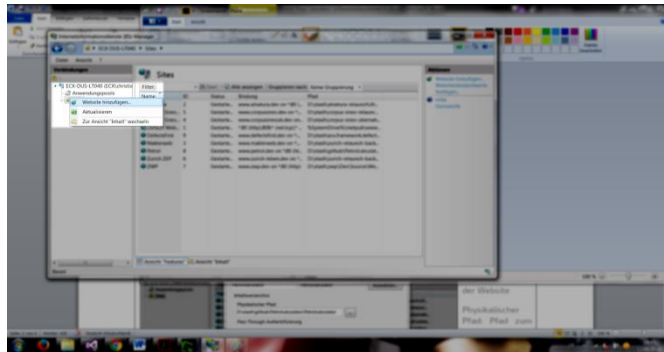
### Wichtig:

Sofern man gerade eine .cshtml Datei geöffnet hat, versucht Visual Studio diese Datei im Browser zu öffnen. Dies kann man an der Url, die versucht wird aufzurufen, erkennen. Das wird aber bspw. bei Partial Views nicht funktionieren. Es ist daher ratsam beim Starten der Website **keine** .cshtml Datei im aktiven Tab ausgewählt zu haben, um sicherzustellen, das Visual Studio die Solution mit der vorgesehenen Startseite startet. In der Regel ist dies die Url „localhost:[PORTNUMMER]“

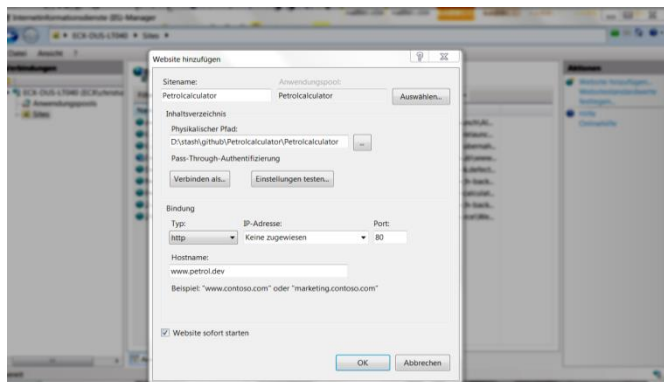
## IIS

Um die Solution mittels des IIS zu starten sind die folgenden Schritte nötig:

Screenshot	Beschreibung
	Öffnen des IIS



Hinzufügen einer neuen Website mit rechtsklick auf „Sites“ und dann „Website hinzufügen“



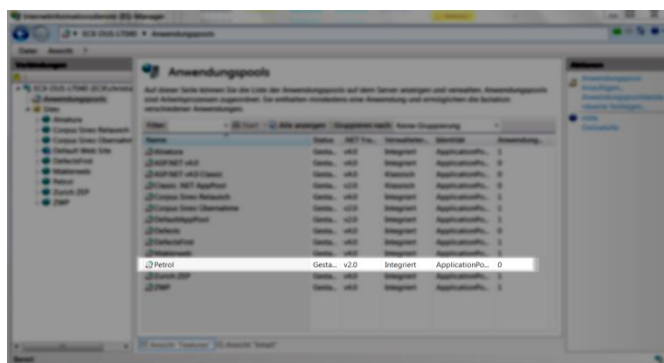
In dem neuen Fenster alle Daten ausfüllen.

Einzustellende Felder sind:

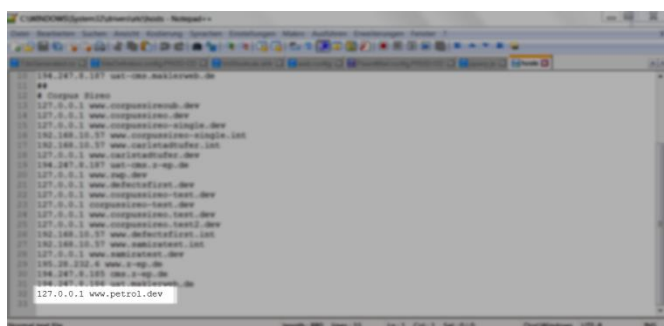
**Sitenname:** Name der Website

**Physikalischer Pfad:** Pfad zum Webprojekt der Website (Hier Petrocalculator auf der gleichen Ebene wie Petrocalculator.Core)

