gRPC

Module 3 - Containerizing gRPC Service

Student Lab Manual

Instructor Edition (Book Title Hidden Style)

**Conditions and Terms of Use**

**Microsoft Confidential**

This training package is proprietary and confidential, and is intended only for uses described in the training materials. Content and software is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or software included in such packages is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URLs and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2021 Microsoft Corporation. All rights reserved.

**Copyright and Trademarks**

© 2021 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

For more information, see Use of Microsoft Copyrighted Content at  
<http://www.microsoft.com/en-us/legal/intellectualproperty/Permissions/default.aspx>

Internet Explorer, Microsoft, SQL Server, Visual Studio, and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Contents

[Lab 3: Containerizing gRPC Service 1](#_Toc62071609)

[Exercise 1: Create a Containerized gRPC Service 3](#_Toc62071610)

[Exercise 2: Install gRPCurl 5](#_Toc62071611)

[Exercise 3: Install gRPCui 8](#_Toc62071612)

# Lab 3: Containerizing gRPC Service

#### Introduction

The aim of this exercise is to explore implementing a containerized gRPC application. In addition, two tools for testing gRPC endpoints, mainly gRPCurl and gRPCui, will be introduced.

#### Objectives

After completing this lab, you will be able to:

* Create a containerized gRPC service
* Install and use gRPCurl to test your gRPC endpoints
* Install and use gRPCui to test your gRPC endpoints

