VDAB

samen sterk voor werk

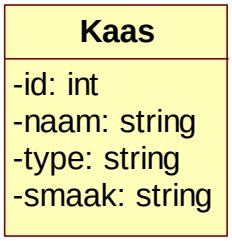
C#2013 Web API Takenbundel

Inhoudsopgave

1 TAKEN

1.1 Kaas service lees operaties

Je maakt een service over kazen:



Je houdt volgende data bij in het interne geheugen, niet in een echte database.

id	naam	type	smaak
1	Oud Brugge	Hard	Pittig
2	Watou	Halfhard	Zacht
3	WynenDaele	Zacht	Pittig
4	La Prihel	Vers	Pittig
5	GrevenBroucker	Blauwgeaderd	Romig

Een REST client kan:

- een lijst vragen met alle kazen
  Deze lijst bevat per kaas id, naam en een hyperlink naar de detailinformatie van die kaas
- de kazen van één smaak vragen
  Deze lijst bevat per kaas id, naam en een hyperlink naar de detailinformatie van die kaas
- één kaas vragen via het nummer

Je gebruikt Fiddler om de service te testen.

1.2 Kaas service andere operaties

Je breidt de service over kazen uit:

- De client kan een kaas verwijderen
- De client kan een kaas toevoegen. De properties Naam, Type en Smaak zijn verplicht in te vullen en bevatten een aantal tekens tussen 1 en 50.
- De client kan een kaas wijzigen.
- Je kan met een browser documentatie opvragen van de service.

1.3 KAZEN client

Je maakt een client van de Kazen service.

De gebruiker tikt in de client de gegevens van een nieuwe kaas.

De client roept de service op om deze kaas toe te voegen.

De client toont onmiddellijk daarna de lijst van alle kazen.
De client toont daarbij per kaas de id en de naam.

2 VOORBEELDOPLOSSINGEN

2.1 Kaas service lees operaties

2.1.1 De class Kaas

```
using System.Runtime.Serialization;
namespace KazenService.Models
{
  [DataContract(Name="kaas", Namespace="")]
  public class Kaas
  {
    [DataMember(Name="id", Order=1)]
    public int ID { get; set; }
    [DataMember(Name = "naam", Order = 1)]
    public string Naam { get; set; }
    [DataMember(Name = "type", Order = 1)]
    public string Type { get; set; }
    [DataMember(Name = "smaak", Order = 1)]
    public string Smaak { get; set; }
  }
}
```

2.1.2 De class KaasBeknopt

```
using System.Runtime.Serialization;
namespace BrouwersService.Models
{
    [DataContract(Name = "kaas", Namespace = "")]
    public class KaasBeknopt
    {
        [DataMember(Name = "id", Order = 1)]
        public int ID { get; set; }
        [DataMember(Name = "naam", Order = 2)]
        public string Naam { get; set; }
        [DataMember(Name = "detail", Order = 3)]
        public string Detail { get; set; }
    }
}
```

2.1.3 De class Kazen

```
using System.Collections.Generic;
using System.Runtime.Serialization;
namespace BrouwersService.Models
{
    [CollectionDataContract(Name = "kazen", Namespace = "")]
    public class Kazen : List<KaasBeknopt>
    {
    }
}
```

2.1.4 De class InMemoryDatabase

```
using System.Collections.Generic;
namespace KazenService.Models
{
  public class InMemoryDataBase
  {
    private static Dictionary<int, Kaas> kazenValue;
    static InMemoryDataBase()
    {
      var oudBrugge =
        new Kaas {ID = 1, Naam = "Oud Brugge", Type = "hard", Smaak = "pittig"};
      var watou =
        new Kaas {ID = 2, Naam = "Watou", Type = "halfhard", Smaak = "zacht"};
      var wynenDaele =
        new Kaas {ID = 3, Naam = "WynenDaele", Type = "zacht", Smaak = "pittig"};
```

```
      var laPrihel =
        new Kaas {ID = 4, Naam = "La Prihel", Type = "vers",Smaak = "pittig"};
      var grevenBroucker =
        new Kaas {ID = 5, Naam = "GrevenBroucker", Type = "blauwgeaderd",
          Smaak = "romig"};
      kazenValue = new Dictionary<int, Kaas> {
        {oudBrugge.ID, oudBrugge}, {watou.ID, watou}, {wynenDaele.ID, wynenDaele},
        {laPrihel.ID, laPrihel}, {grevenBroucker.ID, grevenBroucker}};
    }
    public static Dictionary<int, Kaas> Kazen
    {
      get { return kazenValue; }
    }
  }
}
```

2.1.5 De class KazenController

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using KazenService.Models;
namespace KazenService.Controllers
{
  public class KazenController : ApiController
  {
     public IHttpActionResult GetAll()
    {
      var detail = this.Request.RequestUri.AbsoluteUri + "/";
      var kazen = new Kazen();
      kazen.AddRange(from kaas in InMemoryDataBase.Kazen.Values
                     orderby kaas.Naam
                     select new KaasBeknopt
                     {
                         ID = kaas.ID,
                         Naam = kaas.Naam,
                         Detail = detail + kaas.ID
                     });
      return this.Ok(kazen);
    }
    public IHttpActionResult GetBySmaak(String smaak)
    {
      smaak = smaak.ToUpper();
      var detail = this.Request.RequestUri.AbsoluteUri;
      detail = detail.Substring(0, detail.IndexOf('?'));
      detail += "/";
      var kazen = new Kazen();
      kazen.AddRange(from kaas in InMemoryDataBase.Kazen.Values
                     where kaas.Smaak.ToUpper() == smaak
                     orderby kaas.Naam
                     select new KaasBeknopt
                     {
                         ID = kaas.ID,
                         Naam = kaas.Naam,
                         Detail = detail + kaas.ID
                     });
      return Ok(kazen);
    }
```

