

**Projekthandbuch**Projekt: WebInterface Reisebüro Graf

[1](#_Toc461027628)

[1. Einrichten der Netzwerkumgebung 3](#_Toc461027629)

[1.1. Workstations konfigurieren 3](#_Toc461027630)

[1.2. Domäne (Diagramm) 5](#_Toc461027631)

[1.3. Serverstruktur 5](#_Toc461027632)

[1.4. Aufstufung des DomänenControllers 7](#_Toc461027633)

[1.5. DHCP Server aufsetzen 10](#_Toc461027634)

[1.6. Netzwerkdiagramm(Grafik) 13](#_Toc461027635)

[2. Datenbankentwurf 14](#_Toc461027636)

[2.1. Entity Relationship Model (ERM) 14](#_Toc461027637)

[2.2. Relationensynthese 15](#_Toc461027638)

[3. Datenbankzugriff 15](#_Toc461027639)

[3.1. Entity Framework 15](#_Toc461027640)

[3.2. EDMX 15](#_Toc461027641)

[3.3. Schichtentrennung/Businesslogik Auslagerung 15](#_Toc461027642)

[4. UserInterface Frontend 15](#_Toc461027643)

[4.1. Konzept 15](#_Toc461027644)

[4.2. Frameworks 15](#_Toc461027645)

[4.3. 16](#_Toc461027646)

[5. UserInterface Backend 16](#_Toc461027647)

[5.1. Unterscheidung Testsystem und Echtsystem 16](#_Toc461027648)

[5.2. Modelklassen 16](#_Toc461027649)

[5.3. Logik im Controller 16](#_Toc461027650)

[6. Businesslogik 16](#_Toc461027651)

[6.1. Benutzerverwaltung 16](#_Toc461027652)

[6.2. Reisenverwaltung 16](#_Toc461027653)

[6.3. Buchungsverwaltung 16](#_Toc461027654)

1. Einrichten der Netzwerkumgebung
   1. Workstations konfigurieren

Ausgangssituation besteht aus 3 Virtuelle Festplatten (VHD)

* IT IN 19
* IT IN 18 🡪 diese muss gelöscht werden und durch IT IN 20 ersetzt werden
* Backup VHD 🡪 diese soll auf Daten:\ kopiert und dort in Volume2 umbenannt werden

IT IN19 booten

Die entstandene einheitliche Ordnerstruktur soll folgendermaßen aufgebaut sein:

C:\

Daten:\

Sicherung:\

Volume1 – IT IN19

Volume2 – IT IN20

EasyBCD

Backup

….

In EasyBCD unter Add new Entry einen neuen Eintrag festlegen. Bei der Pfadangabe den folgenden Pfad auswählen (Daten:\Volume2) und folgenden Namen (IT IN20 WIN 8.1) vergeben.