**Hostname:** Url über die die Website später lokal erreichbar sein wird

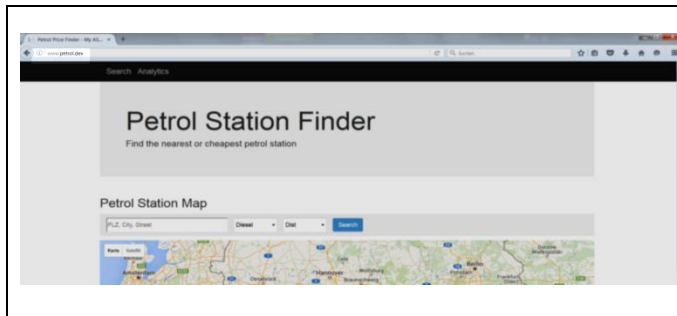


Nach dem Anlegen der Sites innerhalb von „Anwendungspools“ den neuen Eintrag suchen und prüfen, ob dort unter .NET Framework v2.0 eingetragen wurde. Falls dies so ist, Doppelklick auf den Eintrag und auf v4.0 ändern.



Damit die Website nun lokal über den Webbrowser aufrufbar wird, ist es nötig einen Hosteintrag zu machen. Hierfür ist es nötig die Datei c:\windows\system32\drivers\etc\hosts anzupassen. Am Ende der Datei einfach den Eintrag 127.0.0.1 **www.petro.dev** eintragen.

Dabei steht die 127.0.0.1 für den eigenen PC und die Url, ist die Url die man vorher unter Hostname im IIS eingetragen hat.



Nach all diesen Einstellungen ist die Website über den Browser aufrufbar.

## AUFBAU DER UI

Zur Interaktion mit dem System sind zwei verschiedene Seiten bereitgestellt wurden. Die erste Seite „Search Page“ ist die Startseite der Website und dient zum Suchen von Tankstellen und zum Anzeigen der jeweiligen Preise. Die zweite Seite ist die „Analytics Page“ und dient zum Sammeln und Auswerten von Preisdaten zu jeweils eingestellten Tankstellen.

## SEARCH PAGE

### WICHTIG:

Vor der ersten Nutzung müssen in der web.config die Einträge:

- TankerKoenigApiKey
- GoogleMapsApiKey

Angepasst werden, damit die Website funktioniert

Screenshot	Beschreibung
	<p>Grundsätzlich gliedert sich die Search Page in 6 Bereiche:</p> <ol style="list-style-type: none"> <li>1. Navigation</li> <li>2. Header</li> <li>3. Eingabemaske zur Suche</li> <li>4. Google Maps</li> <li>5. Textuelle Ausgabe (Initial leer)</li> <li>6. Footer</li> </ol>
	<p>Über die Sucheingabemaske aus kann man nun nach Tankstellen in seiner Nähe suchen.</p> <ol style="list-style-type: none"> <li>1. Texteingabe</li> <li>2. Dropdownliste zur Auswahl des Benzintyps</li> <li>3. Dropdownliste zur Auswahl zur Sortierung der Resultate</li> </ol> <p>Zu 1.: Hier kann man folgende Eingaben machen: Postleitzahl, Straße und Stadt.</p> <p>Zu 2. Zur Auswahl stehen die Typen: Alle, Diesel, E5 und E10</p>

Zu 3. Zur Auswahl stehen: Distanz und Preis.

**Wichtig:** Wenn man als Benzintyp „Alle“ gewählt hat, steht keine Sortierung nach Preis mehr zur Verfügung, da dies von der benutzten API nicht unterstützt wird.