```
    public IHttpActionResult Get(int id)
    {
      if (InMemoryDataBase.Kazen.ContainsKey(id))
      {
        return this.Ok(InMemoryDataBase.Kazen[id]);
      }
      return this.NotFound();
    }
  }
}
```

2.2 Kaas service andere operaties

2.2.1 De method Delete

```
public IHttpActionResult Delete(int id)
{
  if (InMemoryDataBase.Kazen.ContainsKey(id))
    {
      InMemoryDataBase.Kazen.Remove(id);
      return this.Ok();
  }
  return this.NotFound();
}
```

2.2.2 De class Kaas

```
using System.ComponentModel.DataAnnotations;
using System.Runtime.Serialization;
namespace KazenService.Models
{
  [DataContract(Name = "kaas", Namespace = "")]
  public class Kaas
  {
    [DataMember(Name = "id", Order = 1)]
    public int ID { get; set; }
    [DataMember(Name = "naam", Order = 2)]
    [Required]
    [StringLength(50, MinimumLength = 1)]
    public string Naam { get; set; }
    [DataMember(Name = "type", Order = 3)]
    [Required]
    [StringLength(50, MinimumLength = 1)]
    public string Type { get; set; }
    [Required]
    [StringLength(50, MinimumLength = 1)]
    [DataMember(Name = "smaak", Order = 4)]
    public string Smaak { get; set; }
  }
}
```

2.2.3 De method Post

```
public IHttpActionResult Post(Kaas kaas)
{
  if (! this.ModelState.IsValid)
  {
    return this.BadRequest();
  }
  var id = InMemoryDataBase.Kazen.Keys.Max() + 1;
  kaas.ID = id;
  InMemoryDataBase.Kazen[id] = kaas;
  return this.Created(this.Request.RequestUri.AbsoluteUri + "/" + id,  kaas);
}
```

2.2.4 De method Put

```
public IHttpActionResult Put(int id, Kaas kaas)
{
  if (!this.ModelState.IsValid)
  {
    return this.BadRequest(this.ModelState);
  }
  if (InMemoryDataBase.Kazen.ContainsKey(id))
  {
    InMemoryDataBase.Kazen[id] = kaas;
    return this.Ok();
  }
  return this.NotFound();
}
```

2.3 KAZEN client

2.3.1 De class Kaas

```
using System.Runtime.Serialization;
namespace KazenClient
{
  [DataContract(Name = "kaas", Namespace = "")]
  public class Kaas
  {
    [DataMember(Name = "id", Order = 1)]
    public int ID { get; set; }
    [DataMember(Name = "naam", Order = 1)]
    public string Naam { get; set; }
    [DataMember(Name = "type", Order = 1)]
    public string Type { get; set; }
    [DataMember(Name = "smaak", Order = 1)]
    public string Smaak { get; set; }
  }
}
```

2.3.2 De class KaasBeknopt

```
using System.Runtime.Serialization;
namespace KazenClient
{
  [DataContract(Name = "kaas", Namespace = "")]
  public class KaasBeknopt
  {
    [DataMember(Name = "id", Order = 1)]
    public int ID { get; set; }
    [DataMember(Name = "naam", Order = 2)]
    public string Naam { get; set; }
    [DataMember(Name = "detail", Order = 3)]
    public string Detail { get; set; }
  }
}
```

2.3.3 De class Kazen

```
using System.Collections.Generic;
using System.Runtime.Serialization;
namespace KazenClient
{
  [CollectionDataContract(Name="kazen", Namespace= "")]
  public class Kazen:List<KaasBeknopt>
  {
  }
}
```

2.3.4 Program.cs

```
using System;
using System.Net;
using System.Net.Http;
namespace KazenClient
{
  class Program
  {
    static void Main(string[] args)
      {
        var kaas = new Kaas();
        Console.Write("Naam:");
        kaas.Naam = Console.ReadLine();
        Console.Write("Type:");
        kaas.Type = Console.ReadLine();
        Console.Write("Smaak:");
        kaas.Smaak = Console.ReadLine();
        var client = new HttpClient();
        var response = client.PostAsJsonAsync<Kaas>(
          "http://localhost:53810/api/kazen/", kaas).Result;
        if (response.IsSuccessStatusCode)
        {
          response = client.GetAsync("http://localhost:53810/api/kazen").Result;
          if (response.IsSuccessStatusCode)
          {
            var kazen = response.Content.ReadAsAsync<Kazen>().Result;
            kazen.ForEach(
              beknopt => Console.WriteLine(beknopt.ID + ":" + beknopt.Naam));
          }
        }
        else
        {
          Console.WriteLine("Probleem bij toevoegen kaas:" + response.StatusCode);
        }
      }
    }
  }
```

3 COLOFON

Domeinexpertisemanager:	Rita Van Damme
Moduleverantwoordelijke:	Hans Desmet
Medewerkers:	Hans Desmet
Versie:	8/10/2014
Nummer dotatielijst:	