20. November 2021

Scherzer Tobias

HTL ZWETTL

Zwettl, Nö

Skript – Technologien Backend

Diplomarbeit

Inhalt

[Modelle in JSON POSTen 1](#_Toc91682685)

[Andere Variante zum POSTen 2](#_Toc91682686)

[Funktionaler POST 3](#_Toc91682687)

[Extras 3](#_Toc91682688)

[Nugets 3](#_Toc91682689)

[Model für Produkte 4](#_Toc91682690)

[ProductDependencies 5](#_Toc91682691)

[AProduct 5](#_Toc91682692)

[OptionGroup 5](#_Toc91682693)

[OptionSection 6](#_Toc91682694)

[Option 6](#_Toc91682695)

[Product 6](#_Toc91682696)

[INameable 6](#_Toc91682697)

[IIndexable 7](#_Toc91682698)

[IImageable 7](#_Toc91682699)

[IDescribable 7](#_Toc91682700)

[Beispiel – JSON (Vom Code oben) 7](#_Toc91682701)

[Emails senden 9](#_Toc91682702)

[Schritt 1 (Installation): 9](#_Toc91682703)

[Installation 9](#_Toc91682704)

[Schritt 2 (Aufbau): 10](#_Toc91682705)

[Emailaufbau + Senden 10](#_Toc91682706)

[Empfangene Email 10](#_Toc91682707)

[Schritt 3 (Aufbereitung): 11](#_Toc91682708)

[Projektdatei bearbeiten 11](#_Toc91682709)

[Email mit Aufbereitung 12](#_Toc91682710)

# Modelle in JSON POSTen

<https://andrewlock.net/model-binding-json-posts-in-asp-net-core/>

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# Andere Variante zum POSTen

<https://www.stevejgordon.co.uk/sending-and-receiving-json-using-httpclient-with-system-net-http-json>

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

# Funktionaler POST

Ein Bild, das Text, drinnen, Bildschirm, schließen enthält.

Automatisch generierte Beschreibung





## Extras

### Nugets

Newtonsoft.JSON installieren in Nugets

<https://www.educative.io/edpresso/how-to-create-a-json-string-in-c-sharp>

# Model für Produkte

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

public string Product()

{

//User user = new User() { Age = 18, Email = "jokers@mail.com", Name = "Tobias Scherzer" };

List<Option> options = new List<Option> { new Option("D150", "Fetter Diesel Motor", new List<string> { "youtube.com" }, "D150") };

List<string> productImages = new List<string> { "boahnhub.tk" };

List<OptionGroup> optionGroups = new List<OptionGroup> { new OptionGroup("Color", "The exterior color of the product", "COLOR\_GROUP", new List<string> { "RED", "WHITE", "GREEN" }), new OptionGroup("Motor Type", "The motor of your car", "MOTORTYPE\_GROUP", new List<string> { "DIESEL", "PETROL", "ELECTRIC" }), new OptionGroup("Motor", "The selected Motor power", "MOTOR\_GROUP", new List<string> { "D150", "D200", "D250" }) };

List<OptionSection> optionSections = new List<OptionSection> { new OptionSection("Exterior", "EXTERIOR", new List<string> { "COLOR\_GROUP" }), new OptionSection("Motor", "MOTOR\_SECTION", new List<string> { "MOTORTYPE\_GROUP", "MOTOR\_GROUP" }) };

ProductDependencies productDependencies = new ProductDependencies(50000, new List<string> { "RED", "DIESEL", "D150" }, new List<OptionGroup> { new OptionGroup("Color", "The exterior color of the car", "COLOR\_GROUP", new List<string> { "BLUE", "RED", "WHITE" }) }, new Dictionary<string, List<string>> { { "D150", new List<string> { "DIESEL" } } }, new Dictionary<string, List<string>> { { "D150", new List<string> { "PETROL" } } }, new Dictionary<string, int> { { "D150", 1500 } });

Product product = new Product("0", "Alfa Romeo 159", "A really nice car", productImages, productDependencies, options, optionGroups, optionSections);

string stringjson = JsonConvert.SerializeObject(product);

return stringjson;

}

## ProductDependencies

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## AProduct

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## OptionGroup

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## OptionSection

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Option

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Product

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## INameable

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## IIndexable

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## IImageable

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## IDescribable

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Beispiel – JSON (Vom Code oben)