## Petrol Station Map

Diesel

Dist

Search

Please insert a value for searching

## Petrol Station Map

Diesel

Dist

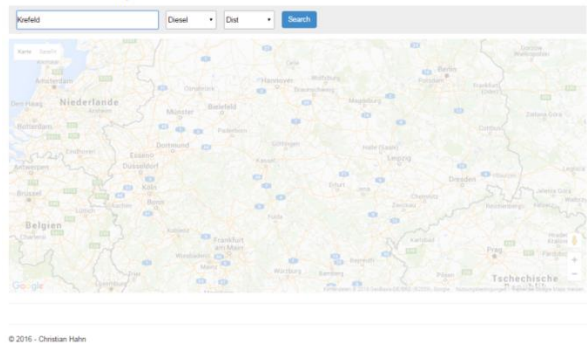
Search

No Geolocation found

Bei der Eingabe und dem Klick auf „Search“ gibt es eine Validierung auf zwei Fehlerfälle:

1. Es wurde keine Eingabe in das Textfeld gemacht
2. Die Eingabe im Textfeld konnte nicht auf eine Geokoordinate gemappt werden.

## Petrol Station Map



Hat man eine erfolgreiche Anfrage gestartet wird die Google Map leicht durchsichtig, um ein Laden der Applikation anzuzeigen.

**Wichtig:** Es kann bei manchen insbesondere der ersten Anfrage zu einer kurzen Wartezeit bei der Geocodinganfrage kommen. Das System arbeitet aber im Hintergrund weiter und antwortet, sobald es fertig ist.

Search Analytics

### Petrol Station Map

Diesel

Dist

Search

Total Number of Results 32 for Diesel sorted by Dist

Price	Distance	Name	Address	Place	ID
1.084 €	0.7 km	JET	COMPOSER STR. 115-129	47708 Krefeld	51180511-4055-1a6d-a150-83030439a7c9
1.089 €	1 km	STAR	Uerdinger Straße 372	47602 Krefeld	505059a-7a65-1a62-b0a6-b1109a20a9d5
1.094 €	1.1 km	Monnet	Neue Mühle Str. 41	47601 Krefeld	10742171-3a61-4452-8587-5a8f744599f5
1.094 €	1.1 km	STAR	Münster Str. 136	47601 Krefeld	505059a-7a65-1a62-b0a6-76730455a2c5
1.099 €	1.3 km	ESSO	PHILADELPHIASTR. 35	47708 Krefeld	7778625-6945-4a77-aaf7b-59a5520214
1.099 €	1.3 km	Shell	DEISSBÜCHER BRÜCKEN 75	47601 Krefeld	76337147-6865-4652-a877b-5a355a17056
1.106 €	1.3 km	Shell	GLÖCKCHENTZ 235	47601 Krefeld	073728a-470a-4945-8a5b-2a4a544a4a62
1.074 €	1.5 km	NA - Markt Autoservice AG	Chadeler Str. 51	47601 Krefeld	1788765a-8c7a-5377-6a65-4655a6a4a652
1.089 €	1.5 km	Shell	MÖRDERER STR. 405	47601 Krefeld	3a6c72a-3953-473b-6a1a-1a3077311106
1.089 €	1.9 km	APUL	Urdenberg 212	47601 Krefeld	80574747-4585-4652-8522-8b2a30a5a584
1.079 €	2 km	Frax	Friedberg 35-41	47601 Krefeld	1a7594d5-148b-47a2-47a3-a91b151535a8
1.089 €	2.1 km	ST-IG Düsseldorf	Hedden 202	47601 Krefeld	48722a4-6865-1a62-b0a6-40578a05a49a
1.084 €	2.2 km	STAR	Uerdinger Straße 555	47601 Krefeld	505059a-7a65-1a62-b0a6-76a6a6a6a6c5
1.074 €	2.2 km	Supermarkt-Tankstelle am rest. Markt	Waldstr. 200	47708 Krefeld	71a4a65f-4055-1a6d-a150-83030439a7c9
1.079 €	2.3 km	JET	GRANIERSTR. 63	47708 Krefeld	51180511-4055-1a6d-a150-83030439a7c9
1.089 €	2.7 km	STAR	Hedden Straße 201	47601 Krefeld	505059a-7a65-1a62-b0a6-76a6a6a6a6c5

Ist das Laden abgeschlossen, so geschieht folgendes:

1. Die Google Map wird wieder normal sichtbar
2. Es werden Marker zu allen gefunden Tankstellen angezeigt
3. Es erscheint eine textuelle Ausgabe der Resultate unterhalb der Google Map
4. Der Screen scrollt soweit herunter, bis der Viewport genau über der Headline über der Eingabe liegt.