Daten:\Volume2

IT IN20 WIN 8.1

In den Tools Edit Boot Menu den Eintrag IT IN18 löschen und

anschließend durch Save Settings bestätigen



Anschließend die VHD Volume2 löschen und die Backup VHD auf Daten kopieren

Vor dem Neustart die VHD Backup in Daten:\ in Volume2 umbenennen

Folgende TCP/IP Konfiguration vergeben

IP-Adressen der Clients: ab 192.168.120.101

Subnetmaske: 255.255.255.0

Standardgateway: 192.168.120.254

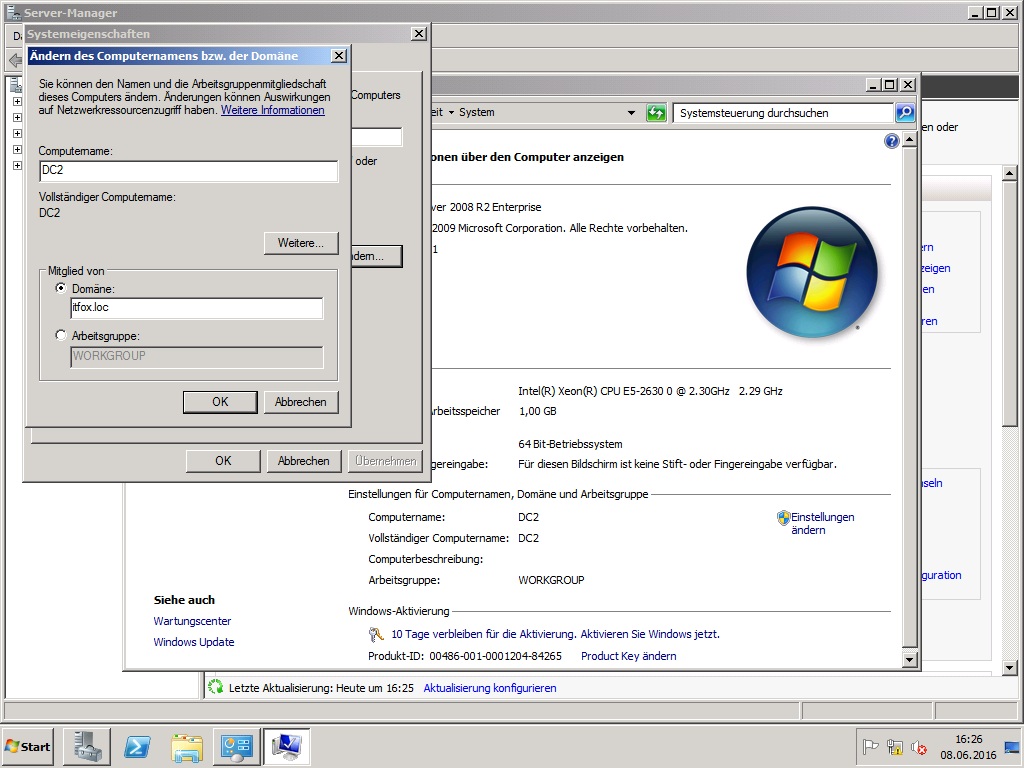
Anschließend Standard-Gateway anpingen, um die Erreichbarkeit zu testen

Windows Updates installieren

* 1. Domäne (Diagramm)
* Domäne: itfox.loc
* Server in der Domäne
  + Domänencontroller 1 (DC1): 192.168.120.1
  + Domänencontroller 1 (DC1): 192.168.120.2
  + Web: 192.168.120.3

?????????

* + SQL: 192.168.120.4
* Active Directory /DC1
  + Organisationseinheiten
    - Server
    - Clients
    - Benutzerkonten
      * Admin
      * Nicht-Admin
  1. Serverstruktur
* Server/Clients in Domain aufnehmen
  + Überprüfung, ob Server bereits aktiviert sind
    - Windows Update: Product Key ändern
      * Product Key: „7PJBC-63K3J-62TTK-XF46D-W3WMD“
  + Server in Domain aufnehmen
    - Unter System – Einstellungen ändern – Systemkonfiguration
    - In die Domäne wird „itfox.loc“ eingetragen, durch die Windows-Sicherheit muss man sich als Administrator anmelden
* Clients in Domäne hinzufügen
  + - Zuweisung eines DNS-Servers
    - Prüfen, ob die Domäne gefunden wird (mittels ping – ping itfox.loc)SCREENSHOT
    - Zur Domäne hinzufügen unter System – Einstellungen ändern
    - Bei Domäne wird „itfox.loc“ eingetragen