{

    "Dependencies": {

        "BasePrice": 50000,

        "DefaultOptions": [

            "RED",

            "DIESEL",

            "D150"

        ],

        "ReplacementGroups": [

            {

                "Id": "COLOR\_GROUP",

                "Name": "Color",

                "Description": "The exterior color of the car",

                "OptionIds": [

                    "BLUE",

                    "RED",

                    "WHITE"

                ]

            }

        ],

        "Requirements": {

            "D150": [

                "DIESEL"

            ]

        },

        "Incompabilities": {

            "D150": [

                "PETROL"

            ]

        },

        "PriceList": {

            "D150": 1500

        }

    },

    "OptionGroups": [

        {

            "Id": "COLOR\_GROUP",

            "Name": "Color",

            "Description": "The exterior color of the product",

            "OptionIds": [

                "RED",

                "WHITE",

                "GREEN"

            ]

        },

        {

            "Id": "MOTORTYPE\_GROUP",

            "Name": "Motor Type",

            "Description": "The motor of your car",

            "OptionIds": [

                "DIESEL",

                "PETROL",

                "ELECTRIC"

            ]

        },

        {

            "Id": "MOTOR\_GROUP",

            "Name": "Motor",

            "Description": "The selected Motor power",

            "OptionIds": [

                "D150",

                "D200",

                "D250"

            ]

        }

    ],

    "OptionSections": [

        {

            "Name": "Exterior",

            "Id": "EXTERIOR",

            "OptionGroupIds": [

                "COLOR\_GROUP"

            ]

        },

        {

            "Name": "Motor",

            "Id": "MOTOR\_SECTION",

            "OptionGroupIds": [

                "MOTORTYPE\_GROUP",

                "MOTOR\_GROUP"

            ]

        }

    ],

    "Name": "Alfa Romeo 159",

    "Description": "A really nice car",

    "Images": [

        "boahnhub.tk"

    ],

    "Id": "0",

    "Options": [

        {

            "Name": "D150",

            "Description": "Fetter Diesel Motor",

            "Images": [

                "youtube.com"

            ],

            "Id": "D150",

            "Options": **null**

        }

    ]

}

# Emails senden

(Quelle P0015K)

## Schritt 1 (Installation):

### Installation

Ein Bild, das Text, Screenshot, Bildschirm, schwarz enthält.

Automatisch generierte Beschreibung

## Schritt 2 (Aufbau):

### Emailaufbau + Senden

Ein Bild, das Text, Screenshot, Bildschirm enthält.

Automatisch generierte Beschreibung

### Empfangene Email

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Schritt 3 (Aufbereitung):

### Projektdatei bearbeiten

Ein Bild, das Text, Monitor, Screenshot, Bildschirm enthält.

Automatisch generierte Beschreibung

(PreserveCompilationContext)

### Email mit Aufbereitung

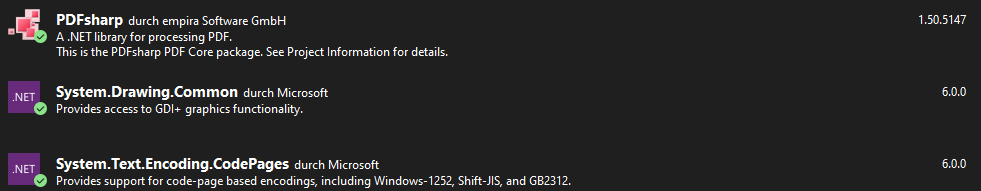
Ein Bild, das Text enthält.

Automatisch generierte BeschreibungEin Bild, das Text enthält.

Automatisch generierte Beschreibung

# PDF generieren

## Schritt 1 (Installieren):



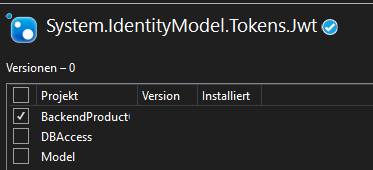
## Schritt 2 (Ausgabe):

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

# JSON Web Token (JWT)

## Schritt 1 (Installieren):



## Schritt 2 (Implementieren):

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

P0019K

## Schritt 3 (Ausführen):

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Sollte im GET von den gespeicherten Produkten passieren