**Petrol Station Map**

Krefeld Diesel Dist Search

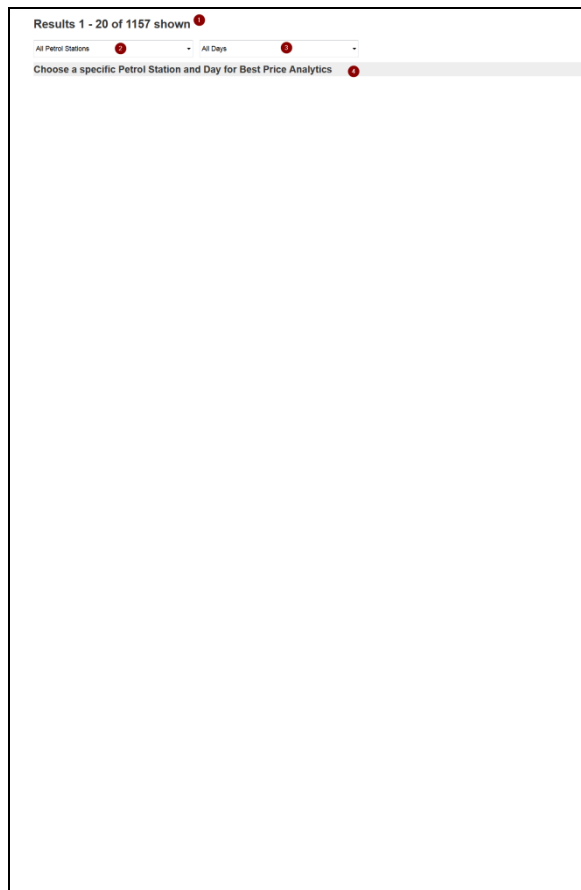
Total Number of Results 32 for Diesel sorted by Dist

Price	Distance	Name	Address	Place	ID
1.089 €	0.7 km	jet	OPPLAUER STR. 115-129	47769 KREFELD	51046541-4095-1a4d-4100-40004019a03a
1.089 €	1 km	CT&E	Umsingstr. 21a/ba 170	47700 Krefeld	2005082a-7a6f-4a42-a0a0-41133a426a46
1.089 €	1.1 km	Markt	Nauen Riber Str. 41	47800 Krefeld	10742121-3a45-4450-4887-50975a43895d

Bei Klick auf einen der Marker erscheinen in einem kleinen Overlay noch Detailinformationen zu dieser Tankstelle, damit man direkt die Preise mit der Position verknüpfen kann.

## ANALYTICS PAGE

Screenshot	Beschreibung
	<p>Grundsätzlich ist die Analyticsseite in 6 Bereiche eingeteilt:</p> <ol style="list-style-type: none"> <li>Navigation</li> <li>Header</li> <li>Eingabemaske für den Datensammlungsprozess</li> <li>Eingabemaske zur Ausgabe der gesammelten Analyticsdaten</li> <li>Ausgabe der Analyticsdaten in Tabellenform</li> <li>Pagingelemente</li> <li>Footerbereich</li> </ol>
	<p>Die Datensammlungseingabemaske unterteilt sich in einen kleinen Einleitungstext und einen Button zum Aktivieren der automatischen Datensammlung.</p> <p>Drückt man diesen Button wird im Hintergrund ein Thread gestartet, welcher auf Basis der Favoriten alle X Minuten den aktuellen Preis anfragt und das Ergebnis abspeichert. Des Weiteren verändert der Startbutton in einen Stopbutton.</p>
	<p>Man kann diesen gestarteten Thread nun manuell wieder stoppen, indem man auf den „Stop Data Collection“ Button drückt. Dieser bezieht seine Information über einen laufenden Datensammlungsthread über die aktuelle Session.</p>