#### Prerequisites

* Visual Studio 2019 or higher
* .Net 5 or higher

#### Estimated Time to Complete This Lab

60 minutes

Exercise 1: Create a Containerized gRPC Service

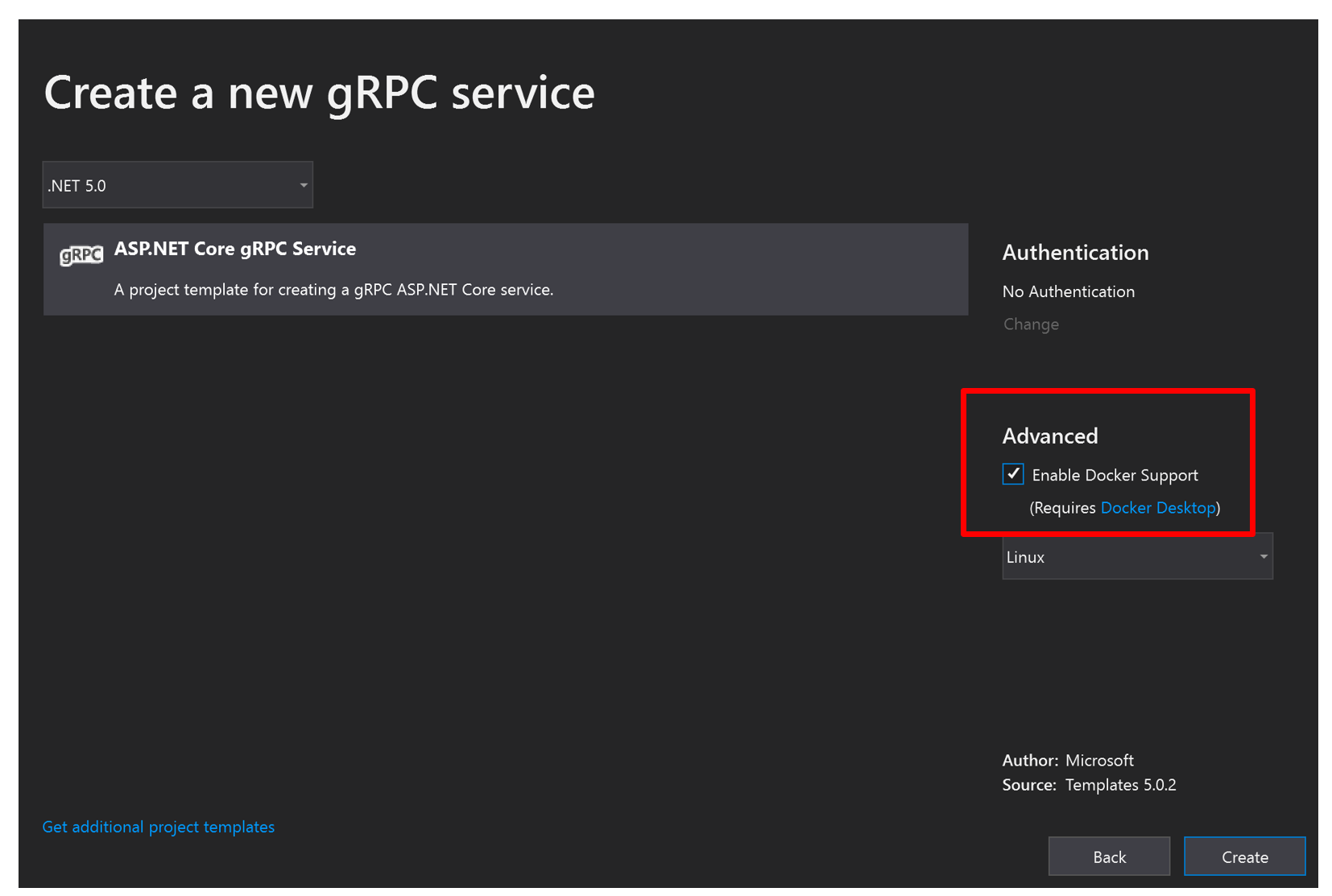
#### Objectives

In this exercise, you will:

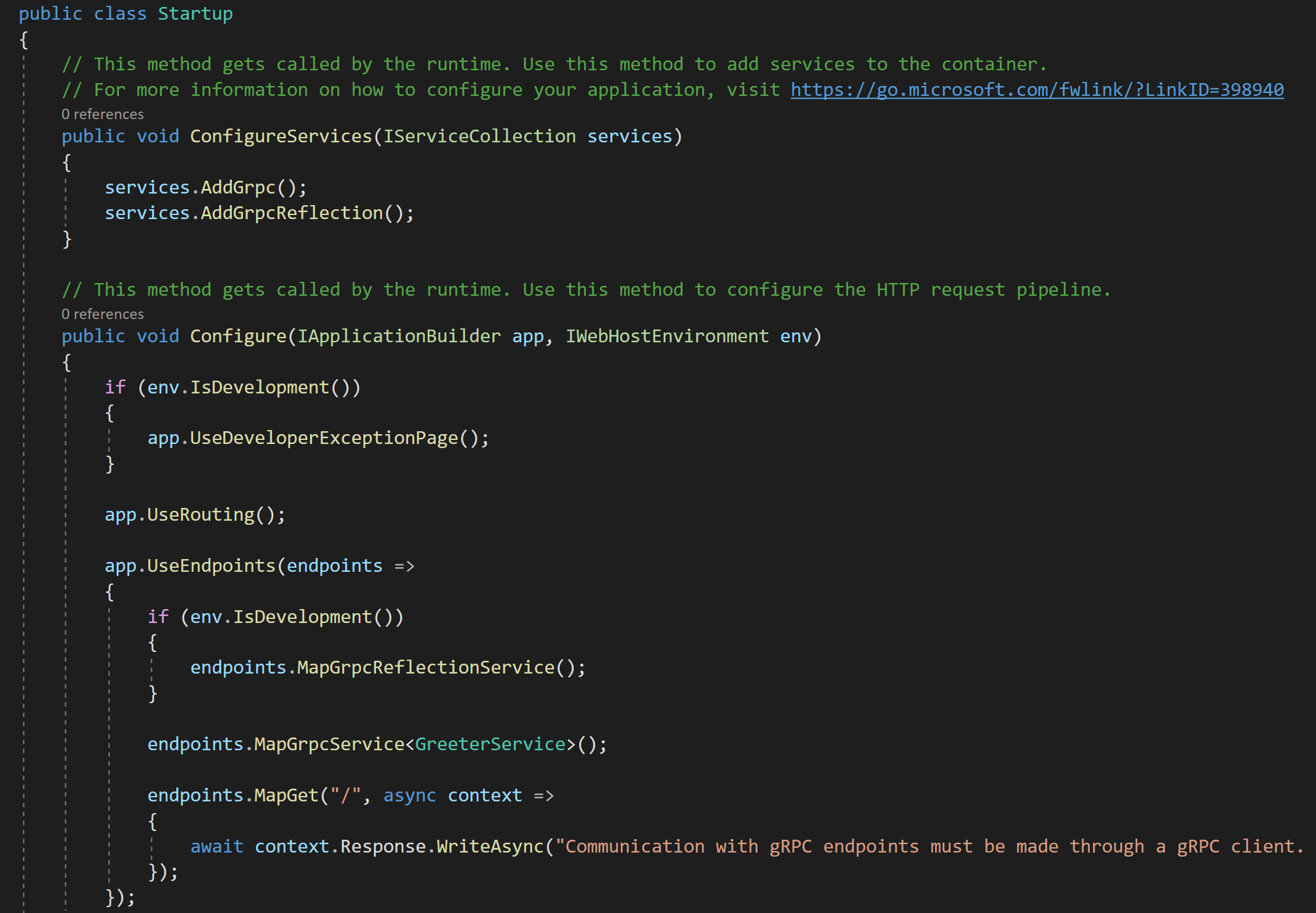
* Create a new ASP.NET Core gRPC Project