* Remotedesktop aktivieren
  + Unter Systemsteuerung – RemoteApp/Desktopverbindung einrichten
    - Unter Server-Manager: Remoteverwaltung für Server konfigurieren – die höchste Sicherheit wählen.
  + Remotedesktopverbindung
    - IP für Domäne eintragen : Bei DC1 = 192.168.120.1 – als Administrator anmelden
  + Remotedesktoptools unter Windows 8.1
    - Als Administrator von itfox.loc anmelden
      * Itfox.loc/Administrator
      * [Administrator@itfox.loc](mailto:Administrator@itfox.loc)
* Benutzerkonten anlegen
  + Erstellung von 2 Gruppen
    - AdminUser
    - NichtAdminUser
  + Struktur der Anmeldung:
    - NichtAdminUser: die ersten 3 Buchstaben vom Nachnamen und die ersten 3 Buchstaben vom Vornamen. (Bsp.: GroSte für Groinig Stefan)
    - AdminUser: die ersten 3 Buchstaben vom Nachnamen und die ersten 3 Buchstaben vom Vornamen und „-ad“ angehängt. (Bsp.: StigClau-ad für Stiglmayr Claudia)
    - SCREENSHOT der aktuellen Benutzer
  + Mitgliedergruppen:
    - Admins sind Mitglieder der Gruppe Domänen/Admins und Domänenbenutzer. Ein AdminUser kann lesen und ändern.
    - Ein NichtAdminUser kann lesen, aber nichts ändern
  + Struktur im AD
* SCREENSHOT der Organisationseinheiten
  + - Erstellung neuer Organisationseinheiten
      * Trennung Server/Client- zwecks anderer Richtlinien
      * Server : SQL, WEB
      * Domain Controller: DC1, DC2
      * Clients: alle aktuellen Clients
  1. Aufstufung des DomänenControllers



* + - * Installation der Rolle „Active-Directory-Domänendienste“
      * Neue Domäne in neuer Gesamtstruktur erstellen
      * Domäne: itfox.loc
      * Aktuelle Anmeldeinformationen: ITFOX\Administrator
* Als Standort wird „Default-First-Site-Name“ ausgewählt
* Der Domänencontroller DC1 wird auch als DNS-Server und als Globaler Katalog definiert.

* Die Speicherorte sind vorgegeben, hier wird nichts geändert
* DC2 hochstufen: Festlegen eines Computernamens und Mitgliedschaft einer Domäne
  1. DHCP Server aufsetzen

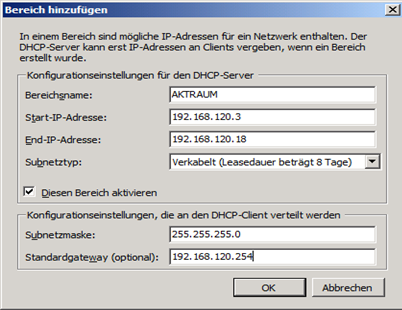
Dokumentation zu der Konfiguration unseres DHCP-Servers



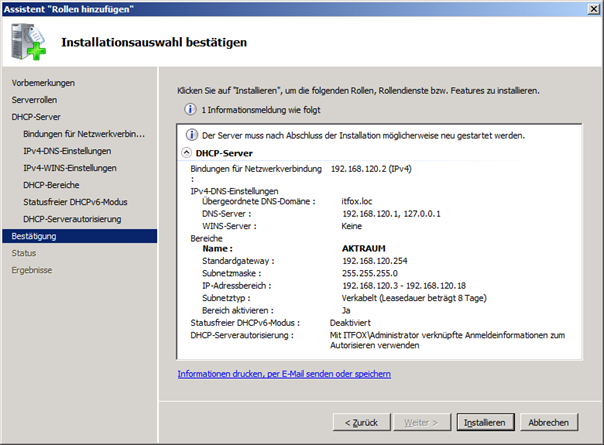
* Vergabe der Rolle „DHCP-Server“ unter dem Assistenten „Rollen hinzufügen“
* Als statische IP-Adresse wurde folgende Adresse definiert: 192.168.120.2 /DC1 (Domänencontroller).
* Als Netzwerkverbindung wird die IP-Adresse 192.168.120.2 vom Typ IPv4 genommen