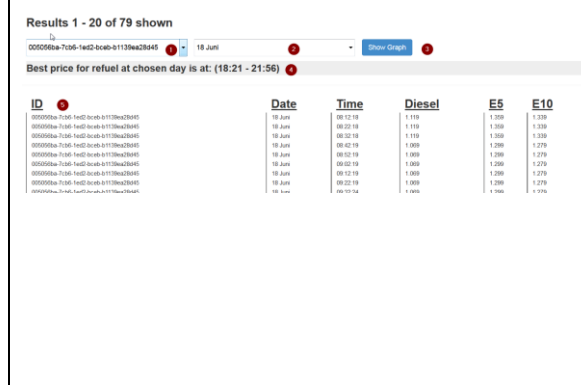
Die Eingabemaske zur Ausgabe der Analyticsdaten hat folgende Elemente:

1. Informationstext über die Anzahl der aktuell ausgegebenen Elemente
2. Dropdownliste zur Filterung nach einer speziellen Tankstelle
3. Dropdownliste zur Filterung nach einem speziellen Tag
4. Ausgabe der Analyticsanalyse (Aktuell Ermittlung der Zeitspanne, an dem das Tanken am günstigsten ist)

Zu 1. Die Anzahl der Elemente unterscheidet sich je nach ausgewähltem Filter

Zu 2. und 3. Die Auswahl einer Tankstelle oder eines Tages generiert sich jeweils dynamisch und basiert auf allen zugrunde liegenden verfügbaren Daten

Zu 4. Diese Ausgabe ist abhängig von den eingestellten Filtern für Tankstelle und Tag. Aktuell ist eine Auswertung nur möglich, wenn eine spezielle Tankstelle und ein spezieller Tag ausgewählt wurde. Sobald ein spezieller Tag ausgewählt wurde, werden alle Zeitspannen mit niedrigstem Preis angegeben.



Hat man bei 1. eine spezielle Tankstelle und bei 2. einen speziellen Tag ausgewählt wird

- ein neuer Button 3. mit dem Text „Show Graph“ angezeigt.
- Die Analyticsausgabe bei 4. Verändert sich zu einer Auswertung der besten Zeit zum Tanken an einem Tag
- Die Anzeige der Resultate in Tabellenform bei 5. passt sich automatisch gemäß der neuen Filterung an.



Hat man den neuen Knopf bei 1. zur Anzeige der Resultate in einem Graphen gedrückt, so erscheint bei 2. unter der Eingabemaske ein kleiner Graph zur Visualisierung der gesammelten Daten. Zudem scrollt die Website so weit herab, bis der neue Graph komplett im Viewport ist

Es sind bis zu 3 verschiedene Kurven zu erkennen:

- Diesel
- E5
- E10

Jeder genommene Messwert ist gesondert als Punkt im Graphen vermerkt. Es kann so zu einem speziellen Zeitpunkt der konkrete Preis angezeigt werden. Möchte man den Graphen wieder verbergen, so kann man den

	Knopf 1. erneut drücken und die Seite scrollt wieder hoch.
--	--



## EINSTELLUNGSMÖGLICHKEITEN

Grundsätzlich hat man einige Einstellungsmöglichkeiten, die man alle in der web.config vornehmen kann.

```
<!-- General Options -->
<add key="Logging" value="true" />
<!-- Request Option -->
<!-- Key für den Zugriff auf die freie Tankerkönig-Spritpreis-API
Für eigenen Key bitte hier https://creativecommons.tankerkoenig.de
registrieren. -->
<add key="TankerkoenigApiKey" value="xxxx" />
<!-- For usage replace this dummy key with you own-->
<add key="GoogleMapsApiKey" value="xxxx" />
<!-- Time in Min less than 60, default is 4 -->
<add key="DelayBetweenTwoIdenticalRequests" value="4" />
<!-- Analytics Options-->
<!-- Time in ms default is 600000-->
<add key="DelayWhileDataCollection" value="600000" />
<!-- Pipe Seperated IDs of Petrol Stations-->
<add key="FavoritePetrolStations" value="005056ba-7cb6-1ed2-bceb-b1139ea28d45|11571341-3296-4f16-a363-28b8c188872c|51d4b683-a095-1aaf" />
<!-- Path for Serialization the Analyticsdata, default is Serialization/AnalyticsDataModel -->
<add key="AnaylticsSerializationPathFromWebRoot" value="Serialization/AnalyticsDataModel" />
<!-- Request Base Urls (Without Parameters)-->
<add key="BaseListUrl" value="https://creativecommons.tankerkoenig.de/json/list.php" />
<add key="BaseDetailUrl" value="https://creativecommons.tankerkoenig.de/json/detail.php" />
<add key="BasePriceUrl" value="https://creativecommons.tankerkoenig.de/json/prices.php" />
<add key="BaseGoogleGeolocationService" value="https://maps.googleapis.com/maps/api/geocode/json" />
<!-- Base Settings -->
<!-- Positive Value in Km, default is 5 -->
<add key="DistanceOfPetrolStationRequest" value="5" />
<!-- Paging -->
<add key="ElementsPerPage" value="20" />
<add key="NumberOfPaingElements" value="5" />
```

