**App Configuration with a Spring Config Server**

This tutorial takes you through setting up a .NET Core application that gets configuration values from a Spring Config Server.

First, **create a Github repository to hold config values**.

1. Navigate to [Github](https://github.com/) and either login or create a new account
2. Create and initialize a new repository, named Spring-Config-Demo
3. Once created note the url of the new repo

Next, **add a config file to the repository**.

1. Create a new file in the repo named my-values.yml
2. Add the following to the file

Value1: some-val

Value2: another-val

1. Commit the new file to the repo

Then, **start a config server instance**. Depending on your hosting platform this is done in several ways.

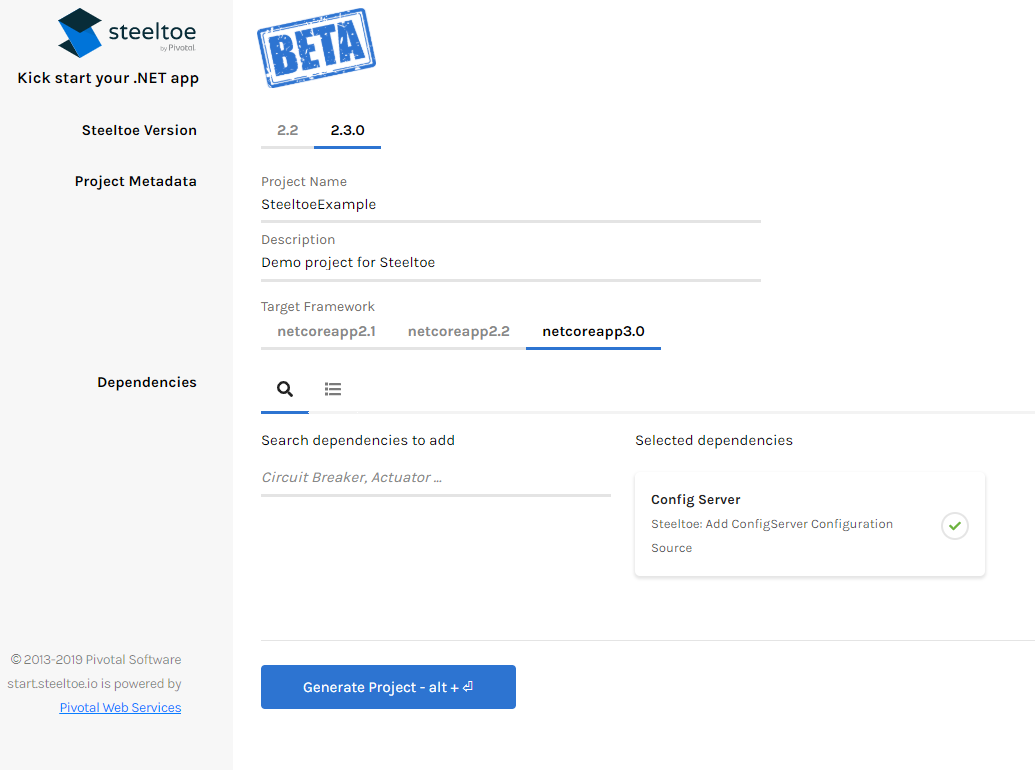
* **Local**
* Cloud Foundry

1. Using the [Steeltoe dockerfile](https://github.com/steeltoeoss/dockerfiles), start a local instance of Spring Config Server. The dockerfile will choose the correct image based on your OS.

docker run --publish 8888:8888 steeltoeoss/config-server --spring.cloud.config.server.git.uri=https://github.com/eswaribala/spring-config-demo.git

Next, **create a .NET Core WebAPI** that retrieves values from the Spring Config instance

1. Create a new ASP.NET Core WebAPI app with the [Steeltoe Initializr](https://start.steeltoe.io/)



* + **SteeltoeVersion:** 2.4 for the latest stable
  + Project Metadata:

**Name:** Spring\_Config\_Example

**Target Framework:** netcoreapp3.1 is the latest stable

* + **Dependencies:** Config Server
  + Click **Generate Project** to download a zip containing the new project

1. Extract the zipped project and open in your IDE of choice (we use Visual Studio)
2. Set the instance address and name in **appsettings.json**
   * **Local**
   * Cloud Foundry

Update with the Spring Config Server info

**json**

**json**

**json**

**json**

**json**

**json**

**json**

**json**

{

...

"spring": {

"application": {

"name": "my-values"

},

"cloud": {

"config": {

"uri": "http://localhost:8888",

"validateCertificates": false

}

}

},

...

}

Note

For the application to find its values in the git repo, the spring:application:name and the yaml file name **must** match. In this example my-values matched.

1. Run the application
   * **Local**
   * Cloud Foundry
   * Using the .NET cli

**powershell**

**powershell**

**powershell**

**powershell**

**powershell**

**powershell**

**powershell**

**powershell**

dotnet run <PATH\_TO>\Spring\_Config\_Example.csproj

Navigate to the endpoint (you may need to change the port number) <http://localhost:5000/api/values>

* + Using Visual Studio

Choose the top Debug menu, then choose Start Debugging (F5). This should bring up a browser with the app running.

Navigate to the endpoint (you may need to change the port number) <http://localhost:8080/api/values>

1. Once the app loads in the browser you will see the two values output.  
   ["some-val","another-val"]