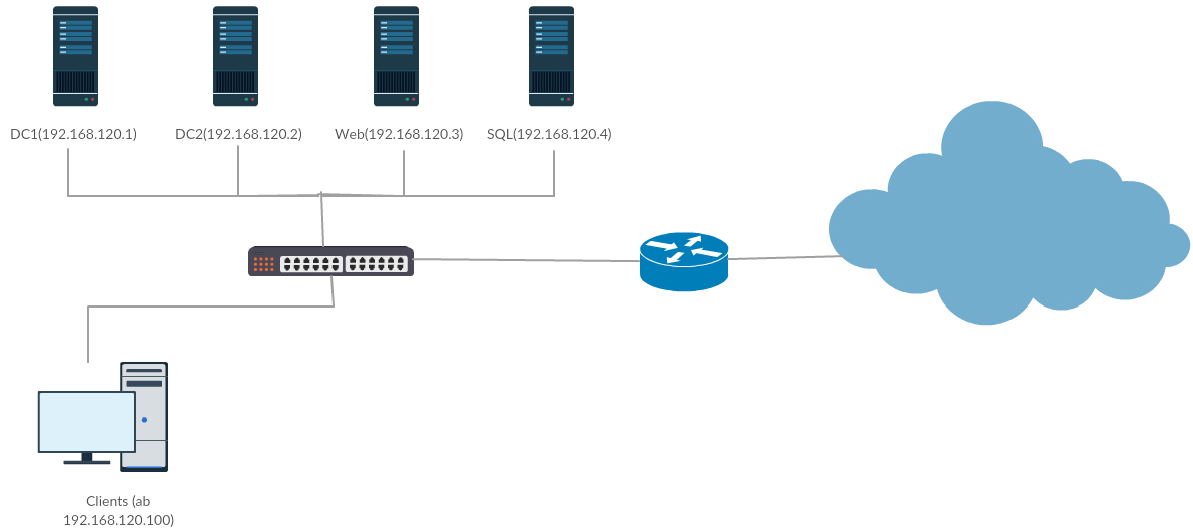
* Übergeordnete Domäne ist itfox.loc
* IPv4-Adresse des bevorzugten DNS-Servers: 192.168.120.1
* Keine Vergabe eines alternativen DNS-Servers
* Keine Vergabe von WINS-Server, da nicht benötigt
* Hinzufügen der DCHP-Bereiche