- **Logging**  
Flag um das Logging ein- oder auszuschalten (Logs werden unter “/Log/\*.log” gespeichert)
- **TankerkoenigApiKey**  
Eindeutiger Identifier, welcher bei der Nutzung der Tankerkönig API benötigt wird
- **GoogleMapsApiKey**  
Eigener Google Maps Api Key zur Nutzung von Google Funktionalität  
<http://googlegeodevelopers.blogspot.de/2016/06/building-for-scale-updates-to-google.html>
- **DelayBetweenTwoldenticalRequests**  
Zeitspanne in Minuten, innerhalb derer keine identischen Requests erlaubt werden.
- **DelayWhileDataCollection**  
Zeitspanne in ms, die bei der Data Collection gewartet wird, bis erneut nach Preisen zu gespeicherten Tankstellen gefragt wird.
- **FavoritePetrolStations**  
Liste von gespeicherten Tankstellen, die zur Data Collection genutzt werden. Alle Ids werden Pipesepareded (|) angegeben
- **AnalyticsSerializationPathFromWebRoot**  
Pfad vom Websiteroot zur Serialisierungsdatei
- **BaseListUrl**  
Url ohne Parameter von der Tankerkönig-Listenanfrage
- **BaseDetailUrl**  
Url ohne Parameter von der Tankerkönig-Detailanfrage
- **BasePriceUrl**  
Url ohne Parameter von der Tankerkönig-Preisanfrage
- **BaseGoogleGeolocationService**  
Url ohne Parameter vom genutzten Geolocationnservice
- **DistanceOfGeolocationRequests**  
Wert des Radiuses in Km, der bei jeder Suchanfrage genutzt wird
- **ElementsPerPage**  
Anzahl von Elementen die pro Page angezeigt werden

- **NumberOfPagingElements**  
Maximale Anzahl von Pagingelementen, die angezeigt werden

## BASISFEATURE

1. **Wie ist der aktuelle Spritpreis in der Umgebung einer Postleitzahl**  
Dieses Feature wurde über die Suchseite und die dort implementierte Eingabemaske erfüllt. Mit dieser ist es möglich eine Postleitzahl anzugeben und erhält eine Liste von Tankstellen mit jeweiligem Preis.
2. **Zu welcher Uhrzeit ist das Tanken im Schnitt am günstigsten**  
Dieses Feature wurde über die Analyticsseite implementiert. Es wird anhand von gesammelten Preisdaten für eine spezielle Tankstelle zu einem speziellen Tag per textueller Ausgabe dem Nutzer die Information gegeben zu welchen Zeiten am Tag das Tanken am günstigsten war.