Task 1: Setup a Containerized ASP.NET Core gRPC

1. Create a new gRPC project and make sure you select the “Enable Docker Support” checkbox.



1. To make gRPCurl working, the latter needs to know gRPC message and endpoints definition, there are two possibilities: use Proto files or use Reflection. We will use Reflection because it’s the easiest way to proceed with ASP. NET Core. To be able to use Reflection, you have to install **Grpc.AspNetCore.Server.Reflection** nuget package.
2. Then configure you ASP.NET Core gRPC application with services.AddGrpcReflection() and endpoints.MapGrpcReflectionService() methods. The first one enables the gRPC reflection and the second one is the endpoint that provides to gRPCurl for example endpoints and messages information by reflection:



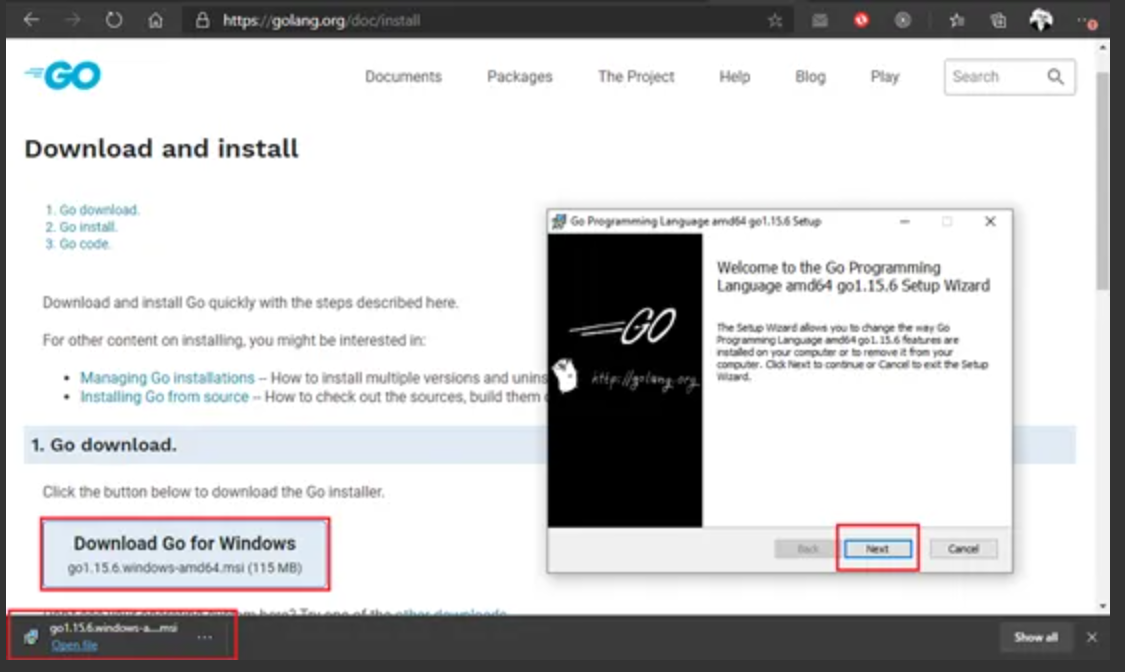
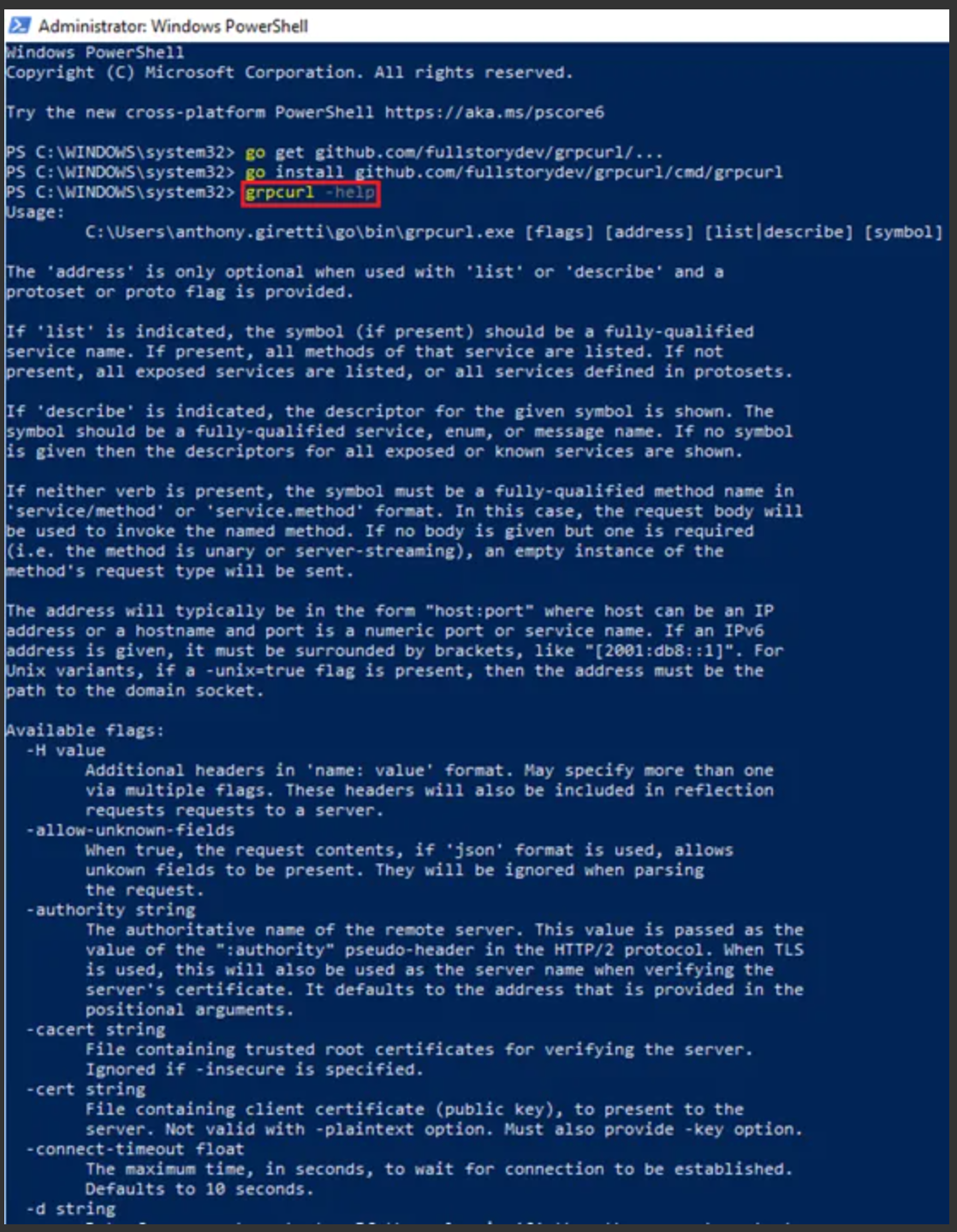
Exercise 2: Install gRPCurl