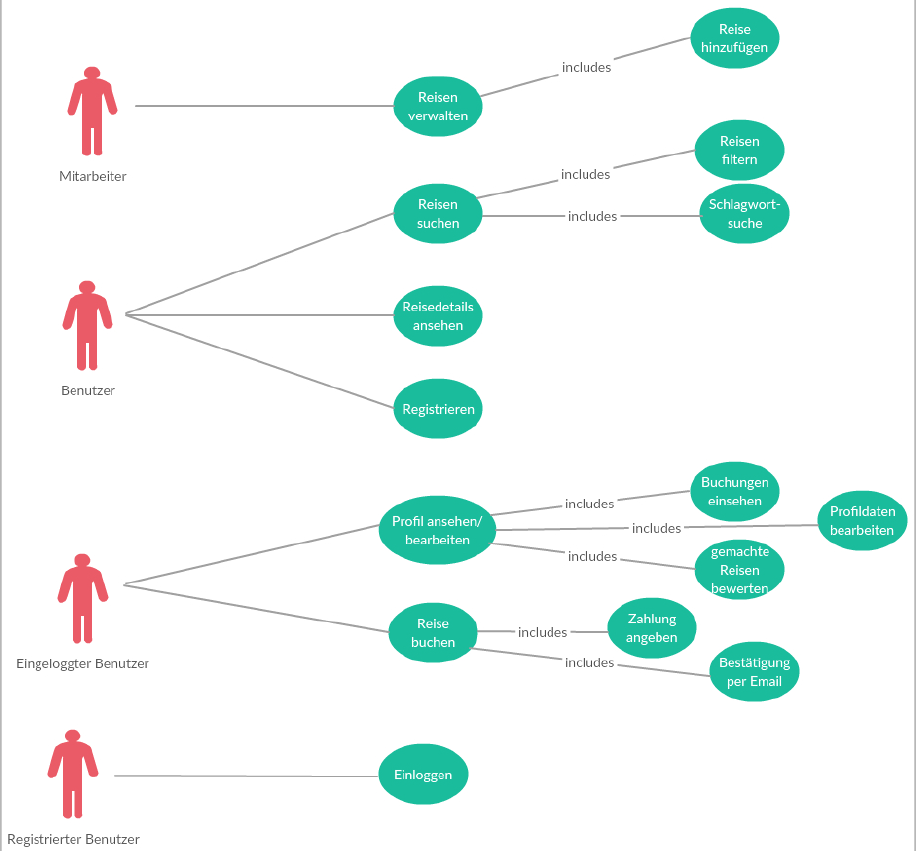
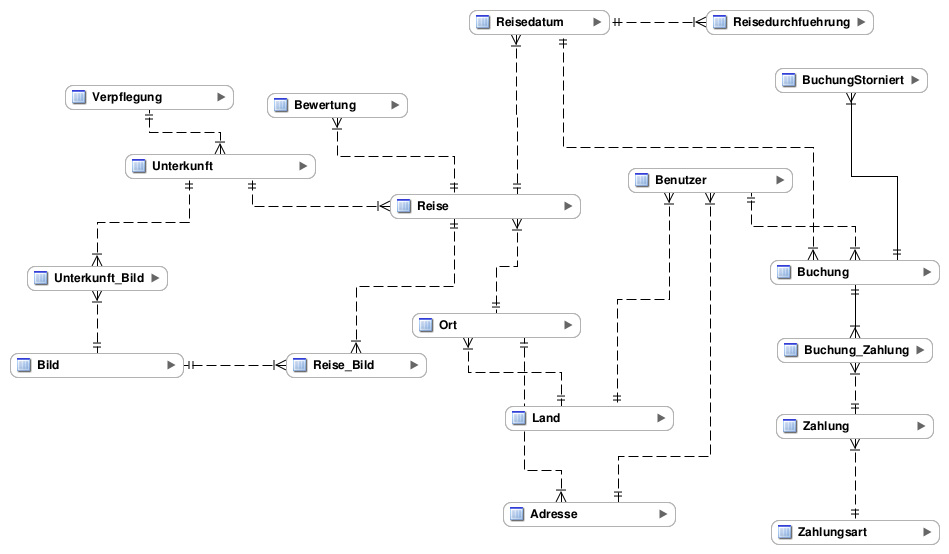


* Deaktivierung des statusfreien DHCP-v6-Modus
* Wir haben den DHCP-Server ins unsere AD mit eingebunden um ihn in unseren AD-Domänendiensten zu autorisieren. Bzw. vereinfacht uns dies die Anmeldung innerhalb unserer Domäne.
* Einbindung des DHCP-Servers in unser AD
* Autorisierung in unseren AD-Domänendiensten, hierzu werden die aktuellen Anmeldeinformationen verwendet
* Übersicht über alle zuvor eingestellten Punkte



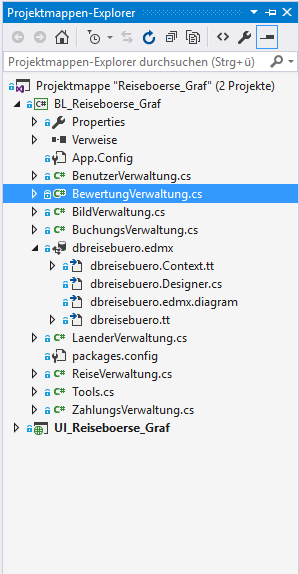
* 1. Netzwerkdiagramm(Grafik)



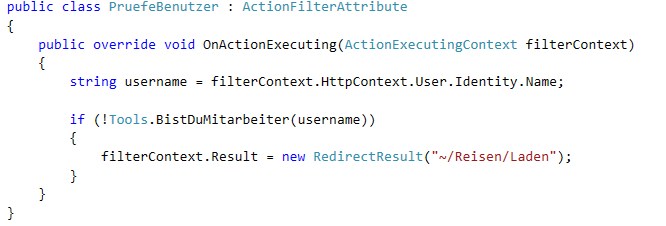
1. Anwendungsfälle (usecase Diagramm)
2. Datenbankentwurf
   1. Entity Relationship Model (ERM)
   2. Relationensynthese
3. Aufbau der Applikation
   1. Model View Controller