### Anmerkung zu Umsetzung

- Diese Preisdaten können über den vorher erwähnten Datensammlungsthread gesammelt werden. Es wurde bewusst darauf verzichtet, die Analyticsdaten aus bisherigen Requests der Suchseite zu nutzen, da diese Daten bzgl. der gegebenen Aufgabenstellung nur sehr beschränkt Aussagekräftig wären und nur eine sehr rudimentäre Datenbasis abgeben würden. Die Daten eines kontinuierlichen Threads, welcher die Preisdaten über den gesamten Tag sammelt, dagegen ist wesentlich Aussagekräftiger. Daher werden nur die Daten des Datensammlungsthreads zur gegebenen Aufgabenstellung genutzt.
- Des Weiteren gilt die Einschränkung, dass nur pro Tag eine Analyse erfolgt, weil nur pro Tag eine verlässliche Aussage über den günstigsten Preis getroffen werden kann.  
Es gab einen experimentellen Ansatz, welcher über alle verfügbaren Tage die jeweiligen Schnittmengen von sich überlappenden Zeitperioden für das jeweils pro Tag günstigste Tanken vornahm.  
Beispiel: Tag A hat eine Zeitperiode von 8 - 12 Uhr und Tag B von 10 - 13 Uhr, wo das Tanken am günstigsten war. Als Ergebnis dieser zwei Tage erhielt man nun 10 - 12 Uhr.  
Dies hatte aber das Problem, dass wenn nur ein Tag dabei war, der aus dem Schema der anderen Tage ausriss, bspw. wenn das günstigste Tanken erst spät abends war, es keinerlei Überlappung mehr gab und die Schnittmengen immer leer waren.  
Leider war in der Kürze der Zeit keine Verbesserung des Ansatzes möglich um auch mit Ausreißern umzugehen, so dass dieser Ansatz entfernt nicht weiter verfolgt wurde.

## ZUSATZFEATURE

1. Suchseite
  - a. Es ist nicht nur eine Suche nach einer Postleitzahl möglich. Durch den genutzten Geolocationsservice ist es sogar möglich eine Stadt oder eine Straße im Suchfeld anzugeben, um die Spritpreise in seiner Umgebung zu ermitteln.
  - b. Zur besseren Übersicht wurden die Ergebnisse in einer Google Map visualisiert. Es werden alle Tankstellen als Marker auf der Karte angezeigt. Zusätzlich kann man mit Klick auf einen speziellen Marker dessen Detailinformationen direkt abrufen.
  - c. Implementierung einer ausgiebigen Facettierungsmöglichkeit bei der Suche
2. Analyticsseite
  - a. Zum Aufbau einer geeigneten Datenbasis wurde ein eigener Datensammlungsthread implementiert, welcher kontinuierlich im Hintergrund Daten für die Auswertung sammeln kann. Diese Art der Datensammlung erzeugt eine wesentlich detaillierte Datenbasis zur späteren Auswertung, als eine einfache Nutzung der Resultate der Requests der Suchseite. Diese wären lediglich sehr sporadisch und kaum für Analyticszwecke einsetzbar.

- b. Es wurde zur besseren Übersicht eine visuelle Ausgabe der gesammelten Analyticsdaten implementiert. Diese kann für eine ausgewählte Tankstelle an einem ausgewählten Tag genutzt werden.
  - c. Implementierung einer ausgiebigen Facettierungsmöglichkeit bei der Ausgabe der Analyticsdaten
  - d. Implementierung eines Pagings, welches die Analyticsdaten sauberer und übersichtlicher darstellt
- 3. Einhaltung der AGB von Tankerkönig (Wichtig hierbei sind besonders die Punkte 7 und 8)
  - a. Der eigene Tankerkönig API-Key wurde durch 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx' ersetzt. Es ist über einen web.config Eintrag sehr leicht möglich den API-Key Applikationsweit auszutauschen
  - b. Es werden keine Preisupdates, wie bei der Analyticsdatensammlung benötigt, mit details.php durchgeführt. Für die Preisupdates wird die dafür vorgesehene prices.php genutzt.
  - c. Es wurde eine Art Caching implementiert, was identische Anfragen innerhalb einer gegebenen Zeitspanne unterbindet. Laut AGB sollten keine identischen Requests innerhalb von 4 Minuten getätigt werden. Die Anzahl der Minuten ist über die web.config einstellbar. Falls eine identische Anfrage innerhalb dieser Zeitspanne reinkommt, so wird der Request unterbunden und die gespeicherten Ergebnisse werden zurückgeliefert.
- 4. Implementierung eines ein- und ausschaltbares Logging über NLog für sämtliche Fehlverhalten der Applikation zur besseren Fehleranalyse