#### Objectives

In this exercise, you will:

* Install gRPCurl
* Test the containerized gRPC endpoint using gRPCurl

Task 1: Setup gRPCurl

1. Do you know Curl? The tool that allows you to transfer data through HTTP (and others such as IMAP, FTP etc.) to the order line? Well know that a tool based on the same principle for gRPC exists: gRPCurl. In this excerice you will install and use gRPCurl to test your gRPC endpoints with ASP. NET Core.
2. gRPCurl is coded with the language GO, to roll it you will have to install GO on your machine, you can download it [here](https://golang.org/doc/install). Its installation is really simple you just need to run the installer and you’re done!  
   
3. Once done you can run these two commands in a PowerShell window, I strongly suggest to run it as Administrator to avoid any installation issue:  
   **go get github.com/fullstorydev/grpcurl/...**  
   **go install github.com/fullstorydev/grpcurl/cmd/grpcurl**
4. If the installation has been successful you should be able to test by running the following command that shows gRPCurl commands:  
   **grpcurl --help**
5. You should see the following in your PowerShell window:  
   

Task 2: Test the containerized gRPC endpoint using gRPCurl

1. gRPCurl allows you to invoke your endpoints, but you can also perform different operations. Before running any of the operations below make sure the gRPC service project is running and note the port number as you will need it in the commands below.  
     
   List all available gRPC services by running the following command:  
   **grpcurl localhost:<port> list**

List all available gRPC endpoints for a given gRPC service by running the following command:  
**grpcurl localhost:<port> list greet.Greeter**

Describe an endpoint (get all details from the proto file) by running the following command:  
**grpcurl localhost:<port> describe greet.Greeter.SayHello**

Describe a gRPC service (get all details from the proto file) by running the following command:  
**grpcurl localhost:<port> describe greet.Greeter**

Describe a message (get all details from the proto file) by running the following command:  
**grpcurl localhost::<port> describe greet.HelloRequest**

Invoke an endpoint without input message by running the following command:  
**grpcurl localhost:<port> greet.Greeter.SayHello**

Invoke an endpoint with an input message (using its JSON representation) by running the following command:  
**grpcurl -d '{ \"name\": \"World\" }'localhost:<port> greet.Greeter.SayHello**

Exercise 3: Install gRPCui

#### Objectives

In this exercise, you will:

* Install gRPCui
* Test the containerized gRPC endpoint using gRPCui

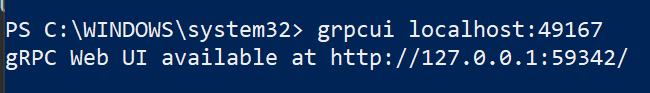
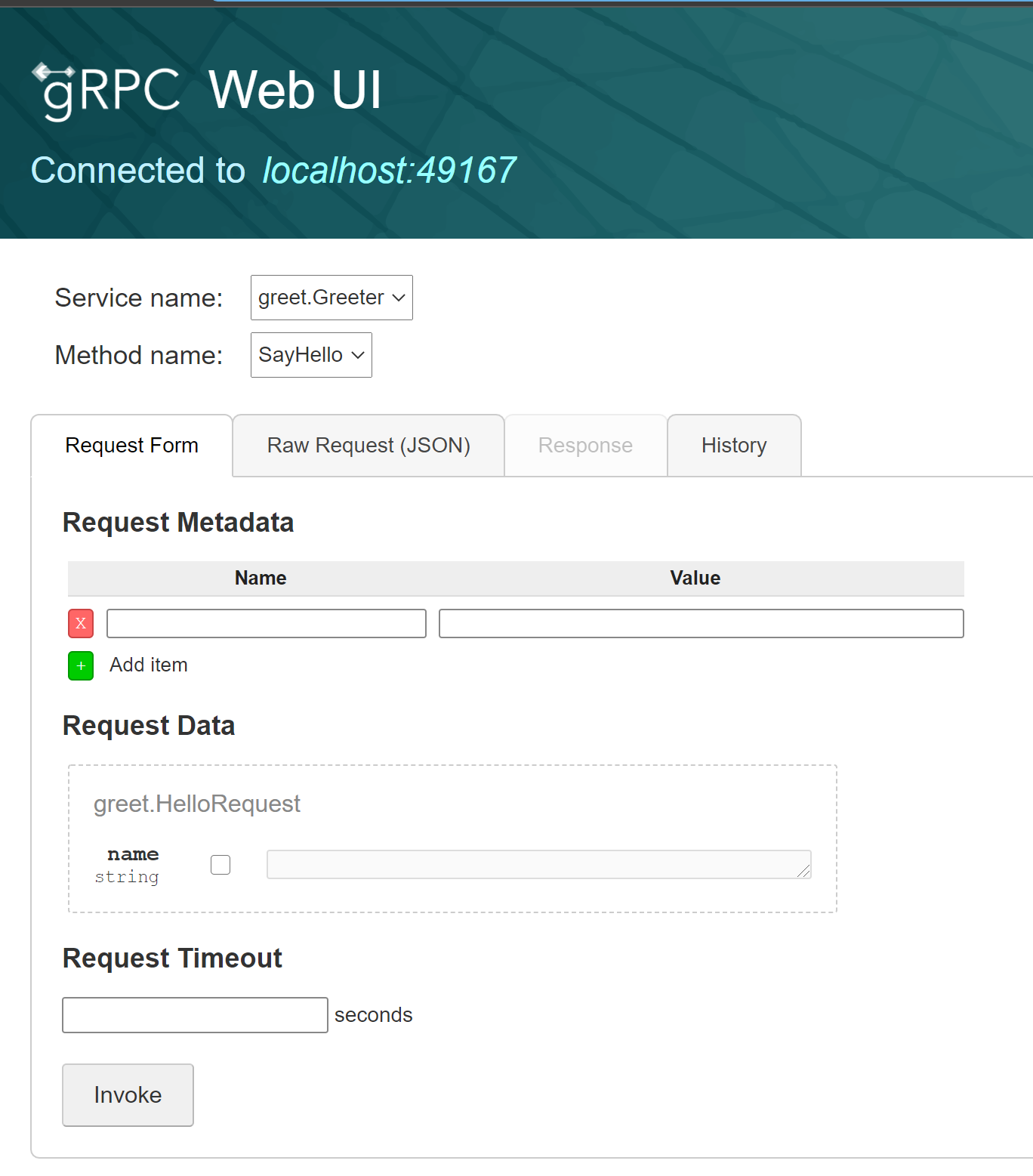
Task 1: Setup gRPCui

1. In a previous exercise, you got introduced to gRPCurl, a command line tool that allows you test you gRPC endpoints. In this exercise you will utilize gRPCui which is quite the same tool, but available as a web tool, built on top on gRPCurl.Core.
2. Download and install gRPCui with the following commands (I suggest you to install it via a PowerShell windows as Administrator):

go get github.com/fullstorydev/grpcui/...  
go install github.com/fullstorydev/grpcui/cmd/grpcui

Task 2: Test the containerized gRPC endpoint using gRPCui

1. Then run the following command via a PowerShell command for example:  
   **grpcui {yourgrpcapplicationurl}:{yourgrpcapplicationport}**

You need here to provide the URl and the port of you gRPC service because gRPCui cannot guess it obviously. If the installation succeed you should have something like this in return of your command:  
  
gRPCui will expose a local URI that is the URI of the GUI you use to test you gRPC services as shown in the following:  


1. The UI is very intuitive, you can easily find the service to test and its method in the dropdown list at the top. Create your request is also intuitive, you can easily fill:

* Request Metadata (Trailers)
* Request Data (Fill streamed data or not is the same, like a collection to fill)
* Request Timeout (Deadline)