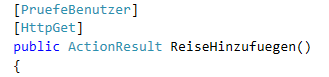
Die Webanwendung wurde mittels dem Entwurfsmuster Model – View – Controller realisiert.   
[Asp.net 🡪 1-2 Sätze]

* 1. Datenbankzugriff
     1. Entity Framework

Die Anbindung der Datenbank im Projekt sowie der Zugriff darauf erfolgten mittels Entity Framework. Aus Gründen der Wiederverwendbarkeit von Code sowie der besseren Strukturierung wurden diese Teile der Anwendung in eine eigene Klassenbibliothek (*BL\_Reiseboerse\_Graf*) ausgelagert.

* 1. Authentifizierung und Autorisierung

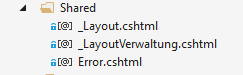
Die Unterscheidung zwischen Mitarbeiter und Kunden erfolgt in der Datenbank mittels Bitfeld. Bei der Anmeldung wird der Benutzer – unabhängig ob Kunde oder Mitarbeiter – anhand seiner Email und seines Passworts authentifiziert. Ob der aktuelle Benutzer Mitarbeiter ist wird mit einer Methode in der Klasse Tools der Businesslogik abgefragt. Die Autorisierung der Mitarbeiter erfolgt mittels eines benutzerdefinierten Action Filters.  
  
Die Anwendung bei den einzelnen Actionmethoden, die die Rechte eines Mitarbeiters erfordern erfolgt über das erstellte Attribut.



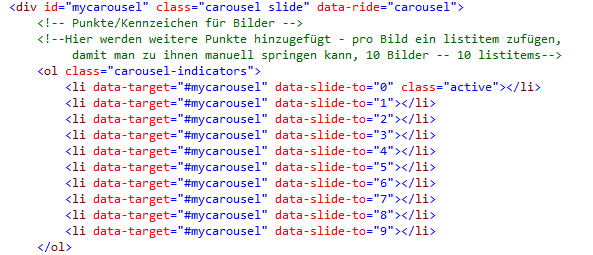
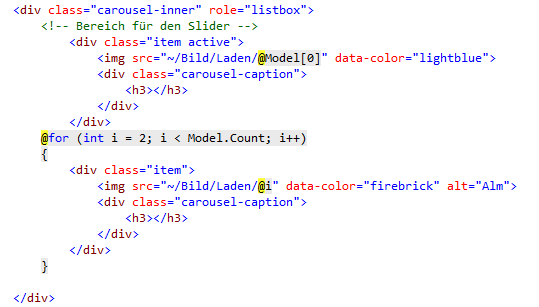
* 1. Grafische Gestaltung
     1. Bootstrap

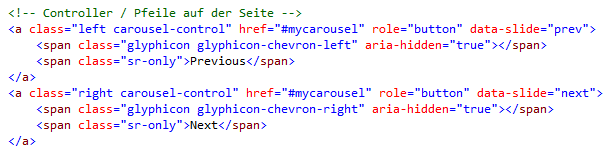
Bootstrap wurde verwendet, um in künftigen Versionen die Software einfacher auch für mobile Plattformen zugänglich zu machen.

* + 1. Geteilte Vorlagen

Sowohl für die Oberflächen für die Kunden, als auch für Verwaltung durch die Mitarbeiter, wurden geteilte Vorlagen erstellt, um ein einheitliches Aussehen zu gewährleisten.

* + 1. Bildslider auf der Startseite (Home/Index)

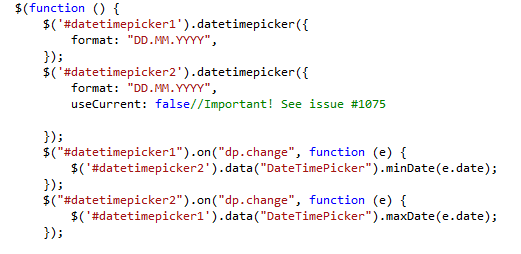
Quelle: <http://codepen.io/SitePoint/pen/ZbGwqe>   
  
Über die Klasse *carousel* wird der Slider dargestellt. Mit <li data-target=““> wird ein neuer Aufzählungspunkt für ein Bild hinzugefügt.  
  
Die Liste aller Bilder aus der Datenbank wird als Model an die View übergeben und dann in den Slider integriert.  
Des Weiteren wurden Pfeile zum Navigieren zwischen den Bildern hinzugefügt werden.



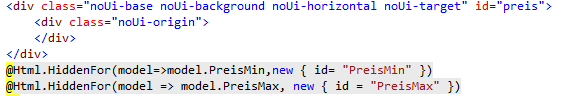
Mit *interval: 3400* wird festgelegt , dass das nächste Bild nach 3,4 Sekunden angezeigt werden soll.

*var $wHeight = $(window).height()* definiert, dass *$wHeight* die Höhe vom Fenster ausnutzen soll.  
*$item.height($wHeight)* definiert, dass *$wHeight* als Höhe jedes Elements verwendet werden soll.

* + 1. DateTimePicker

Quelle: <https://eonasdan.github.io/bootstrap-datetimepicker/>   
  


* + 1. Preisslider bei Reise filtern

Quelle: http://refreshless.com/nouislider/

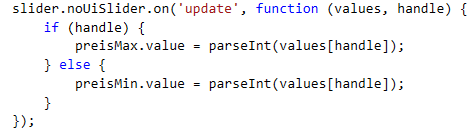
*noUiSlider* ruft die Methode *create* auf um einen Slider zu erstellen  
Folgende Parameter werden gesetzt:

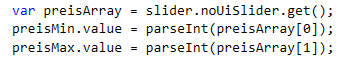
* *start*: Werte an denen die Handler starten
* *range*: Spanne an Werten
* *connect*: true (farbiger Balken als ausgewählte Spanne)
* *pips*: Skala anzeigen (mode und density: Durchsichtigkeit??)
* *step*: in 200er Stufen
* *margin*: zum Rand (man kann nicht weniger als 200 als Spanne auswählen)



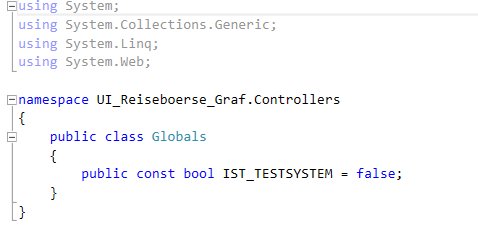
Die Html Elemente, auf denen die Werte gespeichert werden sollen, um diese an die Action zu schicken, werden den Variablen zugewiesen.



Beim Verändern der Handler im Slider wird der Wert automatisch geändert und den Variablen *preisMin* und *preisMax* zugewiesen.

Dann werden die aktuellen Werte ausgelesen, in *int* umgewandelt und auf dem *value* Attribut zugewiesen.

* 1. Unterscheidung zwischen Test- und Echtsystem

Um beim Programmieren nicht von der Datenbank abhängig zu sein, wurde im ersten Sprint eine Unterscheidung zwischen Testsystem (mit hartcodierten Werten) und Echtsystem (Datenbankzugriff) eingeführt. Diese wurde über die Klasse *Globals.cs* realisiert.

1. Usecases Kunden
   1. Reisen suchen und Reisen filtern
      1. Logischer Ablauf
      2. Modelklassen mit Validierung
      3. Controller
   2. Buchung tätigen
   3. Registrierung
   4. Einloggen
   5. Profil bearbeiten
   6. Reisen bewerten

1. Usecases Mitarbeiter
   1. Neue Reise anlegen
   2. Reise ändern
   3. Reise löschen
   4. Buchungen stornieren
   5. Stornierungen rückgängig machen
2. Businesslogik
   1. Benutzerverwaltung
   2. Reisenverwaltung
   3. Buchungsverwaltung
   4. EmailVerwaltung
   5. Tools